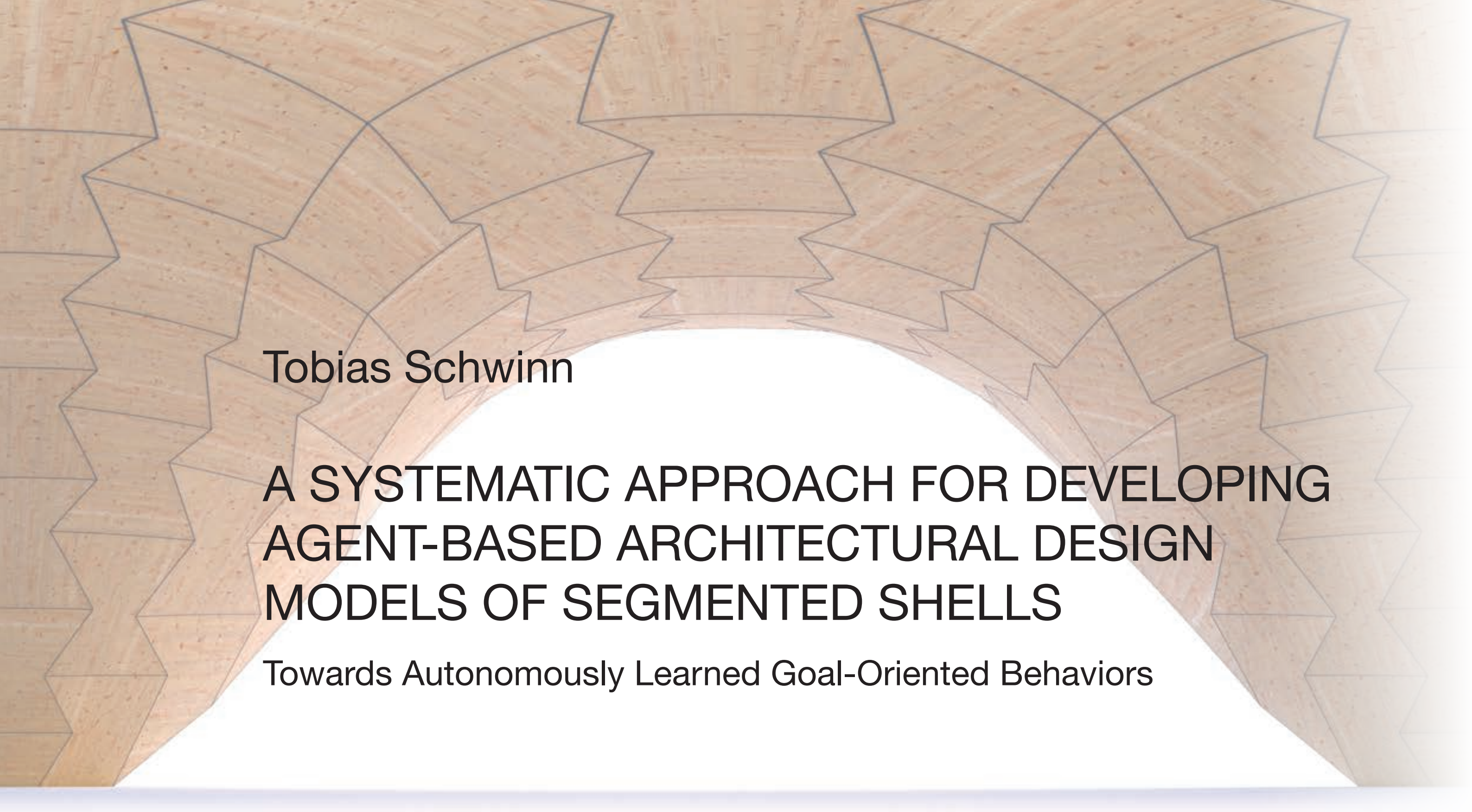
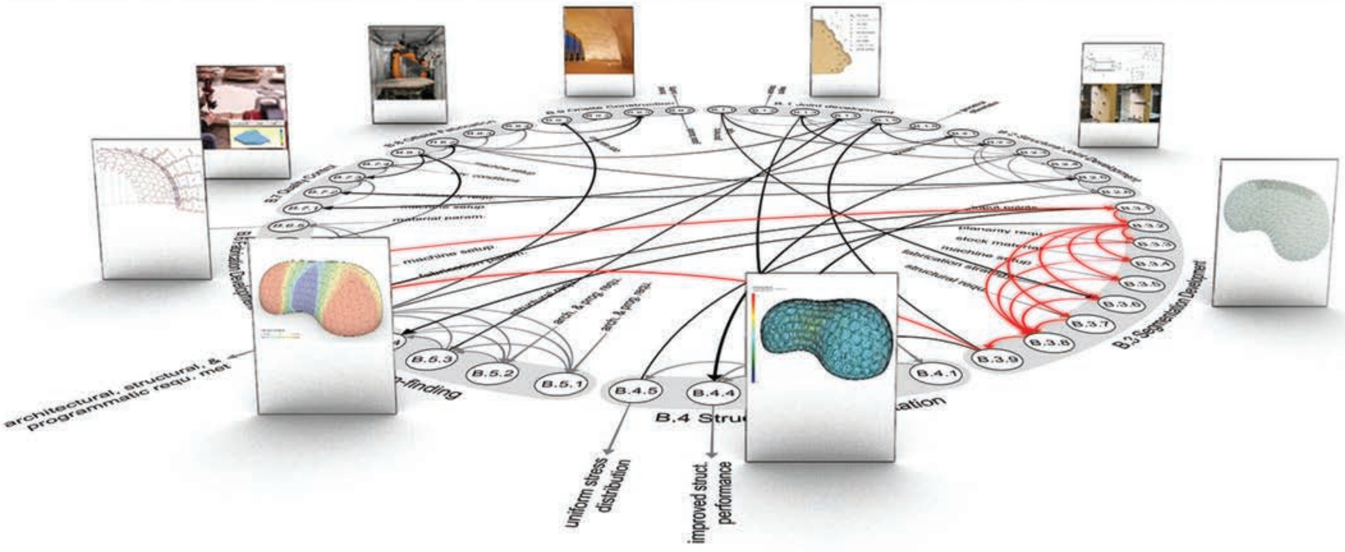


Tobias Schwinn

A SYSTEMATIC APPROACH FOR DEVELOPING AGENT-BASED ARCHITECTURAL DESIGN MODELS OF SEGMENTED SHELLS

Towards Autonomously Learned Goal-Oriented Behaviors



RESEARCH REPORTS
Institute for Computational Design and Construction

5

Tobias Schwinn

A SYSTEMATIC APPROACH FOR DEVELOPING AGENT-BASED ARCHITECTURAL DESIGN MODELS OF SEGMENTED SHELLS

Towards Autonomously Learned Goal-Oriented Agent Behaviors

RESEARCH REPORTS
Institute for Computational Design and Construction

5

RESEARCH REPORTS
Institute for Computational Design and Construction
Edited by Prof. Achim Menges

Tobias Schwinn
A SYSTEMATIC APPROACH FOR DEVELOPING AGENT-BASED ARCHITECTURAL DESIGN MODELS OF SEGMENTED SHELLS
Towards Autonomously Learned Goal-Oriented Agent Behaviors

University of Stuttgart
Keplerstrasse 11
70174 Stuttgart
Germany

University of Stuttgart
Institute for Computational
Design and Construction

D 93

RESEARCH REPORTS
Institute for Computational Design and Construction
5

ISBN 978-3-9819457-5-1

Foreword

The dissertation of Tobias Schwinn explores agent-based modeling. In contrast to other fields, in which agent-based modeling and simulation is already established and methodologically comprehensively researched, there are only a few investigations in architecture. The research of Tobias is therefore highly topical and also impresses with its methodological stringency and originality. It examines agent-based modeling as a design method using the example of segment shells. In a captivating way, it analyzes the relevant functional principles and cross-domain interactions, transfers them very convincingly into design principles, and implements them in an agent-based model. The methodological development of an integrative and interdisciplinary design approach based on agent-based modeling is also of considerable relevance beyond the specific application field of segment shells and represents a considerable contribution to research in this area.

Die Dissertation von Tobias Schwinn beschäftigt sich mit der agentenbasierten Modellierung. Im Gegensatz zu anderen Feldern, in denen sich die agentenbasierte Modellierung und Simulation bereits etabliert hat und methodisch umfassend erforscht ist, gibt es in der Architektur hierzu nur vereinzelte Arbeiten. Die Forschung von Tobias ist somit von hoher Aktualität und beeindruckt auch durch ihre methodische Stringenz und Originalität. Sie untersucht den Einsatz von agentenbasierter Modellierung als Entwurfs- und Planungsmethode am Beispiel von Segmentschalen. Auf imposante Weise analysiert sie die dabei maßgeblichen funktionalen Prinzipien und domänenübergreifenden Wechselwirkungen, transferiert diese sehr überzeugend in Entwurfsprinzipien und implementiert sie in ein agentenbasiertes Modell. Die methodische Entwicklung eines integrativen und interdisziplinären Entwurfsansatzes auf Basis der agentenbasierten Modellierung ist auch jenseits des spezifischen Anwendungsfeldes der Segmentschalen von erheblicher Relevanz und stellt einen beachtlichen Beitrag zur Forschung in diesem Gebiet dar.

Prof. Achim Menges

A SYSTEMATIC APPROACH FOR DEVELOPING AGENT-BASED ARCHITECTURAL DESIGN MODELS OF SEGMENTED SHELLS

Towards Autonomously Learned Goal-Oriented Agent Behaviors

A dissertation approved
by the Faculty of Architecture and Urban Planning of the
University of Stuttgart
for the conferral of the title of
Doctor of Engineering (Dr.-Ing.)

Submitted by
Tobias Schwinn
from Marktredwitz

Committee Chair:
Prof. Achim Menges

Committee member:
Prof. Jan Knippers

Date of the oral examination:
25.02.2021

Institute for Computational Design and Construction of the University of Stuttgart

2021

A SYSTEMATIC APPROACH FOR DEVELOPING AGENT-BASED ARCHITECTURAL DESIGN MODELS OF SEGMENTED SHELLS

Towards Autonomously Learned Goal-Oriented Agent Behaviors

Von der Fakultät für Architektur und Stadtplanung der
Universität Stuttgart
zur Erlangung der Würde
Doktor-Ingenieur (Dr.-Ing.)
genehmigte Abhandlung

Vorgelegt von
Tobias Schwinn
aus Marktredwitz

Hauptberichter:
Prof. Achim Menges

Mitberichter:
Prof. Dr.-Ing. Jan Knippers

Tag der mündlichen Prüfung:
25.02.2021

Institut für Computerbasiertes Entwerfen und Baufertigung der Universität Stuttgart

2021

Acknowledgements

This dissertation, written between 2017 and 2020, is based on previous work that started in 2011 with my time as a research associate at the Institute for Computational Design and Construction (ICD), University of Stuttgart, and extends all the way to the summer of 2020 when the latest research findings were made. During this time, I was fortunate enough to be able to build a knowledge base from research projects, such as the ICD/ITKE Research Pavilion 2011, the Landesgartenschau Exhibition Hall (2014), or the TRR141 A07 project (2014-2018), that forms the foundation for the research that is described in this dissertation. I would therefore like to thank and acknowledge all of my former colleagues and students who accompanied me along the long way to the completion of this work.

First and foremost, I would like to thank Prof. Achim Menges for his supervision and guidance throughout this time. He created and fostered the research environment at the ICD where so much unique and exciting research has been conducted over the years, which in many ways contributed to this work and shaped me as a researcher. His expertise and advice combined with the freedom to explore unknown research avenues and his trust are in my mind the foundation of this environment. Similarly, I would like to thank Prof. Jan Knippers at the Institute for Buildings Structures and Structural Design (ITKE) not only for being my second supervisor but also for his engaging in and promoting of collaboration between architects and structural engineers, which is the basis for the ground-breaking research demonstrators that have been produced in collaboration between ICD and ITKE since 2010.

Furthermore, I would like to extend my gratitude to my former ICD colleagues Oliver David Krieg for the collaboration during the design and development of ICD/ITKE Research Pavilions 2011 and 2015-16, and the Landesgartenschau Exhibition Hall project; Steffen Reichert for the collaboration during the ICD/ITKE Research Pavilion 2011; Ehsan Baharlou for laying the ground work for agent-based modeling at the ICD; as well as to Abel Groenewolt and Long Nguyen for the stimulating conversations and collaborative code development of ICD's Agent-Based Modeling Framework. I am particularly thankful for Long's patience when being

bugged with silly programming questions and, most importantly, for setting up the ABM Framework in such a robust and flexible way.

I would also like to thank my former ITKE colleagues Jian-Min Li for the collaboration during the Landesgartenschau Exhibition Hall, his open-mindedness, and for his encouragement to complete the dissertation; Markus Gabler (ICD/ITKE Research Pavilion 2011), Ricardo La Magna and Frederic Waimer (ICD/ITKE Research Pavilions 2011 and 2012), Simon Bechert (ICD/ITKE Research Pavilion 2015-16), and Daniel Sonntag (ICD/ITKE Research Pavilion 2015-16, TRR141 A07) for the collaboration during the respective projects.

Such interdisciplinary collaborations also form the basis of the unique M.Sc. program “Integrative Technologies and Architectural Design Research" (ITECH), which not only attracts the brightest minds from all over the world to Stuttgart every year, but in which many of the research ideas presented in this document have been tested (and sometimes challenged). Therefore, I would like to extend my gratitude to my ICD teaching colleagues past and present Martín Alvarez, Oliver Bucklin, Karola Dierichs, Benjamin Felbrich, Samuel Leder, Mathias Maierhofer, Marshall Prado, Katja Rinderspacher, Yasaman Tahouni, Lauren Vasey, and Maria Yablonina. Through discussion and exchange of ideas, mostly as part of the ‘Computational Design and Digital Fabrication’ and ‘Behavioral Fabrication’ seminars as well as the ‘Fabrication Agency’ thesis topic, the behavioral artificial intelligence approach to architectural design and fabrication has evolved to its current state. It is now applied in various different research projects including the Cluster of Excellence “Integrative Computational Design and Construction for Architecture” (IntCDC).

Last but not least, I am most grateful to my family who has supported every effort to see this work finished: to my parents Renate and Friedrich as well as my in-laws Anne and Hans-Christian for hosting and feeding me at various intervals during the last three and a half years and for proof-reading parts of this document; to my wife Christine and my sons Theo and Oskar not only for their unconditional support but also for bearing with me through long hours, lost weekends, and bungled vacations. Without you this endeavor would have been futile.

Tobias Schwinn, May 2021

Contents

Acronyms

ABM Agent-based Model. 18

ABMS Agent-based Modelling and Simulation. 18

ABS Agent-based System. 384

ACO Ant Colony Optimisation. 54

AI Artifical Intelligence. 18

ALife Artifical Life. 23

B-rep Boundary Representation. 241

CA Cellular Automaton. 30

CAD Computer-Aided-Design. 8

CAE Computer-Aided-Engineering. 8

CAGD Computer-Aided-Geometric-Design. 20

CAM Computer-Aided-Manufacturing. 8

CAS Complex Adaptive System. 18

CNC Computer-Numerical-Control. 8

CS Computer Science. 331

DES Discrete-Event Simulation. 27

DfX Design for Excellence. 12

DOF Degree of Freedom. 134

DR Dynamic Relaxation. 87

EBM Equation-based Modelling. 44

FDM Force-Density-Method. 87

FE Finite-Element. 219

FEA Finite-Element-Analysis. 5

FEM Finite-Element-Modelling. 154

FRP Fibre-Reinforced-Polymer. 85

GA Genetic Algorithm. 33

HRC Human-Robot-Cooperation. 285

ICD Institute for Computational Design and Construction. 105

IIGS Institute of Engineering Geodesy. 248

ITECH “Integrative Technologies and Architectural Design Research". x

ITKE Institute for Building Structures and Structural Design. 105

KRL KUKA Robot Language. 244

LVL Laminated-Veneer-Lumber. 215

MAS Multi-Agent System. 18

NURBS Non-Uniform Rational Basis Spline. 159

PS Particle Spring method. 87

PSO Particle Swarm Optimisation. 49

RL Reinforcement Learning. 383

RMS Root Mean Square. 248

SD System Dynamics. 27

TCP Tool Center Point. 246

TNA Thrust-Network-Analysis. 87

TPI Tangent-Plane-Intersection. 203

XML Extensible Markup Language. 276

List of Figures

List of Tables

Abstract

Segmented shell design constitutes a novel and promising research area in shell design that has emerged in the last 10 years. The prospect of dividing a continuous shell surface into segments is to resolve some of the constraints of continuous shells that have limited their application in building practice. As part of large-span surface structures, segmented shells have shown to possess similar desirable features, such as low material usage, the ability to cover large areas column-free, and lightweight aesthetics, while allowing for a high degree of prefabrication. The geometry of individual building elements and global form are, however, complex, which poses a challenge to designing and building segmented shells.

One of the challenges of segmented shell design in particular is meeting multiple interrelated, sometimes conflicting, evaluation criteria: geometric validity, structural stability, and producibility. In segmented shell design geometric validity and producibility are aspects that can be considered locally, meaning on the level of the individual building element, while structural stability needs to be evaluated globally and can be conceived of as the global effect of the properties and interactions of all segments in the shell.

A 'microscopic' modeling perspective has therefore been proposed in previous work that bridges the gap between local characteristics and global performance: agent-based modeling and simulation (ABMS). By focusing on the detailed description of the individual building elements and their interactions and by conceiving of the global form as the result of a myriad of local interactions of virtual agents representing building elements, the global design problem can be solved in parallel on the level of the individual building elements. In particular, the Landesgartenschau Exhibition Hall project, completed in 2014, has shown the viability of the agent-based modeling approach for segmented shells.

While ABMS is an established method in simulation-based research fields, such as ecology and robotics, the method needs to be considered far from established in the area of architectural design despite early adoption. This is visible in the lack of a common methodology (and even the discussion thereof) and of software frameworks geared towards Computer-Aided Geometric Design (CAGD). To address the latter issue, the development of an ABMS software framework for CAGD was started at the Institute of Computational Design and Construction (ICD) in 2016 in order to provide a shared platform for experimentation with agent-based systems. The former issue, the question of a common methodology, is the subject of this dissertation.

The aim is to propose a methodology for developing agent-based models of buildings where agents constitute building elements. With the special focus on plate shells (segmented shells with planar elements) the research pursues and synthesizes two investigative strands: on the one hand, generalizing findings from previously built plate shells as part of a case study-based, inductive research approach, which is geared towards building a catalog of validated design principles for plate shells; on the other hand, systematizing the agent-based modeling approach for architectural design-oriented applications in general, and plate shell design in particular.

This dissertation is organized into three parts: The first part introduces motivation, aim, relevance, and scope of the dissertation as well as the related work in the fields of architectural shell design and agent-based modeling and simulation. It also situates the topic within the historical context of the two fields.

The second part presents the case study-based approach for abstracting validated design principles from previously built segmented shells. Through *analysis* and *abstraction*, functional principles are distilled from the basic case data, followed by the *transfer* from functional to design principles and their mapping to constructs of the agent-based modeling paradigm.

The third part focuses on the *implementation* of selected design principles as the final step in the proposed methodology for developing agent-based models of segmented shells. The proof-of-concept takes the form of a problem-oriented case study. The study is conceived of as a series of experiments with increasing complexity that not only addresses previously identified challenges of agent-based models (ABM) of segmented shells, such as the need for defining a ‘hand-designed’ reference surface, but also presents novel findings with respect to 1) parameter optimization of ABMs of segmented shells and 2) autonomous learning of agent behaviors using Reinforcement Learning.

In summary, part II, with its analysis, abstraction, and transfer of design principles, presents the first three steps and part III the last step of the proposed methodology. The main contributions of this work are thus seen in generalizing knowledge from specific cases of plate shells as part of a catalog of validated design principles (part II) and in proposing and demonstrating a systematic approach for developing agent-based architectural design models of segmented shells (parts II and III combined).

Zusammenfassung

Die Planung von Segmentschalen ist ein neues und vielversprechendes Forschungsfeld im Bereich der Schalenplanung. Die Aufteilung kontinuierlicher Schalenflächen in Segmente stellt die Lockerung und Auflösung von Einschränkungen hinsichtlich der Planung und Realisierung kontinuierlicher Schalen in Aussicht, die deren Anwendung in der Baupraxis begrenzen. Als Teil der Kategorie der weitspannenden Flächentragwerke besitzen segmentierte Schalen vergleichbare erwünschte Eigenschaften, wie zum Beispiel ein geringer Materialverbrauch, die Fähigkeit große Flächen stützenfrei zu überspannen und eine eigene Leichtbauästhetik, während sie gleichzeitig an großes Maß an Vorfertigung ermöglichen. Die Geometrien individueller Bauteile sowie die Globalform der Schale sind jedoch komplex, woraus sich wiederum besondere Anforderungen an Planung und Umsetzung segmentierter Schalen ergeben.

Eine besondere Herausforderung bei der Planung von Segmentschalen sind die vielfältigen und teilweise widersprüchlichen Wechselbeziehungen zwischen den Bewertungskriterien geometrische Validität, Stabilität und Herstellbarkeit. Darüber hinaus sind geometrische Validität und Herstellbarkeit *lokale* Aspekte, die auf der Ebene der Bauteile bewertet werden können, wohingegen Stabilität und Tragfähigkeit *globale* Aspekte darstellen, die auf der Ebene der Schale bewertet werden müssen. Aus der lokalen Perspektive betrachtet kann die Tragfähigkeit einer Segmentschale somit als ein globaler Effekt betrachtet werden, der sich aus den Eigenschaften aller Segmente in der Schale und aus deren Verbindungen untereinander ergibt.

In Vorarbeiten wurde daher eine sogenannte *mikroskopische* Modellierungsperspektive eingenommen, die eine Brücke zwischen den Eigenschaften auf der lokalen Ebene der Bauteile und den globalen performativen Eigenschaften der Schale schlägt: agentenbasierte Modellierung und Simulation (ABMS). In diesem Ansatz geht man davon aus, dass Globalform und -verhalten der Schale das Ergebnis unzähliger lokaler Interaktionen virtueller Agenten ist, die Bauteile darstellen. Indem sich dieser Modellierungsansatz auf die detaillierte Beschreibung der individuellen Bauteile und deren Verbindungen (Interaktionen) untereinander konzentriert, wird

das globale Entwurfsproblem sozusagen auf die Ebene der individuellen Bauteile verlagert und dort parallel gelöst. Die Anwendbarkeit des agentenbasierten Modellierungsansatzes für die Planung von Segmentschalen wurde insbesondere in dem Forschungsprojekt "Robotik im Holzbau" aufgezeigt, im Rahmen dessen das Ausstellungsgebäude der Landesgartenschau 2014 in Schwäbisch Gmünd realisiert wurde.

Während ABMS in simulationsaffinen Forschungsbereichen, wie zum Beispiel der Umweltforschung oder der Robotik, eine etablierte Methode darstellt, ist sie im Bereich der Architektur alles andere als etabliert, obwohl sie dort bereits seit Anfang der 90er Jahre des 20. Jahrhunderts angewendet wird. Dieser Umstand zeigt sich unter anderem in der Abwesenheit einer etablierten Methodik (und auch einer entsprechenden Diskussion darüber) und dem Fehlen eines Softwareframeworks, das den Anforderungen des computergestützen geometrischen Planens (CAGD) gerecht wird. Um den letztgenannten Punkt anzugehen, wurde 2016 am Institut für Computerbasiertes Entwerfen und Baufertigung (ICD) mit der Entwicklung eines ABMS Softwareframeworks für CAGD begonnen, mit dem Ziel eine einheitliche und gemeinsame Experimentierplattform für agentenbasierte Systeme in Architektur und Bauwesen bereitzustellen. Der erstgenannte Punkt, die Frage nach einer gängigen Methodik, ist der Untersuchungsgegenstand dieser Dissertation.

Ziel dieser Arbeit ist es, eine Methodik für die Entwicklung agentenbasierter Gebäudemodelle vorzuschlagen, in denen Agenten Bauteile darstellen. Mit dem besonderen Schwerpunkt auf der Planung von Schalen mit planaren Segmenten verfolgt und verbindet diese Arbeit zwei Untersuchungsstränge: einerseits die Verallgemeinerung von Erkenntnissen, die in vorherigen gebauten Segmentschalen gewonnen wurden, im Rahmen eines induktiven, fallstudienbasierten Forschungsansatzes; und zweitens die Systematisierung des agentenbasierten Modellierungsansatzes für architektonische Entwurfsanwendungen im Allgemeinen und für die Planung von Segmentschalen im Besonderen.

Die Dissertation besteht aus drei Teilen: Der erste Teil stellt Motivation, Ziele, Bedeutung und Forschungsschwerpunkte der Arbeit vor, sowie verwandte Arbeiten aus den Bereichen architektonische Schalenplanung und agentenbasierte Modellierung und Simulation. Desweiteren positioniert dieser Teil das Forschungsthema in den jeweiligen historischen Kontexten der beiden Themenbereiche.

Der zweite Teil stellt den fallstudienbasierten Ansatz für die Definition valider Entwurfsprinzpien anhand gebauter Segmentschalen vor. Mittels *Analyse* und *Abstraktion* werden zunächst funktionale Prinzipien aus den Falldaten destilliert. In dem anschließenden *Transfer* werden dann Entwurfsprinzipien definiert und den Konstrukten des agentenbasierten Modellierungsansatzes zugeordnet.

Im Mittelpunkt des dritten Teils der Arbeit steht der letzte Schritt der vorgeschlagenen Methodik zur Entwicklung agentenbasierter Modelle segmentierter Schalen: die *Implementierung* ausgewählter Entwurfsprinzipien. Der Schritt der Implementierung dient als Nachweis der Plausibilität und nimmt die Form einer problembezogenen Fallstudie an. In einer Reihe von Experimenten mit zunehmender Komplexität werden darin nicht nur die in vorherigen Arbeiten identifizierten Herausforderungen angegangen, die segmentierte Schalen an die agentenbasierte Modellierung stellen, wie z.B. die Notwendigkeit, eine händisch gestaltete Referenzfläche zu definieren; sondern auch neue Erkenntnisse vorgestellt: vor allem hinsichtlich 1) der Parameteroptimierung von agentenbasierten Modellen segmentierter Schalen und 2) dem selbständigen Erlernen von Agentenverhalten mittels bestärkendem Lernen.

Zusammenfassend lässt sich somit sagen, dass Teil II mit Analyse, Abstraktion und Transfer die ersten drei Schritte der vorgeschlagenen Methodik vorstellt und Teil III mit der prototypischen Implementierung von Entwurfsprinzipien den letzten Schritt. Die Arbeit liefert daher die folgenden wesentlichen Beiträge zu den genannten Bereichen der Planungsmethodik für Segmentschalen und der agentenbasierten Modellierung: die Abstraktion und Generalisierung von projektspezifischem Wissen aus konkreten, gebauten Beispielen von Segmentschalen (Teil II); sowie, in Kombination der Teile II und III, Vorschlag und Demonstration eines systematischen Ansatzes für die Entwicklung von architektonischen Entwurfsmodellen segmentierter Schalen mittels agentenbasierter Modellierung.

PART I

Introduction

Buildings are always changing, so that a city's coherence is somehow imposed on a perpetual flux of people and structures. Like the standing wave in front of a rock in a fast-moving stream, a city is a pattern in time.

—John Holland [85, p. 27]

1

Context

1.1 An Iconic Example of a Thin Shell

In 1958, Spanish-born architect and engineer Félix Candela (1910-1997), one of the protagonists of 20th century shell design,[1] saw one of his most seminal designs completed: the roof of the “Los Manantiales” Restaurant at Xochimilco, Mexico. With its mathematically defined shape, where four intersecting saddle-shaped hyperbolic paraboloids (hypars) form a groined vault over an octagonal plan [61, p. 142], and its thin reinforced concrete shell, it is considered one of the iconic examples of 20th century shell design (Fig. 1.1).

The shell—once surrounded by floating gardens and resembling a floating flower—raises up to 8.25 m height at the apex of its exterior arches and lowers down to 5.90 m at the interior. The building is characterized by the light-filled, column-free interior space, which is used as restaurant, and the general impression of lightness that the structure conveys: its undulating roof barely seems to touch the ground. Emery [54, p. 7] even states that Candela’s buildings “seem to defy gravity, like giant swaths of fabric blown aloft by a soundless wind.” The effect is the result of the shell’s cantilevering edges, free of any stiffeners, that reveal its remarkable thinness of only 4 cm while spanning 32.4 m (measured between the supports along the groins) [61, p. 142]. Remarkably, Candela did not perform sophisticated calculations and structural analysis during the design of the shell. According to Burger and Billington [35, p. 278], who have analyzed the structural behavior of the shell using Finite-Element-Analysis (FEA) and confirmed the structural effectiveness of the design, Candela was fond of saying that “the quality of a structure is in inverse proportion with the amount of calculations.” The structural achievement is further

---

[1] David Billington (1927-2018), professor emeritus of structural engineering at Princeton, called Candela “one of the giants of thin-shelled structures” [54, p. 7].

Figure 1.1: Los Manantiales Restaurant in Xochimilco, Mexico City by Félix Candela. From [159] by RIBA Collections (1958).

underscored by the fact that the shell has survived the devastating 2017 earthquake that struck the Mexico City region "with just a few bruises and bends" [145].

Elegance and simplicity of the design are also reflected in its construction, where the formwork for the concrete could be built from straight timber boards that follow the path of the straight-line generators of the hypar surface. This, together with low labor costs at the time, made the construction of such a geometrically complex structure economically feasible. The shell has been celebrated for its technical virtuosity and its aesthetic sensitivity [61, p. 152] and, consequently, has been extensively studied[2] and even copied.

Notably, the shell has been rebuilt with slight variations in different locations, for example, in Stuttgart (1977 by Jörg Schlaich), in Potsdam (1983 by Ulrich Müther), and most recently, in Valencia (2002 by CMD Ingenieros with Candela), where it is part of an aquarium in the 'Ciutat de les Arts i de les Ciències'. This case, as the most recent 'incarnation' of Candela's thin-shell design, provides a unique opportunity for a comparison of the tools and technologies used during the development of the shells and, in particular, of those used during their construction.

---

[2] For example, as part of the exhibition "Félix Candela: Engineer, Builder, Structural Artist" at the Princeton University Art Museum in 2008.

1.2 The Problem with Thin-Shell Structures

When comparing photographs of the construction site in Xochimilco from 1958 with those from 1999 in Valencia, it is most striking to see that the construction methods appear to have barely changed in more than 40 years (Fig. 1.2). Most notably, in each case, the shell has been built twice: first in timber, as custom-built formwork, and then in concrete. Given low labor costs, this was considered economical at the time of construction of the shell in Xochimilco [54, p. 7]. Furthermore, no formwork has been reused in either case even though the form of the structure is repetitive, because Candela chose to form the entire structure at once in order to balance the forces in the shell during construction [61, p. 148]. In Xochimilco, steel reinforcement was installed through labor-intensive manual work, whereas in Valencia a steel-fibre reinforced concrete in combination with a steel mesh was used [25]. Finally, Los Manantiales was poured by hand, one section at a time, with workmen moving over the formwork [35, p. 276] (Fig. 1.2, left), whereas in the case of L'Océanografic concrete was applied using shotcrete technology [25]. In both cases, scaffolding consequently needed to support the entire shell surface during construction.

Figure 1.2: Comparison of construction methods. Left: Los Manantiales Restaurant in Xochimilco, Mexico (Guzmán, 1958). Right: L'Oceanogràfic in Valencia, Spain (CMD Ingenieros, 1999).

Despite the efficiency of the final structure and its indisputable elegance, the comparison suggests why design and construction of thin continuous shells have effectively remained a niche endeavor after a period of experimentation in the early-to-mid 20th century. While some of the most exceptional continuous surface shells were designed and realized during this period, as demonstrated by the work of Candela, Dieste, Nervi, Isler, Saarinen, Müther and others (see section 4.1.3), beginning in the 1970s, the construction of new monolithic shells made of reinforced concrete or masonry declined. Though still considered highly efficient from a structural point of view, they were not competitive anymore mainly due to “scarcity of skilled workers, rising prices for formwork, cost and schedule challenges, inefficiency of on-site fabrication” [164, p. 259] and, in the case of concrete shells, the high effort needed for customized reinforcement. Instead, “grid shells consisting of cost-efficient lattice systems covered by planar glass or metal panes” [91, p. 184] have become the predominant structural type to realize large-span, form-passive surface structures.[3]

While the study of the exact reasons for this transition from continuous surface shells to discrete grid shells is outside the scope of this research, the following aspects have certainly contributed to it:

1. raising costs of manual labor;
2. planarity of building elements increasing cost-efficiency;
3. the higher degree of pre-fabrication in the construction of grid-shells yielding a reduction of on-site labor and increase in quality.

In any case, “architects lost interest” in reinforced concrete shells [164, p. 259].

1.3 New Opportunities for Thin-Shell Structures

The 1990s saw the maturing development and increasing presence of digital design, simulation and fabrication methods and, correspondingly, a changing industrial production paradigm from serial production to mass customization. The technological context of the early 21st century now provides the opportunity to revisit the challenges posed by the design and realization of thin shell structures.

In particular, Computer-Aided-Design (CAD), Computer-Aided-Engineering (CAE), Computer-Aided-Manufacturing (CAM), Computer-Numerical-Control (CNC) of machining, and increasing automation enabled by industrial robotics enable a new approach to the design and construction of thin shell: segmentation. Segmentation has the potential to combine advantages of continuous shells (structural efficiency) and grid-shells (prefabrication), while taking advantage of contemporary technology. The application of these technologies, such as CNC and robotics, not only allows for economic pre-fabrication of geometrically unique building parts (batch size 1) required by complex surface structures, but also for quality control and just-in-time delivery on-site, where the segments can be assembled over minimal false-work. The viability of this proposition has been confirmed with the realization of the Landesgartenschau Exhibition Hall in Schwäbisch Gmünd, Germany, in 2014 [103] and, most recently, with the BUGA Wood Pavilion in Heilbronn, Germany, in 2019 [7] (Fig. 1.3). Both segmented shells have been

---

[3] In contrast to form-active surface structures, such as membranes, shells are form-passive structures that do not significantly change shape when loaded.

designed, engineered, and fabricated by the Institutes for Computational Design and Construction (ICD) and for Building Structures and Structural Design (ITKE) at the University of Stuttgart in collaboration with Müllerblaustein Holzbau GmbH.

Figure 1.3: Recent examples of segmented timber shells. Left: Landesgartenschau Exhibition Hall. From [103] by Krieg (2014). Right: BUGA Wood Pavilion (Faulkner, 2019).

In addition to the works mentioned above, examples for segmented shells can be found in recent work by the Block Research Group at the ETH Zürich and by the Institute of Lightweight Structures and Conceptual Design (ILEK), University of Stuttgart. The majority of segmented shells that have been built so far have thus been developed in the academic context. Barriers to the adoption of segmented shell design in practice are posed by the challenges associated with the design and fabrication of segmented shells and by open research questions. In particular, thin-shell structures typically exhibit complex double-curved geometry as the result of form-finding methods, which are not readily available to practicing architects, or require mathematical modeling approaches that cannot be easily implemented in conventional architectural design tools. Furthermore, the segmentation of the shell surface interrupts the stiffness continuity and therefore not only requires advanced computational design and engineering methods in order to meet given performance criteria; but it is also not yet fully understood how the interplay between global geometry, component geometry, and joint geometry affect the structural performance of segmented shells [177, p. 122].

1.4 Lightweight Design and Sustainability

The renewed interest in thin shell structures and, more generally, in lightweight construction, is not only due to technological developments, but also due to a growing interest in and need for resource-effectiveness in the building industry. Buildings are responsible for more than 40% of global energy use and as much as one third of global greenhouse gas emissions, both in developed and developing countries [187, p. 6]. This includes energy and emissions associated with the production of the building materials, such as kiln-fired bricks and Portland cement used in concrete, so-called embodied or ‘grey’ energy, and with the operation of buildings over their lifespan. With industrial production of building materials starting in the 1800s and given the median lifetime of buildings (58 years for small office buildings in the US [143, p. 3-10]), the building sector is historically responsible for a substantial part of the present societal challenges posed by climate change. In view of a projected doubling of the global building stock by 2060 relative to 2017 [194, p. 13] due to global urbanization trends, if no ambitious effort is made to address low-carbon and energy-efficient solutions for buildings and construction, then buildings-related CO_2 emissions will consume half of the remaining carbon budget that is associated with an average global warming limit of 2°C (as set forth in the Paris Agreement of 2015) [194, p. 20]. Therefore, strategies and measures are required to reduce the impact of buildings on the environment.

There are essentially two complementary strategies to reduce the environmental impact of buildings:

1. to reduce resource consumption and emissions during the operation phase of buildings (operation challenge), and
2. to reduce the amount of (fossil) energy and greenhouse gas emissions associated with the production of buildings (production challenge).

While the operation challenge primarily constitutes a challenge for the existing building stock, it is essentially solved for new buildings, e.g., through Passive Building standard, the use of renewable energy sources, net zero and net positive energy strategies (where buildings can serve as energy sources for the electrical grid). The challenge of reducing resource consumption and emissions during production, on the other hand, remains open.

One approach to address the production challenge is to increase the use of regional and renewable materials, such as wood, thereby aiming for low embodied energy. A second approach is lightweight design and construction, which aims at ‘building more with less’: in other words, increasing the load-bearing capacity to self-weight ratio of a structure, thereby increasing the utilization of the structural material. Lightweight design, however, requires an integrative design approach as form and force-flow are highly correlated. This means that conventional building designs, which typically require post-rationalization, can hardly be re-engineered in later design phases for light weight. Therefore, lightweight design principles, principles of materialization, and other performance-related aspects have to be integrated into the earliest design stages. The challenge is that these aspects are typically evaluated at different stages of the planning process with design and materialization being located at its opposite ends.

1.5 The Challenge of Design Integration

According to Bogenstätter [22, p. 376]), programming and building specifications in early design phases determine up to 80% of the environmental pollution as well as the building operation costs. According to HOAI [186, §34], early design stages account for 22% of the time and budget available for the entire design process. The generation and evaluation of design variants based on given requirements even has only 7% of the planning effort allocated to it (ibid.). Within this short timespan the architect has to navigate an extremely large solution space, whose definition might change during the course of the project. Consequently, there is an obvious discrepancy between the impact of design decisions and the time and resources allocated to develop and make those decisions. The early-stage design methods available to architects therefore play a crucial role in the resource-effectiveness of buildings.

On the one hand, the significance of early design stages and the role of architects as the originators of building designs has been recognized and design integration for conventional buildings is a research topic that is actively worked on in academia and practice (see, for example, MacLeamy [125]). There, the focus is on bridging the gap between project stages and stakeholders and aims at data continuity through a *Digital Chain*. On the other hand, the design of lightweight structures, and of segmented shells in particular, requires a higher integration of the inter-disciplinary planning effort than conventional buildings, as design tools and methods for lightweight structures need to anticipate performative potentials and the possibilities of materialization during the early design stages. Those aspects are not only at the opposite end of the planning spectrum, but such tools and methods are also largely unavailable to architects.

Somewhat paradoxically, while digitalization enables the design and realization of lightweight structures and segmented shells, the formalization of the conventional planning approach in software constrains feedback between stages and stakeholders and, by consequence, the application of segmented shells in practice.

With pre-fabrication and automation being the pre-conditions for the effective construction of segmented shells, designing for segmented thin shells in particular, and for light weight in general, is therefore out of reach for most architectural practices. With further advances in fabrication technology on the horizon, including autonomous robotics, the gulf between methods of design and fabrication in architectural production is expected to broaden even further and the benefits of technological advances in manufacturing will remain inaccessible to architects and, in turn, to the design of the built environment. It will thus be next to impossible, given the current state of design technology, to translate technological advances into ecological and economic benefits, and architectural opportunities.

1.6 Automotive and Aerospace Industries

The lack of integration and design methods for lightweight construction in architecture needs to be seen in contrast to other industries, such as automotive and aerospace industries. There, *Design for Lightweight* is an established design approach, particularly with respect to fuel efficiency, as part of the Design for Excellence (DfX) paradigm [24, p. 195] and has spurned research into new materials, fabrication and simulation technology. Design and production are also much more integrated than in the building industry. This shows that a higher integration is possible. The ensuing question is why the tools and methods from production engineering have not yet been transferred to architecture. Again, while the study of such a transfer is outside of the scope of this research, it is straightforward to point out differences between the industries that limit the prospects of such a transfer: every building design responds to the specific requirements of its client and to the site, in which it is to be situated. Buildings therefore usually are one of a kind (batch size 1) and their design and production are more akin to bespoke manufacturing than assembly line production.

Another major difference is that the building sector has grown over hundreds of years and is highly socially embedded consisting of many specialist trades and small enterprises and therefore is also highly segregated. As a consequence, design and construction are separated and architects typically do not own the means of production. A notable exception is, again, Félix Candela, who was not only architect and engineer, but also contractor [62, p. 254], which allowed him to pursue the development of thin-shell structures. In Germany, the separation between design and construction is even ingrained in the system and a licensed architect is prohibited from acting as the contractor of a project due to the respective mutual control obligations towards the client (see [186, p. 92], [209]).[4] In comparison, the other two industries have emerged fairly recently within the last 100 years and therefore established and consolidated their business structures in the context of industrial mass production. This is particularly the case in the automotive industry, which even invented industrial scale mass production with Henry Ford's invention of the assembly line.

Nevertheless, the potential benefits of lightweight design are undoubtedly also

---

[4] Interestingly, a contractor is not prohibited from offering architectural services.

available to buildings: a possible reduction of embodied energy in buildings through the use of less material; a larger span of lightweight structures through form-optimization. Given the environmental challenges that the building industry is facing, lightweight design therefore provides the opportunity for more resource-effective production of buildings. Within the field of lightweight design in architecture, thin-shell structures offer the potential for lightweight structural buildings envelopes, where the shell is at the same time separation between interior and exterior, and load-bearing structure. Thin shells are considered material efficient—material is only needed where the internal forces 'flow' and vice versa—and elegant, as their form is the expression of the internal forces of the structure. The reciprocity of form, function, and structure, considered a given in the case of automotive and aircraft design, also offers an opportunity for architectural design. Particularly the work of Frei Otto in the 20th century has shown that lightweight design can yield an aesthetic that is the expression of this reciprocity.

2

From Motivation and Aim to Approach and Structure

2.1 Motivation and Aim

In light of the previously mentioned technological and architectural opportunities on the one hand and environmental and technical challenges on the other, building lightweight segmented shells out of wood appears to be a meaningful proposition. However, as outlined above, conventional design approaches and their incremental development as part of the Digital Chain appear ill-equipped to take advantage of the opportunities and to be able to address the challenges. One main aspect is the lack of concurrent design possibilities, also known as simultaneous engineering [24, p. 195], due to the separation of the planning process; another the lack of systematic computational feedback between planning steps and stakeholders.

From a system-theoretical point of view, feedback can be defined as: "the outputs of a system at time t affect the inputs of that system at time $t + 1$" [38, p. 10]. In the specific context of architectural design process, the system is the network of individual actors in the planning process. This system changes over time, i.e. over the course of the planning process. The output of the system at the end of the planning process is the finished building, while every planning step in the process generates individual outputs that form inputs for subsequent steps. Feedback might thus be defined as:

> Outputs (as the result of design decisions) at individual steps in the planning process affect the inputs (premises on which decisions are taken) of the same or previous steps in the planning process.

A diagrammatic representation of this definition is shown in Fig. 2.1. An

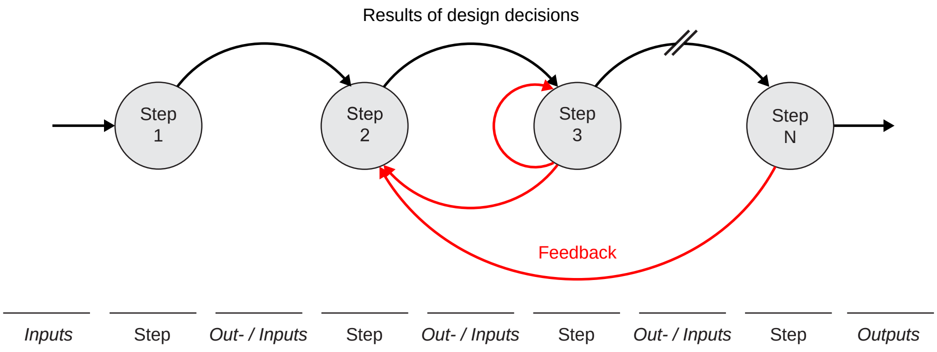


Figure 2.1: Schematic definition of feedback in the architectural design process.

example would be the decision, typically taken at a later stage of the planning process, to use a particular fabrication approach, which affects what can be produced and therefore what should have been designed earlier in the process.

The challenge that lightweight design is facing in practice is therefore one of design integration and computational methods. The premise of this research is that computational approaches can enable and support this kind of systematic computational feedback between design, engineering, fabrication, and construction of segmented shells. The prospect of such an integration is an expansion of the domain, in which design decisions can take place that inform building design (Fig. 2.2).

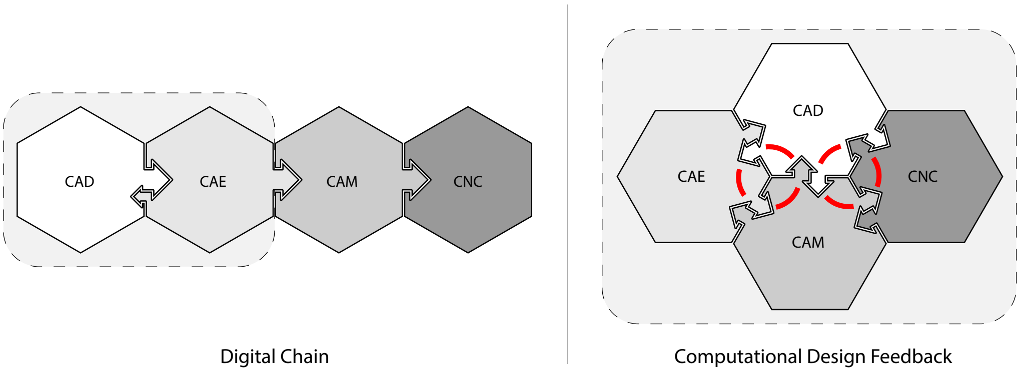


Figure 2.2: Digital Chain vs. Digital Loop (Schwinn, 2015). Left: the status quo. Right: the aim of design integration. The dotted lines indicate the areas where design feedback can take place.

The expansion of the design domain, on the other hand, entails the question of how such ever increasing domains can effectively be navigated. According to W. Ross Ashby, a pioneer of cybernetics who introduced the term Machine Augmented Intelligence,[1] any problem-solving task can ultimately be reduced to "out of a certain set, indicate one element." The process of defining the right set

---

[1] see W. Ross Ashby. Design for an Intelligence-Amplifier. In C. E. Shannon & J. McCarthy, editors, *Automata Studies*, pages 215–233. Princeton: Princeton University Press, 1956.

is therefore just as important as the process of selecting the right candidate, in that only if the set actually contains the desired candidate, it can be found by the selection process. Architectural design, as the task of solving a particular "design problem," can similarly be described as an iterative search process—however, as of today, mostly unfolding intuitively based on prior experience and knowledge—that alternates between generating alternatives (divergence) and selection (convergence).

Computational design offers the possibility to *augment* the design process, in that it can not only assist with the selection of viable design candidates from a set of possible candidates, which is defined by the intersection of all objective functions (design optimisation), but, in the first place, it can redefine the solution space and its set of possible candidates (design exploration). Architectural design exploration takes into account that design problems are typically ill-defined and that the formulation of the design problem is subject to change and evolve during the early stages of the design process, as mentioned above. Consequently, optimality criteria cannot be defined at an early stage. The advantage of a changing solution space (divergance) during design is, however, that it can avoid missing a potentially superior solution [45, p. 649] and thus can ultimately lead to a better overall solution to the design problem.

The aim of this research therefore is to investigate methods that allow for such computationally augmented divergence and convergence by exploiting systematic computational feedback between design, engineering, fabrication, and construction of segmented shells; and to prototypically test such methods with a focus on the early architectural design stages. In summary, this aim is motivated by the following reasons:

1. Recent developments in digital design and engineering address the problem of geometric description and analysis of complex surface geometry.
2. Recent developments in digital fabrication address main limitations of the serial production paradigm. Together, points 1 and 2 provide the opportunity to revisit the design and realization of thin shell structures.
3. Segmentation of shell structures provides the possibility to raise the level of pre-fabrication in the construction of shell structures.
4. Lightweight structures can reduce the amount of embodied energy in structures as material is only used where it is needed ("build more with less").
5. Design methods for lightweight structures are currently not available to architects despite the significance of the early design stages.
6. Feedback is required between the different stages of the planning process, specifically between design and materialization.
7. The use of a naturally renewable and regionally available resource, such as

wood, provides an additional opportunity for reducing the environmental footprint of buildings.

Furthermore, from a historical perspective, “technology has always been a catalyst for design innovation in architecture” [176, p. 224] as new technologies have yielded novel architectural expressions and structural typologies. Inventions, such as steel-reinforced concrete in the 19th century, the assembly line, and serial production in the 20th century, have been synthesized by the Bauhaus into a new expression and defined the modernist movement in the 20th century. The fact that modernism still influences the majority of contemporary construction demonstrates the impact that these inventions continue to have on the built environment. 100 years after the Bauhaus, a similar development can be currently observed, which is based on the synthesis of computation, robotics, and material advances. Exploring the architectural potential of these contemporary technologies for segmented shells thus is a further motivation to pursue this research.

2.2 Framing the Research Questions

The research is framed by the question of which means and methods are necessary to enable architects to design and realize high-performing structures, such as segmented shells, according to discipline-specific requirements within an interdisciplinary design context.

Enabled by increasing computational capacities, behavioral Artifical Intelligence (AI) approaches, such as Agent-based Modelling and Simulation (ABMS), seem particularly promising to be able to integrate and synthesize different disciplines and stages of the architectural design and production process. Previous related work by Baharlou [10], Baharlou and Menges [11], and Schwinn et al. [172] has focused on the concurrent integration of multidisciplinary design requirements in the design process using ABMS. Together this work has shown that this approach allows for feedback between materialization and design by representing both aspects within the same behavioral paradigm. In this approach, agents and behaviors, which constitute the building blocks for building Complex Adaptive System (CAS) and Multi-Agent System (MAS), are used for the concurrent integration of multidisciplinary design requirements. Furthermore, Agent-based Models (ABMs) have a number of characteristics that make them applicable in the architectural design context:

1. A bottom-up modeling approach allows for open-ended explorative design as well as goal-oriented search of the solution space.
2. Feedback, which is the prerequisite of design integration as mentioned above, is a fundamental concept of ABMS.

3. The individual-based modeling approach of ABMS allows representing a range of different simulation scenarios within one modeling paradigm; specifically those that are consistent with methodological individualism (see [132, p. 20] and section 3.4).[2]
4. The conceptual models, on which ABMs are based, are easier to communicate to stakeholders than comparable methods, such as system dynamics [179, p. 206].
5. Agent models are flexible, particularly in comparison with parametric models, and can be expanded iteratively in order to produce the desired target phenomenon [132, p. 24] thereby scaling alongside the planning stages of the design process.

Bonabeau [23] outlines further benefits provided by the agent-based modeling approach, such as "ABM captures emergent phenomena" and "ABM provides a natural description of a system." Furthermore, the author describes usage scenarios, such as "it is best to use ABM when the population is heterogeneous, when each individual is (potentially) different" (see section 3.2.1.3)—a scenario that specifically applies to the load-adapted geometry of building parts in lightweight design.

One of the challenges of ABM, which ultimately is also one of its benefits, is the difficulty of predicting the macro-level outcome of the model from the definition of its constituent parts at the micro-level, i.e. agents, rules, and interactions. The inverse problem, designing micro-level parts in order to achieve a specific macro-level goal is considered an even bigger challenge [205, p. 754]. Given these characteristics, the validity of the conceptual models underlying its parts, and the validation of their implementation play an exceptionally important role (see 3.3.3).

2.3 Research Questions and Objectives

Following the introductory considerations above, individual research questions, hypotheses, and objectives can be formulated with respect to conceptual models, implementation, and validation of ABMs.

2.3.1 Definition of Conceptual Models

The first research question is how conceptual models that underlie agent-based segmented shell design models can be validated thereby increasing the quality and 'trustworthiness' of agent-based models. In the field of ABMS, validation strategies of conceptual models are either based on collected data or on expert opinion [75, Sec. 3.13]). In architecture, and particularly in the field of shell design, data are

---

[2] Examples of 'individuals' in the architectural context are building elements, stakeholders in the design process, digital builders, and production machinery.

sparse and only few experts are available in early design stages. The proposed approach thus is a hybridization of these existing strategies and posits that expert knowledge is encapsulated in previously built segmented shells (the data) and can be derived and turned into actionable insights (the knowledge) for the design of new segmented shells. The corresponding *first research goal* therefore is to be able to validate conceptual models based on pre-validated design principles that are derived from the analysis of design, fabrication, and construction of built segmented shells.[3] These insights can then be further scrutinized by experts on the level of the individual design principle. This goal can be achieved through the following research objectives that are pursued with regards to conceptual models:

i. The development and demonstration of a systematic way of translating functional principles of built segmented shells into design principles through analysis of built segmented shells, abstraction of functional principles, and definition of design principles.
ii. The identification of possible feedback loops in the design process, which are to be implemented computationally.

2.3.2 Implementation of Conceptual Models

The second research question is how implementation of agent-based design models of segmented shells can be validated in order to increase transparency of conceptual models and reproducibility of outcomes. In the field of ABMS, validation of implementation is pursued through the use of standardized, ideally open-source, software frameworks that allow for transparency, reproducibility, and comparison of results and through a standardized documentation of the model, for example, using the 'ODD' (Overview, Design concepts, and Details) protocol proposed by Grimm et al. [70]. However, no such framework exists in the field of architecture at the time of writing, and existing ABMs usually are not documented in a standardized way. The *second research goal* therefore is to define and implement such a methodological framework, in which the previously defined principles can be implemented, reused, and documented, while responding to the specific disciplinary requirements of the field of architecture, such as Computer-Aided-Geometric-Design (CAGD), interdisciplinarity, and interoperability. This goal can be achieved by pursuing the following research objectives with regards to implementation:

iii. The development and description of a methodological framework for describing ABMs based on design principles.
iv. The development of a computational framework for ABMS as a platform

---

[3] In other fields such principles are sometimes called 'stylized facts' (see 9.2).

for experimentation, in which different segment types, jointing and segment arrangement strategies can be tested.

v. The ability to extend the software framework to other material systems and application scenarios (extensibility).
vi. The integration of the software framework into an interdisciplinary digital design environment.
vii. The testing of the framework through the implementation of functional prototypes and evaluation.
viii. The transparent description of the functional prototype.

2.4 Relevance and Scope

Research into fabrication-informed design of segmented shells within the ABM paradigm focuses on two crucial aspects of the architectural design process: *modeling* of segmented shells within a computational framework that allows feedback between generation of design variants and multi-disciplinary performance-based analysis; and *exploration* of the architectural, constructive, and performative potentials of segmented shells in the early planning stages.

The presented research is situated at the intersection of the fields of lightweight design, architectural design methodology, computer science as well as production technology. In the field of lightweight design, the focus is on the fabrication-informed design of segmented thin-shell structures. In the field of architectural design methodology, the focus is on an inductive, pattern-based approach for constructing agent-based architectural design models for segmented shells thereby addressing questions of the Digital Chain, Building Information Modeling, and design exploration. Within the field of computer science, agent-based modeling and simulation provides the basic method of the investigation. Finally, in terms of fabrication technology that is primarily considered in this research, the field of industrial robotics in architecture provides the production context, which informs the design of segmented shells.

Summarizing the relevance of the research from the previous considerations, the *segmentation* of thin shells promises to address challenges associated with the production of thin shells by enabling a high degree of prefabrication, which makes its associated benefits, such as quality control and just-in-time on-site delivery, accessible for their construction. *Lightweight design* promises to address challenges posed by the amount of energy embodied in and emissions associated with the construction of buildings. The development of *design methods* that are applicable in the early architectural design stages promises to address the challenge of design integration: making those design methods available to architects not only supports decision making where it has the largest impact, but also lowers the threshold to lightweight design in the first place.

Developing digital methods for architects that allow the anticipation of multi-disciplinary design requirements in the context of the design and fabrication of segmented shells is therefore expected to have a significant impact:

1. Bringing segmented shells within the scope of architectural design will activate the opportunities that lightweight design affords, such as resource effectiveness and novel architectural expression.
2. Focussing on design exploration and fabrication-informed design integration will ensure the relevance of the methods and processes developed within this research beyond the scope of thin-shell design.
3. Taking the characteristics of the established architectural planning process into account will ensure the relevance and applicability of the research findings.
4. The systemic shift from post-rationalization to pre-informed design is relevant beyond the scope of segmented shells: it will enable the realization of more performative building systems that are more geometrically adaptive, lightweight, and more resource-effective than existing building systems.
5. Conceptualizing robotic fabrication within the context of behavioral systems will allow for adaptive fabrication processes, which will become increasingly relevant as robotic fabrication is expanding from the workshop to the construction site.

2.5 Research Approach and Structure of the Document

The research builds upon the review of relevant literature and on previous work by the author, as outlined in section 5.2. The research thereby follows an inductive approach based on case studies: by analyzing specific cases, generally valid principles for the design of segmented shells can be abstracted and synthesized in an agent-based computational framework. The case studies described in part II constitute the core of the research and are framed, on the one hand, by the related work in the fields of ABM and shell structures described in part I and, on the other, by the implementation of selected design principles in the computational framework in part III (Fig. 2.3). Within this framework relevant design principles related to material, segment type, arrangement of segments, robotic fabrication, and assembly are conceptualized as agents and behaviors.

In the following two chapters of the first part of the dissertation, the related work to the research topic is presented in the form of literature reviews in the fields of agent-based modeling and segmented shells. The review of the field of ABMS (chapter 3) starts with an overview on the historical development of the method in relation

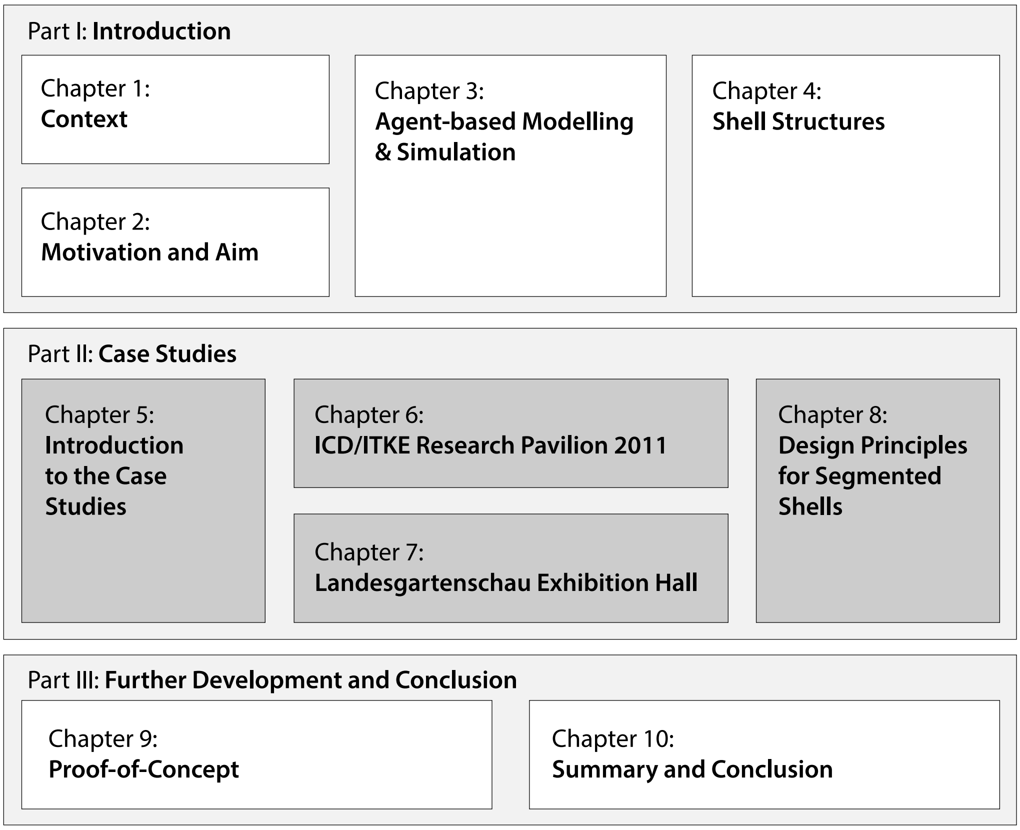


Figure 2.3: The structure of the dissertation.

to research into Artifical Life (ALife) and CAS on the one hand, and to Cybernetics and MAS on the other. Taking into account the wide-spread use of the method and diverging terminology, a characterization of the method is presented followed by an overview on state-of-the-art applications of ABMS in different research areas. In the next section, a more detailed look into the use of ABM in architecture is provided including a categorization of its usage scenarios. The chapter concludes with a discussion of further research questions and challenges related to the main opportunities that are provided by self-organization and emergence for architecture.

The review of the field of shell structures (chapter 4) starts with an overview on the history of man-made vaulted spaces from pre-historic and classical times to the 20th century, describing the expansion of form-defining methods from purely geometric methods, to physical, mathematical/analytical, and numerical methods. A particular focus is placed in this chapter on describing the state-of-the-art in numerical form-finding methods, which are applicable in early stage architectural design. This is followed by a closer look at related work regarding segmentation, in particular at the so-called ‘plate-lattice dualism.’ The chapter concludes with an overview on current research avenues regarding the segmentation of shell structures and further elaborates on the relevance of design integration for early stage architectural design.

The second part of the dissertation is dedicated to the analysis of built segmented shells as part of the case studies. Chapter 5 “Introduction to the Case Studies” starts with a definition of the term ‘case study’ and the reasoning behind the selection of the cases. The following section elaborates on the term ‘principle’ and provides a definition of the terms ‘functional principle’ and ‘design principle.’ After describing the aim and scope of the cases selected for this study, their shared background and context are described including the extent to which they share goals and objectives.

The two case studies, which ensue in chapter 6 and 7, follow a similar pattern starting with an introduction, which describes the particular relevance of the selected case, followed by a section of its specific background and objectives. The third section in each case study provides a catalogue of individual functional principles that have been identified in the literature related to the case. Principles are categorized according to the relevant development domains of joint and segment development, arrangement of segments, structural development, global design development, fabrication development, off-site pre-fabrication, and on-site assembly and construction. This catalogue of principles constitutes the first level of abstraction and forms the fundamental part of the research. It establishes the data set, upon which subsequent research development is built. The ensuing fourth section of the case studies describes a second level of abstraction in the case studies, in which chains of functional principles within the defined development domains are established. The fifth section of the case studies provides an overview of the identified principles and their connections within and between domains. This forms the basis for the final step of the analysis on the third level of abstraction, showing case-specific cross-domain integration and feedback loops. The chapters conclude with a discussion of each case study’s findings.

Chapter 8 finally establishes “Design principles for segmented shells.” Starting with the definition of theoretical design principles based on the functional principles identified in both case studies, design principles are then juxtaposed according to the design domains (called development domains in the analysis stage) in which they occur. This provides the basis for recommendations regarding the last step of the proposed methodology: the transfer, that is, the mapping from design principles to relevant constructs of the ABM paradigm. As part of the results section, the methodology of the analysis process is summarized, followed by a discussion of the findings and general conclusions from the case studies.

The first chapter in the third part of the dissertation (Chapter 9), focuses on the further development and implementation of the research findings within ICD’s agent-based modeling framework. The chapter first summarizes established modeling approaches for ABMs and briefly introduces the framework. Then, selected design principles are implemented in the software framework as a proof of concept in the form of a problem-oriented case study. Of particular interest is a novel modeling strategy for segmented plate structures that obviates the need for a ‘design surface’. This way, the formation process of the plates can directly affect the form-finding of the shell, which was not possible in previous work. Structured as a series of experiments with increasing complexity, the case study addresses previously identified challenges of designing segmented shells and presents novel findings particularly with respect to parameter optimization and autonomous learning of agent behaviors using Reinforcement Learning.

The last chapter of this dissertation summarizes and discusses the findings of the research and provides an outlook towards further research.

3

Agent-based Modeling and Simulation

3.1 Historical Perspective

In the past three decades, ABMS has become an increasingly popular approach to modeling and understanding the complex, nonlinear systems that can be observed in the world [74, p. 163]. Its microscopic perspective, however, is still a relatively novel approach and constitutes an alternative to classical macroscopic simulation approaches, such as System Dynamics (SD) or Discrete-Event Simulation (DES) [124, p. 1]. The fact that ABMS is novel is reflected in different views in literature about its role and state-of-development. Metz [132, p. 12], in his overview on ABMS in the context of social sciences, states that it has consolidated as a field in the 1990s, whereas Heath, Hill and Ciarallo [75] in their review of 279 articles about research using ABMS from 1998 to 2008 state that it still hadn't consolidated in the 2000s, but would still need to in order to activate its full potential [75, Sec. 5.3]. Finally, Bonabeau [23, p. 7280] posits that ABMS is more a mindset than a technology—the mindset being the description of a system from the perspective of its constituent units and their interactions as opposed from the top down.

Given these heterogeneous views, the purpose of this first section is to illustrate and retrace the different lines of development and to identify the protagonists that have contributed to the main concepts in ABMS that are now used across a variety of fields.

3.1.1 Beginnings

Even though ABMS relied on the invention of computers as a precondition for numerical simulation, its origins can be traced back further to initial attempts at explaining complex phenomena that could not be explained using prevailing theory [74, p. 163]. Examples include the phenomenon that a stable economy can emerge even though every participant tries to maximize their own profit, which Adam Smith termed 'Invisible Hand' (ibid.); the phenomenon that the relatively simple rules and building blocks of Darwinian Evolution can lead to the abundance and diversity of life, which Richard Dawkins' termed 'Blind Watchmaking' (ibid.) contrasting the term to the watchmaker analogy proposed by William Payley (1802); the population dynamics of two animal species (predators and prey), for which Volterra proposed a differential equation-based model in 1926 [133, p. 8]; or the observation by G.H Lewes in 1875 that, given the compounds produced by a chemical reaction, "we cannot always trace the steps of the process, so as to see in the product the mode of operation of each factor", which he termed 'emergent' as an alternative to the term 'resultant' [47, p. 97].

Prevailing theory, which was based on Newtonian Philosophy, posited that given an approximate knowledge of a system's initial condition and an understanding of natural law, one could calculate the approximate future behavior of the system. Still, it was known that the corresponding reductionist approach of reducing a system to its parts in order to explain the functioning of the whole system, was not applicable to dynamic systems that involved many individual interacting entities [74, p. 163-64]. The movement known as 'emergent evolutionism' or 'proto-emergentism' thus used the concept of 'emergence' as an argument against reductionism, while having few answers when it came to understanding how emergence itself was possible, that is, what the mechanisms were that lead to emergence [47, p. 97].

The characteristic feature of the observed systems is that they are 'nonlinear', meaning, in mathematical terms, that a multiplicative change in an input variable of an equation system, for example by a factor α, does not imply a multiplicative or proportional change of its output by α [19, Sec. 1.2.2]. In the context of complexity theory, this property is described as "the sum output of the whole system does not equal the sum output of the constituent parts" [74, p. 164]. More commonly, this is known as 'the whole being more than the sum of its parts'. In other words, the behavior of these systems is not predictable from the point of view of the constituting elements or, from a reductionist point of view, that the observed phenomena are not reducible to the parts that constitute the system. This nonlinear property is a necessary, but not sufficient feature of 'emergence' (see Section 3.2.5 for a closer look at emergence).

The relevance and increasing popularity of ABMS over the last three decades is due to its ability to model, simulate, and examine complex, nonlinear systems and, at least for the moment, it is the most suitable approach for analyzing these types of systems [75, Sec. 1.2]. This means that for the longest time in scientific history, scientists essentially lacked tools to study and comprehend nonlinear systems, despite observing and theorizing about their 'emergent' behavior [74, p. 164]. This situation changed with the invention of the computer and the associated theoretical concepts and it is no coincidence that the development of computing has been significantly shaped by the building of computer models for simulating natural processes [59, p. 13].

3.1.2 Automata and the invention of the computer

In 1936, Alan Turing (1912-1954) laid the theoretical foundation for building machines that would be able to recreate the nonlinear systems observed in nature with his invention of the universal Turing Machine: "an abstract digital computing machine consisting of a limitless memory and a scanner that moves back and forth through the memory, symbol by symbol, reading what it finds and writing further symbols. [...] implicit in it is the possibility of the machine operating on and modifying its own program" [43]. Turing said, "What we want is a machine that can learn from experience," for which "the possibility of letting the machine alter its own instructions provides the mechanism" [193, p. 393]. The universal Turing machine is the fundamental concept for digitally replicating any mathematical process, which would allow machines to represent systems [74, p. 164]. The Church-Turing Hypothesis extended this theory by stating that a machine could duplicate not only the functions of mathematics, but also the functions of nature (ibid.).

John von Neumann (1903-1957), another main contributor to the invention of computers, set out to prove Turing's theorem by building a self-replicating machine. His key insight was to focus not on physically engineering a machine, but on passing information [74, p. 164]. Together with Stanislaw Ulam (1909-1984), von Neumann used Automata, deterministic self-operating entities programmed to follow a set of rules, as a way to prove that machines can replicate themselves and the information needed for self-replication. When arranged adjacent to each other in a two-dimensional grid, they can form distinct patterns of information that are able to replicate themselves on the grid [12, p. 154].

Von Neumann saw computers as having the ability to "break the present stalemate created by the failure of the purely analytical approach to nonlinear problems" by giving scientists the ability to heuristically use the computer to develop theories [74, p. 164]. A heuristic being any rule-of-thumb principle that cuts down the amount of searching required in order to find a solution to a problem in the space of possible solutions [43]. Heath [74, p. 164] states that the heuristic use of computers, as viewed by Von Neumann and Ulam, is very much like the traditional scientific method except that the computer replaces or supplements the experimentation process. The essential idea being the understanding that the computer serves as a simulation of the real system (ibid.).

Following the invention of the digital computer, two parallel and highly interconnected lines of development can be observed that led to the development of ABMS: on the one hand, research summarized under the term 'Complexity Theory' studied self-organization and emergence in physical and social systems forming the conceptual framework for ALife and CAS; on the other hand, research summarized under the term 'Systems Theory' focused on communication and feedback as "unifying principles" [200, p. 51] in technology as well as biology and formed the conceptual framework for the fields of Cybernetics, Robotics and Multi-Agent-Systems.

3.1.3 Complexity Theory and Artificial Life

3.1.3.1 Cellular Automata

The first line of development was led by von Neumann's theory of self-reproducing automata and had the objective of synthesizing natural systems. Referencing von Neumann and Langton (see below), Heath [74, p. 164] describes the relevance of Cellular Automata (CAs) for the study of complex systems: based on the recognition that "many natural systems are parallel", the relevance of CAs is that a CA is a genuinely parallel system composed of many individual cells where "each cell can make autonomous decisions simultaneously with other cells in the system." Another significant aspect of CA systems is that their global behavior is the result of interactions between the many locally controlled cells in the system. This requires "engineering a cell's logic at the local level in hopes that it will create the desired global behavior" [74, p. 165].

In the late 1960s, the mathematician John H. Conway (1937-2020), at the time a at Cambridge University, devised a kind of chessboard CA game, which he provocatively called the *Game of Life*. It was a crude model of how living cells or organisms proliferate [12, p. 154]. Using this 'game', it was found that depending upon the starting configuration and the local cell rules, certain shapes or patterns such as the famous 'glider' would emerge and begin to move across the board where it might encounter other shapes and create new ones as if mimicking a very crude form of evolution [74, p. 165]. Conway's Game of Life is now considered a prototype for research on ALife because the surprising richness of the patterns arises from a few simple rules about local interactions between cells [12, pp. 154-55]. According to Macal [120, p. 116], it is also the simplest way to illustrate some of the basic ideas of agent-based modeling.

Further research by Stephen Wolfram (⋆1959) in the 1980s on the properties and potential uses of one-dimensional CA led to the classification of CA systems according to their long-term behavior and sensitivity to their initial state into four classes [74, p. 165] (Table 3.1).

Table 3.1: Classification of CA systems according to Wolfram [211].

Class	Behavior	Sensitivity
1	Homogenous state	Insensitive
2	Simple stable or periodic structures	Locally sensitive
3	Chaotic (non-repeating) pattern	Highly sensitive
4	Complex localized structures	Non-linear relationship between initial state and outcome

The term "life at the edge of chaos", commonly attributed to Christopher G. Langton (⋆1948), describes the idea that Class 4 systems are situated in a thin region between Class 2 and Class 3 systems [120, p. 118]. Life, or synthesized life, thus is characterized as being situated between order and chaos, far-from-equilibrium, with positive and negative feedback cycles. Langton, who coined the term "Artificial Life" in the late 1980s, describes ALife as

> "a new discipline that studies 'natural' life by attempting to recreate biological phenomena from scratch within computers and other 'artificial' media. ALife complements the analytic approach of traditional biology with a synthetic approach in which, rather than studying biological phenomena by taking apart living organisms to see how they work, one attempts to put together systems that behave like living organisms" [110, p. 25].

Research into CAs would lead to the bottom-up approach mainly employed in the field of ALife because they exhibit the same nonlinear relationship between global behavior and local rules that can be observed in complex natural systems [74, p. 165].

Chris Langton and others at the Santa Fe Institute, the pioneering body for the study of ALife, would later release the first agent-based modeling toolkit *Swarm* [121, p. 146] (see also [134]). The aim of Swarm was to provide "standardized equipment" that would allow repeatability of results in the face of various custom built software in the community that made comparing results difficult. As a consequence, many of the early agent-based models were developed in ALife research using the library-based Swarm modeling toolkit [124, p. 152].

Macal [120, p. 114] thus traces the origins of ABMS to the field of ALife with ABMS having evolved as the "computational arm" of ALife. The relationship between ALife and ABMS, however, is complex: on the one hand, the emergence of ALife as a field was essential to the development of ABMS; on the other, the possibility for creating agent-based models was essential to making ALife a promising and productive endeavor. In ALife terminology, one could say that ALife and ABMS have co-evolved to their present states (ibid., p. 115).

The finding of the early experiments with CAs that simple rules can lead to complex and unexpected emergent behavior, was soon picked up by the social sciences field with the aim of modeling artificial social systems. The field so far lacked methods to describe the mechanisms that, despite the many nonlinear interactions, lead to organization and structure in society but also to unforeseen effects such as urban segregation.

3.1.3.2 Social Sciences

With the concept of individual actors being a well-established construct in social sciences, James M. Sakoda (1916-2005) formulated one of the first social agent-based models in 1971, the *Checkerboard Model*, which was based on an analogue version of CA [120, p. 125]. Using a similar approach, Thomas C. Schelling (1921-2016) subsequently developed a digital model of housing segregation in which agents represent homeowners and neighbors, and agent interactions represent agents' perceptions of their neighbors (ibid.). The model, which today is just called the *Schelling Model* [76, p. 1], showed that housing segregation patterns can emerge that are not necessarily implied or consistent with the objectives of the individual agents [124, p. 156].

Another early social simulation based on CAs is Axelrod's *Tit-for-Tat Model*. Robert M. Axelrod (⋆1943), who was one of the founding members of the BACH group at the University of Michigan (originally consisting of Arthur Burks, Bob Axelrod, Michael Cohen and John Holland), investigated the emergence of cooperation. Agents on the grid interacted in local neighborhoods employing a variety of different strategies. Macal [120, p. 125] reports that a simple 'Tit-For-Tat' strategy of reciprocal behavior toward individuals was enough to establish sustainable co-operative behavior.

While these early models were still based on CAs using grid cells and stationary agents as primary constructs, improving technology would lead to systems where agents could travel across the grid. As an example, Heath and Hill [74, pp. 165-66] mention the *Sugarscape Model* of artificial societies by Joshua M. Epstein and Robert Axtell from 1996—another groundbreaking development in the history of ABMS [120, p. 126]. In the Sugarscape Model, agents interact with other agents and their environment as they move around the grid, which allows them to access environmental variables, extract resources, etc., based on location. Macal (ibid.) reports that in numerous computational experiments, Sugarscape agents emerged with a variety of characteristics and behaviors, highly suggestive of a realistic, although rudimentary and abstract, society.

Emerging from these studies was the field of *Generative Social Sciences* with the goal of modeling social processes as the emergent result of social interactions [120, p. 126]. According to Macal (ibid.), Epstein even argued that "social processes are not fully understood unless one is able to theorize how they work at a deep level and have social processes emerge as part of a computational model."

With discoveries about synthesizing complex systems and emergent behavior, the study of social systems extended to other fields, such as the fields of ecology, biology, economics, and other social sciences, where scientists began using CAs to model systems consisting of many autonomous, interacting entities that were traditionally very hard to study due to their nonlinearity [74, p. 165].

3.1.3.3 Complex Adaptive Systems (CAS)

While research so far was focused on identifying local rules that lead to a particular observed system behavior, subsequent research also investigated systems that changed behavior and, correspondingly, their underlying local rules. The field of CAS draws much of its of inspiration from biological systems and is primarily concerned with how complex adaptive behavior emerges in nature from the interactions among autonomous agents [74, p. 168] (see also [119, p. 75]). An important protagonist in this domain was John Holland (1929-2015) who is best known for his invention of *Genetic Algorithms (GAs)*, a family of adaptive search and optimization methods inspired by biological evolution. His main motivation, however, was to develop an interdisciplinary theory of adaptation, one that would inform biology as much as computer science [57, p. 60].

According to Heath and Hill [74, p. 168], Holland's main contributions to the field, and in turn ABMS, was his identification of the main mechanisms (Tagging, Internal models, Building blocks) and properties that compose all CAS, which are: (1) Aggregation - CAS are hierarchically structured and composed of groups and subgroups that have a similar level of abstraction; (2) Nonlinearity - the behavior of the system is the result of dynamic feedback and interactions; (3) Flow - communication and flow of information is necessary; and (4) Diversity - CAS are composed of heterogeneous agents that allow for new interactions and adaptations to develop [81, p. 38]. These terms have provided a heuristic for defining and designing ABMs as they are known today because Holland categorized many of the properties of complex systems allowing for better focus, development, and research [74, p. 168].

One of Holland's main postulates was that adaptive systems create and use 'internal models' of their environments to make predictions in order to cope with "perpetual novelty." These models can be "tacit and learned over evolutionary time" or "explicit and learned over a single lifespan" as in the case of cognitive systems that form internal representations through learning [57, p. 60].

A fundamental concept for studying and modeling of CAS is that complex systems can be represented sufficiently with a simpler model, often called a Homomorphic Model [74, p. 167]. Models that form valid homomorphisms with the part of the world they represent allow making accurate predictions. Holland's emphasis on homomorphisms, as a formal way to evaluate model validity, is an idea that dates back to W. Ross Ashby's 'An Introduction to Cybernetics' [57, p. 60]. Homomorphism is a fundamental concept in ABMS as the model is an abstraction of the system that is being modeled. The validity of the abstraction can thus be judged by how well the model represents behaviors of the real system. The process of forming a valid model can use inductive methods of learning and adaptation. Heath and Hill [74, p. 166] refer to fractal hierarchy, which can be observed in chaotic systems, as an explanation for why Homomorphic models can be effective at modeling complex systems: "hierarchical systems are composed of subsystems such that the subsystems can be represented not as many individual entities but as a single entity."

Another important concept that Holland championed was the idea of *exploratory modeling* as a way to provide insight into the basic principles and mechanisms of a complex system [57, p. 61]. Exploratory models were not primarily intended to provide detailed, domain-specific predictions, but rather to show how certain behaviors could be produced. The insights provided by exploratory modeling might then be used to form more specific, detailed models (ibid., see also 3.2.9).

In order to study CAS, Holland pioneered the style of modeling that has come to be known as 'individual-based' or 'agent-based' modeling. The concepts championed by Holland were formalized into an executable computational framework, the *Echo* toolkit, which was able to produce well-known patterns from nature and thus appealed to immunologists, economists, and evolutionary biologists alike [57, p. 62]. Rather than characterizing the patterns as would be the focus of a phenomenological approach, Holland's focus was understanding the mechanisms by which complex patterns emerge and change [57, p. 63].

ABMS thus provided the missing tool to study emergence in CAS: to test hypotheses, to enhance the understanding of underlying mechanisms and, ultimately, to build a theory of complexity. 'Complexity theory' (also called 'Neo-emergence' [47, p. 97]), equipped with the tools to model emergence, thus superseded 'proto-emergentism' (see Section 3.1.1).

3.1.4 Cybernetics and AI

The second line of development was led by research into the role of feedback in the control of machines starting in the 1940s in the context of anti-aircraft firing systems in World War II [74, p. 166]. Realizing that the same principles found in the control of machines were also true for animals, Norbert Wiener (1894-1964) named the nascent field in his seminal 1948 publication "Cybernetics: Or Control and Communication in the Animal and the Machine" [31, p. 156]. Even though the origin of Cybernetics is arguably quite distinct from Complexity research, they essentially both share the same research subject - understanding complex, self-regulating systems.

The research questions, however, that both fields asked when studying complex systems were different: while researchers in CAS asked questions about the mechanisms that underlie self-organization and emergence in complex natural systems, research in Cybernetics focused more on the question "what complex systems do" with the objective of instrumentalizing complex systems in technical applications [74, p. 166]. These questions led to two different, but related, paths toward discovering the nature of complexity (ibid.).

Cyberneticians built mathematical and mechanical models to investigate how animals work at both the neural and behavioral levels and to explore how machines interacted with their environments [31, p. 156]. The findings of Cybernetics essentially formed the basis for the fields of robotics and 'bottom-up' Artificial Intelligence (AI).

One of the important findings about complex systems stemming from the field of Cybernetics is the role that feedback plays in the patterns and properties of complex systems. W. Ross Ashby (1903-1972), a psychiatrist by training and one of the pioneers in Cybernetics, observed that systems will exhibit different patterns depending upon the type of feedback found in the system: with negative feedback, the system will reach a fixed state; with zero feedback, patterns will remain constant or be periodic; and with positive feedback, patterns will grow indefinitely and possibly out of control [74, p. 166]. An important realization is that in complex systems different types of feedback can coexist leading to a dynamic, but stable state linking Systems Theory to Chaos Theory and Thermodynamics.

Civil applications for Cybernetics were consecutively developed ranging from architecture to management, such as Cedric Price's "Fun Palace" in collaboration with Gordon Pask or Stafford Beer's development of a nationwide data network in Chile, called 'Cybersyn', to monitor and control factory operations [31, p. 156].

3.1.4.1 Vehicles

Part of the research into the control of machines and, similar to Ashby, into the functioning of the brain, is the work by W. Grey Walter (1910-1977), a neurophysiologist by training, who was instrumental in setting up the field of Cybernetics [31, p. 156]. He is most known for his development of and experiments with some of the first electronic autonomous robots in the late 1940s, which he called ‘tortoises’. The main hypothesis in his experiments with “synthetic life” was that a small number of ‘brain cells’ and simple rules could lead to life-like behavior. The light-sensitive robots exhibited “positive tropism”, such as a moth flying towards a candle resulting from the attraction towards the light source, but also repulsion if the robot gets too close to the light source [202, p. 44].

The design philosophy of these simple analogue autonomous robots is reflected in the thought experiments by Valentino Braitenberg (1926-2011) that are documented in his book “Vehicles: Experiments in Synthetic Psychology” (1984). Braitenberg, a neuroscientist and former director of the Max Planck Institute for Biological Cybernetics in Tübingen, explored psychological ideas and the nature of intelligence through the development of increasingly complex robot behaviors based on simple circuits [135, p. 2].

The term ‘Braitenberg vehicle’ is still commonly used in robotics today to describe a robot where the sensor readings are directly mapped to actuators. An example would be an obstacle avoiding robot, where the left ultrasonic sensor is mapped to the left motor and the right sensor is mapped to the right motor. The closer the robot senses an object on its right sensor, the higher the sensor value respectively right motor speed, which results in a turning behavior away from the obstacle [155, p. 5].

Inspired by Braitenberg, Craig Reynolds ($\star$1953) later used the term ‘vehicle’ in the context of the computer graphics to describe autonomous characters capable of locomotion [158, p. 5]. Reynolds’ boids model (the word being a fusion of ‘bird’ and ‘droid’ [12, p. 153]) simulates the flocking behavior of birds using simple steering rules of attraction and repulsion that are observed at the level of the individual boid [156, p. 28]. The vehicles have a limited ability to perceive their environment (limited knowledge about the whole system), and adjust their movements according to those pre-defined steering rules resulting in Collision Avoidance (Separation), Velocity Matching (Alignment), and Flock Centering (Cohesion) behaviors [156, p. 28]. In 1999, Reynolds [158] expanded on those initial behaviors adding lower-level behaviors, such as ‘seek’, ‘arrival’, and ‘wandering’, and other higher-level behaviors, such as ‘path-following’ and ‘containment’. Besides vector-based locomotion (as opposed to grid-based), the main innovation of Reynolds’s model is the definition of local steering rules that provide the agents with improvisational, life-like behaviors [158, p. 2] and, on aggregate, lead to the non-linear behavior of the flock. The model is in many respects a seminal development effectively merging parallel developments in Cybernetics and ALife: from the point of view of Cybernetics it is a virtualization of Braitenberg vehicles that allows exploring aggregate behaviors of robotic swarms; from the point of view of ALife it is, first of all, a model that provides insight into the mechanisms that underlie the emergent behavior of real swarms; but, enabled by advancing technology, it is also a significant further development in terms of agent environments as it frees the agents from the restrictions of grid-based, discrete environments.

Despite resembling AI, in absence of sophisticated learning strategies or a cognitive state of the agents, Reynolds relates his behavioral simulation to the field of ALife [157, p. 2]. From Ashby's point of view, however, the lack of sophisticated strategies does not preclude the emergence of intelligent behavior, as the focus is on the richness of links between the cells or, in CAS, on the interactions between the agents. The answer to the question of "what is artificial intelligence" therefore mostly depends on the definition of intelligence: can the self-organization and emergence that some biological systems, such as eusocial systems, exhibit be called intelligent? Can an elephant be considered intelligent even when it doesn't play chess [30, p. 10]?

3.1.4.2 Behavioral intelligence and robotics

In the 1980s, Rodney Brooks (⋆1954) emerged as a critic of the then established approach towards AI, which was based on the concept of a symbolic representation of the 'world'. Historically, AI research was mostly concerned with human-centered intelligence, such as playing chess, and had ended up in a kind of deadlock with very narrow applications, expert systems, and only incremental progress that made the initial goal of achieving artificial general intelligence seem out of reach. Brooks [30, p. 1] then argued that classical AI was fundamentally flawed, because it had rarely achieved grounding in physical reality and thus was unable to cope with unstructured and uncertain environments.

Inspired by Walter's artificial creatures that move around in the real world [31, p. 156], Brooks became one of the early proponents of an alternative, reactive approach to Artificial intelligence, initially called *nouvelle AI*, *fundamentalist AI*, or in a weaker form *situated activity*, which was based on the physical grounding hypothesis [30, p. 1].

According to Heath and Hill [74, p. 164], the problem with classical AI was the top-down approach towards understanding intelligence: "With reductionism still being the prevalent scientific methodology employed, and perhaps spurred on by the idea of powerful serial computing capabilities, many scientists began trying to synthesize systems from the top down," which is to "take the global behavior, decompose it into small pieces, understand those pieces, and then put them back together to reproduce or predict future global behavior."

The physical grounding approach, on the other hand, is not based on abstractions of the real world, rather the world becomes the model itself. According to Brooks [30, p. 3], “the world is its own best model. It is always exactly up to date. It always contains every detail there is to be known. The trick is to sense it appropriately and often enough.” In his seminal paper “Elephants don’t play chess”, the title being a riposte to critics of the new approach, Brooks juxtaposes the classical symbol hypothesis (being based on decomposition of intelligence into functional information processing modules from the top down) with the physical grounding hypothesis (being based on individual behavior generating modules, from which complex behaviors emerge through coexistence and co-operation from the bottom up) [30, p. 1].

From this point of view, Reynolds’ work can be understood as also being part of the new, bottom-up, reactive approach towards artificial intelligence. Similar to Reynolds, who proposed “prioritization” of behaviors when combining behaviors (to address the problem of vector addition of individual, potentially conflicting behaviors that cancel each other out) [156, p. 29], Brooks proposed a computational architecture, called “subsumption architecture,” for developing physically grounded systems based on incremental layers [29, p. 16].

By 1990, Brooks had built “a family” of physical robots of varying complexity based on the subsumption architecture that share some of the characteristics of Reynolds boids, such as low-level repulsion behaviors, and collision avoidance. For example, in the case of “Squirt,” one of the smallest robots weighing only 50 grams, high level behavior emerges from a set of simple interactions with the world [30, p. 7]. For Brooks [30, p. 10], “nouvelle AI relies on the emergence of more global behavior from the interaction of smaller behavioral units. As with heuristics there is no a priori guarantee that this will always work. However, careful design of simple behaviors and their interactions can often produce systems with useful and interesting emergent properties.”

Langton [110, p. 39] summarized the key priorities of this new approach to robotics pursued by Brooks and his colleagues at MIT:

- no traditional notion of planning is used;
- no central representation is needed;
- notions of world modeling are impractical and unnecessary;
- biology and evolution are good models to follow.

In this view of intelligence, which is driven from the bottom-up rather than from the top-down, higher level intelligence is seen to have arisen as control structures on top of pre-existing simple behaviors, giving rise to higher-order behaviors, over which even higher-order control structures emerged [110, p. 40]. This bottom-up view not only relates the robotics and computer graphics experiments by Brooks and Reynolds back to Braitenberg, but, according to Langton (ibid.), this view also has the advantage that it is consistent with the "probable evolutionary development of higher intelligence".

3.1.5 Complexity and Systems Theory in Architecture

Systems Theory and Cybernetics, as well as Complexity Theory and CAS research were received relatively early in architecture and design theory.

Examples for the dissemination of Cybernetics include Norbert Wiener's lecture at the Hochschule for Gestaltung in Ulm in 1955 [52, p. 9], Rittels work on Cybernetics in design "Kommunikationstheorie in der Soziologie (Kybernetik)" in 1958, Cedric Price's 1964 "Fun Palace" project, Chris Abel's and Gordon Pask's articles in AD magazine in 1969 titled "Urban Chaos or Self-Organization" and "The Architectural Relevance of Cybernetics", respectively, or Nicholas Negroponte's treatises "The Architecture Machine" (1970) and "Soft architecture machines" (1975).

Examples for the dissemination of CAS include Manuel DeLanda's article "Nonorganic Life" (1992), Sanford Kwinter's article "Soft Systems" (1993) and the work summarized in John Frazer's seminal 1995 book "An Evolutionary Architecture" [59].

In this context, a parallel between research into ALife and the development of computation in architecture is worth noting: while John Frazer (⋆1945) was working on his evolutionary structural system 'Reptile' in the mathematical laboratory at Cambridge in the late 1960s, he shared the use of the computer-graphics system with John Conway who was developing the seminal Game of Life CA [59, p. 55]. Through John Frazer then, research topics relating to CAS and ALife such as emergence and self-organization, parallel systems, and adaptation, particularly using evolutionary computation influenced by John Holland, were reflected early on in architectural design research.

3.2 Characterization and Disambiguation

ABMS has emerged from research into Complexity Theory and System Theory as the most 'natural' way to implement relevant concepts, that is, from the perspective of individual agents. Starting from the theory of computing machines (CA), synthesis and analysis of natural systems (ALife), and complex adaptive systems (CAS) on the one hand, and Cybernetics, behavior-based artificial intelligence, and robotics, on the other, it is clear that many fields contributed to the development of ABMS.

The areas of application are similarly diverse comprising biology and ecology, social sciences and economics, business and management, traffic and supply chains just to name a few. The field of ABMS is thus multidisciplinary insofar as it is used in many disciplines but it is not particularly interdisciplinary, as disciplines developed their own ABMS terminology to describe techniques, applications and results, have their own ABMS standards and their own ABMS philosophies [75, p. 16]. This makes comparison and communication difficult and, thus, the purpose of this section is to establish a clear terminology and characterization of the terms that are used in this research.

3.2.1 Comparison to other simulation approaches

ABMS has been applied to the study of complex natural and artificial multi-agent systems such as flocks of birds, schools of fish, colonies of ants, termites, bees or wasps, but also crowd stampedes, traffic, supply chains, stock markets and robot swarms, all of which can exhibit emergent properties and self-organization even though none of the involved entities have an overview of the global system.

There are three main modeling and simulation paradigms for complex systems that have been extensively discussed in literature: SD, DES, and ABMS [17, p. 413:6]. Despite substantial overlap, there are many aspects of ABMS that differ from DES and SD, including applicable problem domains, disciplines, and the underpinnings of its computational implementation [123, p. 1].

Behdani [17, p. 413:1] points out that the selection of a modeling and simulation paradigm is also part of the conceptualization process, since “each of these paradigms comes along with a set of (implicit or explicit) assumptions regarding the key aspects of the world” [17, p. 413:6]. The purpose of this section is therefore to highlight similarities and differences between the assumptions inherent in those approaches.

3.2.1.1 System Dynamics

System Dynamics is considered a top-down approach to modeling complex systems, in which a system is broken down into its constituent level components [121, p. 147]. The components are then described in mathematical terms using a phenomenological approach by focusing on “system observables and modeling the system components with aggregated state variables” [17, p. 413:8]. While the kinds of problems for which SD is applicable are similar to the kinds of problems for which ABMS is applicable, the inherent lower level abstraction is easier to understand in ABMs and easier to communicate to stakeholders [179, p. 206]. The basic constructs of SD are feedback loops, where the existence of multiple feedback loops in a system is considered the driver of the dynamic behavior of the system [17, p. 413:9]. “Diagrams of information feedback loops and circular causality are the tools for conceptualizing the structure of a complex system” [17, p. 413:6]. Causal loops can then be “transformed to a stock-flow diagram, which consists of two fundamental types of variables: Stocks (or levels) and Flows (or Rates)" (ibid.). The rate of change between 'stocks' in the system is modeled using a differential equation-based approach. SD thereby assumes "an 'average individual' which represents a population of entities [...] it has an aggregate view of the interactions in the system and it assumes perfect mixing within compartments of the system" [17, pp. 413:8–9]. ABMS, on the other hand, is an individual-oriented, bottom-up modeling approach, as stated before. According to Bonabeau [23, p. 7280], this does not, however, preclude differential equation modeling since "a set of differential equations, each describing the dynamics of each of the system's constituent units, is an agent-based model." Bonabeau [23] gives two examples of comparing the results of SD to ABMS: one from simulating the impact of changing tick sizes on the stock market, and one example from simulating dissipation as in adoption of a product. In both cases the results of ABMS significantly differ from SD, which not only shows "how useful ABM is when dealing with inhomogeneous populations and interaction networks" [23, p. 7287], but also the impact of 'downward causation' [120, p. 113]. Specifically, SD does not capture the bi-directional link between macro-level and micro-level, which is a characteristic feature of CAS [47, p. 5], which has led several authors to challenge the capability of SD to produce emergent properties in a complex system [17, p. 413:10].

3.2.1.2 Discrete Event Simulation

Whereas ABMS is considered a individual-oriented, bottom-up approach and SD is considered a system-oriented, top-down approach, DES is considered a process-oriented approach toward modeling and simulation of complex systems [17, p. 413:9]. Whether it is a 'top-down' or 'bottom-up' approach toward describing a system, there seems to be some contradiction in the literature: according to Behdani [17, p. 413:8], both DES and ABM share "a 'bottom-up' perspective in modeling; they start with a detailed representation of individual parts of the system and their interactions." However, Siebers et al. [179, p. 207] state that DES follows a top-down modeling approach where the focus is on "modeling the system in detail not the entities." Furthermore, macro-level behavior emerges from micro-decision of the agents in ABMS, whereas in DES the "macro behavior is modeled" by the programmer (ibid.). Nevertheless, DES shares many features with ABMS, which leads some researchers to claim that ABMS is just a subset of DES [121, p. 144].

The ambiguity as to how ABMS compares to DES might also be due to historical reasons. First of all, ABM is younger than DES, which has been the "mainstay of the Operational Research (OR) simulation community for over 40 years" [179, p. 204] and it can be applied to the same problems in OR; second, agent-based models have been implemented in DES software packages; and third, part of the confusion might also be due to the fact that the original ABMS toolkit *Swarm* called itself a “multi-agent discrete event simulation” standing “in contrast to continuous [differential equation-based] simulations” [134, p. 4]. While the discreteness property certainly applies to the discrete spaces of grid-based agent models, as the dominant topology of early ABM, and to the advancing of time in discrete steps, the DES point of view captures only a subset of ABMs, which can also represent continuous time and space (see Section 3.2.4).

The basic constructs of DES, as the name suggests, are sequences of distinct events as a way to approximate complex real-world systems and processes. Systems possess at any point in time a state whose change is triggered by discrete events [17, p. 413:7]. Time is thus simulated through the sequence of events with no changes to the system occurring between events. An example is simulating the operation of a warehouse with orders coming in and inventory going out and being replenished: plotting the number of inventory against time results in a step function, that is, a set of flat line segments with breaks between them [127, p. 3]. This is in contrast to simulation of systems with continuous events, such as in weather systems, where temperature plotted against time forms a continuous curve (ibid.).

In addition to the top-down vs. bottom-up world view, the following main differences between DES and ABM can be summarized from Behdani [17] and Siebers et al. [179]:

1. DES assumes one centralized thread of control whereas in ABMs control is decentralized with each agent having its own thread of control.
2. Entities in DES are considered passive, that is something is done to the entities while they move through the system, while intelligence (e.g. decision making) is modeled as part of the system. In ABMS, entities are considered active, that is, they can take on the initiative to do something, with intelligence being represented within each individual entity.
3. In DES entities are static, whereas in ABMs agents can be dynamic including adaptive behavior, the ability to learn, and have dynamic interaction topology.
4. In contrast to DES, ABMs can represent spatial aspects, which is important in certain applications, for example in ecology and urban planning [17, p. 413:8].
5. DES is an appropriate paradigm for modeling the details of physical components of a complex systems, such as the technical components in socio-technical systems, whereas ABMS is able to capture social-level complexity and adaptiveness of actors [17, p. 413:10].

Based on this overview, DES would be an appropriate modeling and simulation strategy for centralized system, whereas ABMS would be an appropriate strategy for de-centralized, distributed systems.

3.2.1.3 The case for ABM

As discussed, both SD and DES have difficulty capturing aspects of emergence and self-organization in CAS. Furthermore, in contrast to ABM, both paradigms also have difficulty capturing the evolution of complex systems, that is their adaptive capacity, as the system or process structure is assumed to be fixed [17, p. 413:9]. Whereas two of the distinguishing features in ABM are its capacity to model heterogeneity of agents across a population, and its ability to capture self-organization and emergent phenomena. This leads to the conclusion that modeling and simulating MAS and CAS is outside the scope of DES and SD. In turn, this make ABMS a promising approach for capturing the complexity arising from the many interactions of goal-driven, decision-making entities, such as building elements in the case of architectural design.

Bonabeau [23, p. 7287] specifically recommends the use of ABM when one or more of the following conditions are true in the system under investigation:

1. when the interactions between the agents are complex, nonlinear, discontinuous, or discrete;
2. when space is crucial and the agents' positions are not fixed;
3. when the population is heterogeneous and each individual is (potentially) different;
4. when the topology of the interactions is heterogeneous and complex; and
5. when the agents exhibit complex behavior, including learning and adaptation.

Many authors make the case for using ABM and provide additional benefits over alternative approaches [17; 23; 122]. For example, according to Bonabeau [23, p. 7280] "ABM provides a natural description of a system" by modeling it from the perspectives of its constituent units that is also flexible and easily extensible. Furthermore, ABM is particularly useful when the appropriate level of description or complexity of a system is not known ahead of time and finding it requires some tinkering [23, p. 7281]. Finally, ABM provides the opportunity to develop models in an iterative way, in which "the modeler may start out with an idealized and general model and then make the underlying structure of model more [complex] by iteratively adding details" [17, p. 413:8] (see also Metz [132, p. 24]).

Particularly the last two aspects regarding the level of description and the iterative way of developing ABMs are relevant in the context of architectural design, where models are iteratively refined over the course of the design process and become more and more detailed.

3.2.2 Characteristics of Agent-based Modeling and Simulation

ABMS is a method to model and simulate the dynamics of complex adaptive systems that are composed of many autonomous decision-making entities called agents [124, p. 153]. ABMS can thus be seen as a variation of particle-based modeling approaches, which can be used to model and simulate complex physical systems, but where the constituent entities (particles) are endowed with decision-making capacity and the ability to adapt their behaviors.

In ABMS, the agents interact with each other and the environment according to rules that are defined at the micro-level, that is, at the level of each agent. This approach can apply to biological and social systems but also to engineered systems such as robot swarms and, by extension, to hypothetical systems. These systems typically have in common that the macro-level outcome of the interactions between agents can be counterintuitive and hard to predict using other classical simulation approaches [23, pp. 7281-84].

ABMS, which is also called individual-based modeling, is naturally suited to simulate such systems since the model is built from the perspective of the individual entities. In ABMs, the result of the interactions is computed iteratively, which relies on the power of computers to explore the dynamics of a system that are out of reach of pure mathematical methods [23, p. 7280]. The main difference of ABMS to, for example, differential Equation-based Modelling (EBM) employed in system dynamics with regards to complex systems is that in EBM, the models are based on system observables rather than on the inner workings of the system and thus do not capture the nonlinear interactions of the involved entities [133, p. 10] (see also Section 3.2.1).

The premise of ABMS thus is to aid science and engineering in gaining insight into the real world and how the real world behaves in order to develop models and theories of real systems for particular purposes [75, Sec. 2.11]. According to Macal [121, p. 145], this is possible, since "ABMS offers a method of implementing causal processes and mechanisms in a model not only to determine the implications of theory [...] but also to provide a basis for obtaining causal explanations of modeled phenomena."

ABM is a form of modeling and simulation that typically represents a real system over time. The simulation can simply be the "playing out" of the model dynamics or be considered a 'classical' experiment as in testing a hypothesis [132, p. 7]. Citing Clarke und Primo, Metz [132, p. 17] summarizes representation as the crucial aspect, as in any simulation: "We study one thing, the phenomenon, by studying another thing, the model."

In cases where an ABM is representing a real system, the model is then by necessity a simplification of reality. It does not need to correspond 100% to reality, but rather aims at representing the characteristic features of the section of reality under investigation in order to allow understanding of observed phenomena [132, p. 7], which Holland called 'homomorphisms' [57, p. 60]. According to Bonabeau [23, p. 7287], modeling the real system at the right level of abstraction "remains an art, more than a science" and hints at the challenges associated with ABM with respect to the validation of the model (see also Section 3.3.3).

Relevant, intrinsic features of ABM thus are:

1. an agent can represent and simulate any entity that is consistent with methodological individualism, including psychological, empirical or normative models that are situated in a corresponding environment [132, p. 20];
2. time plays a crucial role, since ABMs have to be "played out" in order to see the emergent system effects [88, p. 13] (see also [23, p. 7280]);
3. agents may be capable of adapting and evolving, allowing unanticipated behaviors to emerge; sophisticated ABM sometimes incorporates neural networks, evolutionary algorithms, or other learning techniques to allow realistic learning and adaptation [23, p. 7280]; and
4. behaviors of the agents might be influenced by the global behavior of the system in a process called 'downward causation' [47, p. 5].

Motivations for using ABMS as a modeling and simulation approach can be summarized as follows: as a way to model the world around that is more faithful to the real world [121, p. 145]; as a way to advance science directly and to either test or develop new theories (ibid.); as a way to explore emergent behaviors that might be exhibited by the simulated system [120, p. 119].

The ability to design experiments from the bottom-up using ABM in order to provide insight into the inner workings of complex (adaptive) systems has led some researchers, especially in fields that were previously lacking adequate simulation methods, to consider ABM a "scientific revolution." Axelrod even considered ABM a "third way of doing science" in addition to inductive and deductive reasoning [121, p. 145].

3.2.3 Classifications of ABMS

In order to distinguish ABMS from other modeling and analytical approaches, Macal [121, p. 149–50] provides four WIP definitions ranging in increasing order of complexity from individual ABMs, to autonomous, to interactive, and finally to adaptive ABMs. In an *interactive* ABM autonomous agents interact with other agents and with the environment. This admits the possibility of models that not based on processes or activities in the real world, as for example in applications of swarm intelligence (see Section 3.2.8), and are thus not simulations in the usual sense [121, p. 151]. An *adaptive* ABM is one in which the interacting, autonomous agents change their behaviors during the simulation incorporating adaptation at the level of the agents as the result of learning. This can go beyond simple rules that change rules by using other nature-inspired algorithms, such as machine learning and genetic programming [121, p. 150].

Reynolds [158, p. 1–2], with his boids model one of the protagonists of ABMS (see 3.1.4.1), proposes another classification distinguishing between isolated vs. situated (in a world shared by others), reactive vs. deliberative, abstract vs. embodied, and real (mechanical devices that exist in the real world) vs. virtual (real agents in a virtual world) agents. The autonomous characters of boids model are: "situated, embodied, reactive, virtual agents." This definition extends the one given by Macal above towards deliberation of cognitive agents that goes beyond simple learning mechanisms used in adaptive ABM. However, it is worth noting that deliberation, as implemented in cognitive agents, is not a necessary requirement for adaptation since adaptation can also occur in reactive agent models, both on the level of the individual agents, as well as on the level of the population (see Section 3.2.7).

Based on this taxonomy, the agent systems that are the focus of this research into the design of segmented shells are those where agents are situated, embodied, virtual and reactive. These reactive agent models result in interactive agent systems and, through adaptation on the level of the population or by implementing learning mechanisms on the level of the agents, can become adaptive agent systems. By extension, an application of ABM in the context of fabrication and construction will thus require situated, embodied, real and possibly deliberative agents resulting in cyber-physical agent systems where computational simulation and real system are mirrors or 'twins' of each other.

Within the context of an application in architectural design and engineering, agent-based modeling and simulation thus is applicable as (1) an approach for design exploration, in which a dynamic and time-dependent process is modeled involving many interacting entities capable of adaptation; as well as (2) an approach towards optimization or search. For example, 'Swarm intelligence' algorithms, which are inspired by multi-agent systems and their corresponding agent-based modeling approaches, can be used to achieve an (optimal) end state rather than to investigate a dynamic process [124, p. 152] (see Section 3.2.8).

3.2.4 Constituent elements of an ABM

The following section introduces the basic elements and constructs that are used in ABMS. Agent-based models consist of agents that exhibit behaviors, of an interaction topology and methods of interaction between agents, of the environment in which agents interact, and time simulated through different update-mechanisms. The choice of specific constructs is important as they amount to design decisions, which are based on "underlying assumptions" with respect to the intended use of the model. These assumptions themselves are "based on a model of how modeling should be done" [17, p. 413:1].

3.2.4.1 Agents

According to Metz [132, p. 18], there is no universally accepted or applicable definition of what constitutes an agent that goes beyond entities embedded in an environment interacting over time. He subsequently defines agents as "carriers of those features of the entities in the system under investigation that are considered relevant with respect to the phenomenon that is being modeled" [132, p. 20]. Jennings [82, p. 280] defines an agent as "an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives."

Following in the footsteps of Wooldridge and Jennnings [213, p. 116], Macal and North [124, p. 153] distinguish between essential and optional characteristics of agents in an attempt to cover a broad spectrum of agent systems in different fields. Agents thus have to be

1. *self-contained* and *identifiable*;
2. *autonomous* and *self-directed* having behaviors that relate sensed information to decisions and actions.
3. Agents have a *state*, that is, a set or subset of the agent's attributes that varies over time; and
4. agents are *social*, that is, they have dynamic interactions with other agents that influence their individual behavior.

Optional properties are adaptation, goal-orientation and heterogeneity within the population. It is worth noting that some authors, such as Epstein, would use the term heterogeneous to describe agent systems where agents have a range of different attribute values, e.g., different plate outlines, as opposed to having different attributes, while other authors might call these agents homogeneous due to their similarity [38, p. 8]. Within the context of this work, heterogeneity therefore refers to a "differentiation in terms of the agent specifications themselves; that is, a difference in their rules and/or their set of attributes" [37, p. 22]. Furthermore, the characteristics mentioned above define the agents used in this research.

3.2.4.2 Behaviors

In order to model agent behavior, a definition of what constitutes behavior is necessary. Not surprisingly, similar to the problem of defining the term 'agent', there is not universally accepted definition of ‘behavior’, not even in ethology, i.e. the field in biology that studies animal behavior. Facing this lack of consensus on a term that is so widely used in behavioral sciences and especially in ethology, Levitis et al. [113] surveyed the use of the term across the discipline and proposed a working definition: This definition only needs to be slightly adapted in order to encompass digital entities [113, p. 1]:

> “Behavior is the internally coordinated responses (actions or inactions) of whole [entities] (individuals or groups) to internal and/or external stimuli”.

Given that research in multi-agent systems has historically looked for inspiration in biology, and specifically ethology, [137, p. 69], it seems appropriate to adopt this definition. In the context of this research, the internally coordinated relationship between sensed information and agent actions is thus called behavior (Fig. 3.1).

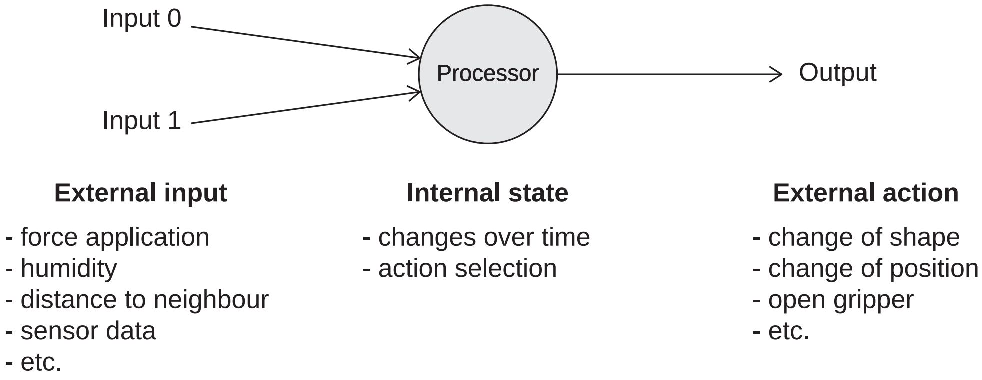


Figure 3.1: A diagram defining behavior.

An agent gathers information through interactions with other agents and with the environment. Anything from simple, ‘reactive’ rules to abstract, ‘cognitive’ models can be specified in order to specify agent behavior [124, p. 153], that is, to define how the state of an agent should change based on a given input over time (see also Section 3.2.6). More advanced algorithms, such as genetic programs or artificial neural networks, allow agents to adapt their behaviors (reactive rules or internal models) effectively learning from past experience [121, p. 150].

The term behavior is widely used also in the contexts of architecture and engineering for example to describe ‘material behavior’ or the ‘behavior of a structural system’. It is worth noting that while behavior in those contexts also denotes a form of adaptation to external stimuli based on inherent rules of the system, these systems are not considered agent systems. The missing criterion is autonomy and social ability of the constituent entities (see agent definition above). These systems can however be considered complex physical systems whose behavior can be modeled, for example, using particle-based approaches.

3.2.4.3 Relationships and interaction

Two of the basic assumptions in ABMS regarding relationships between entities stem from CAS research: (1) only local information is available to each agent; and (2) there is no central authority that either pushes out globally available information to all agents or controls their behavior in an effort to optimize system performance [124, p. 154]. This is due to the main uses of ABMs to represent real systems that are decentralized, where no central control mechanism is available, as in colonies of ants, termites, bees etc. but also in the stock market, crowd stampede or customer behavior. In technical, goal-oriented applications, however, the 'decentrality' constraint does not necessarily need to hold as exemplified by Fully-Informed Particle Swarm Optimisation (PSO), a variant of the standard PSO algorithm [128, p. 204].

With interaction between the constituent elements of the system being the distinguishing feature of complex systems, one of the primary concerns in ABM is to decide who interacts with whom and how the interaction takes place [124, p. 154]. Interaction between agents is determined by an interaction topology, which is chosen by the modeler depending on the system being modeled. Following are predominant topologies:

1. *rectangular grids and lattices* in one, two and three dimensions as in Cellular Automata, where agents are either cells, situated within cells, or on grid points [120, p. 125];
2. *networks*, in which agents represent vertices and relationships are indicated by edges (ibid.); this also includes *global interaction*, in which every agent is connected to and interacts with every other agent;
3. *continuous Euclidean space* in one, two, three, and higher dimensions [124, p. 155];
4. *non-spatial* interactions, where agents have no locational attribute and pairs of agents are randomly selected—also called 'soup' (ibid.);
5. Geographical Information Systems (GIS), in which agents move over geographical patches, relaxing the one-agent per cell restriction [120, p. 125];

Many ABMs include agents interacting in multiple topologies [124, p. 155]. Finally, in order to determine who interacts with whom, the concept of 'neighborhood' is introduced. Typical neighborhoods in cellular and grid-based topologies are von Neumann and Moore's neighborhood [120, p. 116] (parameterized by 'Manhattan distance'). In continuous space topologies, the neighborhood might be defined by a sphere of influence or field of view (such as in Reynolds' boids model [156, p. 30]); in network topologies, the neighborhood is defined by the degree of separation.

3.2.4.4 Environment

In ABMS, the environment typically fulfills two purposes: first, it is the substrate, which enables actions of the agents, for example by transmitting information; second, it represents the context in which agents 'live' and interact [132, p. 20] using any of the interaction topologies outlined above. From the agent's point of view, the environment can have different characteristics, for example it can be fully or only partially visible, static or dynamic, deterministic or stochastic, agent actions and the current state can influence the future of the environment or not, it can be discrete or continuous and it can be populated by other agents or by just one agent (ibid.). Very often, the environment is also active and dynamic, affecting the other agents contained within it, in which case the environment is defined as a special type of agent [134, p. 7].

3.2.4.5 Time and update mechanisms

Similar to the previous design decisions regarding the agent model, the way time is represented is dependent on the intended use of the model and the system being modeled. According to Michel et al. [133, p. 22], there are three main approaches to model time: (1) continuous time (e.g. using differential equations, where the system state can be computed for every time stamp); (2) discrete time (time unfolds discretely with constant time intervals, where it is assumed that no activity occurs in the system in-between increments); (3) discrete-event based (where time evolves discretely from one event to the next, as for example in simulations of customer queuing or of assembly lines). The ability to model systems that change their states based on events, makes ABMS applicable for DES. Conversely, only a subset of problems for which ABMS is applicable, can be implemented in DES software packages [179, p. 206] (see also Section 3.2.1.2).

At each iteration in the simulation, be it based on continuous or discrete time, the state of the agents needs to be updated in order to calculate the current state of the overall system. There are three main updating mechanisms, which are (1) synchronous updating; (2) asynchronous updating with random or fixed order and (3) asynchronous updating based on incentive, that is, only those agents update that have the largest difference between a current and a target state [132, p. 22]. According to Dorigo [51, Sec. 2.2], asynchronous updates lead to a faster propagation of the best solutions through the swarm in PSO.

3.2.5 Emergence vs. Self-Organization

One of the characteristic features of complex systems that can be modeled using ABMS is that they exhibit what is called *emergence*. Emergence is commonly defined as "the whole [being] greater than the sum of its parts" or, from a reductionist perspective, "the macro-level emergents not [being] reducible to the micro-level parts of the system" [47, p. 4]. These systems often also exhibit *self-organization*, which describes a spontaneous increase in order or structure of a system without external control (ibid.). According to De Wolf and Holvoet [47, p. 1], these two phenomena are often confused in literature. The authors thus highlight similarities and differences between the two concepts and subsequently propose working definitions:

> "A system exhibits emergence when there are coherent emergents at the macro-level that dynamically arise from the interactions between the parts at the micro-level. Such emergents are novel w.r.t. the individual parts of the system" and
>
> "self-organization is a dynamical and adaptive process where systems acquire and maintain structure themselves, without external control."

Both definitions are adopted in the context of this research, as both phenomena are relevant in an application of ABM in architectural design and specifically in the context of design of segmented shells. The emphasis on novelty of emergent systems is an important aspect in design exploration, which aims at uncovering novelty with respect to the basic assumptions inherent in the design decisions. Similarly, the emphasis on increasing order and structure of the system is highly relevant in agent systems where agents represent building elements. As De Wolf and Holvoet [47, p. 1] state, both phenomena can exist in isolation, but they can also coexist in one dynamical system (ibid.). The authors thus propose a combination of emergence and self-organization as a promising approach to engineer large-scale multi-agent systems [47, p. 11].

For Bonabeau [23, p. 7280], ABMS is, by its very nature, the "canonical approach to modeling emergent phenomena": in ABMs, one models and simulates the behavior of the system's constituent units (the agents) and their interactions, capturing emergence from the bottom up when the simulation is run.

3.2.6 Knowledge-based vs. Behavior-based AI

Another important distinction regarding ABMS approaches has to be made between *cognitive* and *reactive* agent architectures—a distinction that corresponds to the differences between classical and 'anti-classical' AI or knowledge-based and behavior-based AI (see also Section 3.1.4.2). Krause [96], in the context of the first experiments with ABM in the field of architecture, provides a comparison.

On the one hand, Knowledge Based Artificial Intelligence (KBAI) is the classical, 'high-level' approach towards modeling intelligence, where "models are top-down in structure and usually model single competencies [...]. A KBAI system tends to be closed and non-autonomous; information in the system is largely interpreted though reasoning and planning roles. Empirical knowledge is not acquired within the system because this knowledge is built in from the start. [...] The internal models used within these methods are complete and need to be correct relative to the problems under investigation" [96, p. 64].

Behavior-based Artificial Intelligence (BBAI), on the other hand, is an alternative, 'anti-classical' approach for "evolving multiple competencies at a very low level," where models are bottom-up in structure consisting of "rudimentary agents which either grow in capability through collaboration, union or disjunction, or learn through countless tests' driven by fitness. BBAI systems are autonomous and open. Less emphasis is placed on the system and more on the behavior of the agents. The agent's focus tends to be on development, adaptation, and interaction with the environment" [96, pp. 64–65].

Michel et al. [133] provide additional definitions and examples for the two principal architectures. In *reactive architectures*, an "agent does not have an explicit representation of its environment nor of other agents. Its behavior is entirely described in terms of stimuli-response loops which represent simple connections between what they perceive and the set of available operations that may be performed" [133, pp. 14–15]. Examples for this kind of architecture are the (1) subsumption architecture, (2) competitive task architecture, and (3) connectionist architecture, where (2) and (3) implement learning mechanisms. Additionally, there are (4) combinations of prioritization and neural learning in a hierarchical structure, as well as (5) architectures that represent behaviors as combinations of vectors (ibid.). In reactive architectures, behaviors thus are specified at the most basic level using if-else rules of varying complexity [132, p. 19] and at a higher level through "rules to change the rules" [124, p. 153].

Cognitive architectures, on the other hand, are "founded on the computational metaphor, which considers that agents reason from knowledge described with a symbolic formalism. This knowledge explicitly represents their environment (states, properties, dynamics of objects in the environment) and the other agents. The most well-known architecture of this type is the BDI (Belief-Desire-Intention) which postulates that an agent is characterized by its beliefs, its goals (desires) and intentions. [...] The main quality of BDI architectures is to create a behavior which mimics that of a rational human being" [133, pp. 15–16]. For Metz [132, p. 19], the main characteristic of cognitive agents is that they do have internal models and representations of mental concepts, for example mechanisms that can emulate human decision-making, without explicit programming of if-else rules.

Hybrid architectures constitute a third approach towards modeling agents, which combine reactive and cognitive approaches in order to “build [the] best suited architecture to solve a problem” [133, p. 16]. Within the context of this research, however, the main focus will be on reactive architectures that implement learning mechanisms in order to enable adaptation.

3.2.7 Adaptation and Learning in Agent-Based Models

One of the characterizing features of CAS and the corresponding modeling and simulation approach using adaptive ABMs (see Section 3.2.3) is the ability of agents to adapt to changing conditions in the environment and to learn from past experience as the simulation proceeds through time. Furthermore, the ability of agents to adjust their behaviors is the cause of a well-known effect that can be observed in CAS, which is termed ‘downward causation’: the process by which agents change their behavior at the micro-level in response to the emergent system behavior at the macro-level [120, p. 113] (see also [47, p. 5]).

Adaptation in agent-based models can thus occur on two levels: on the level of the individual through learning where agents adapt their individual behaviors based on interaction with each other and the environment; and on the level of the population through evolutionary mechanisms of adaptation by changing the proportional representation of agents in the population according to their fitness and by generating new types of agents and behaviors [120, p. 123]. Correspondingly, other nature-inspired algorithms from the field of ALife are often used with ABM in order to implement adaptation on the individual and on the system level, which include GAs, Evolutionary programming, Artificial Neural Networks (ANN) and methods of Swarm Intelligence [120, pp. 121-23].

On the individual level, Evolutionary Programming can be used, where programs represent agent behaviors that can be evolved directly [120, p. 123]. On the same level, artificial neural networks (ANN) have been applied to modeling adaptive agent behaviors, in which an agent derives a statistical relationship between the environmental conditions it faces, its history, and its actions (ibid).

On the system level, GAs can be used, where “a chromosome effectively represents a single agent action (output) given a specific condition or environmental stimulus (input). Behaviors that [...] enable the agent to respond better to environmental challenges are reinforced and acquire a greater share of the chromosome pool. Behaviors that fail to improve the organism’s fitness diminish in their representation in the population” [120, p. 123]. On the same level, Swarm intelligence approaches can be used for optimizing parameter selection for agent behaviors [120, p. 124] enabling adaptation on the system level.

3.2.8 Swarm Intelligence

The ability of CAS in nature (such as flocks of birds, schools of fish, colonies of ants, termites, bees or wasps) to effectively forage for food, evade predators, or re-locate colonies through cooperation of multiple individual agents without central control has inspired a number of computational approaches that are now widely used in engineering applications for optimization.

These approaches, which are commonly grouped under the term ‘Swarm intelligence’, thus aim at harnessing the ability of CAS to solve problems that are difficult to solve using conventional approaches, especially finding global optima in ‘non-convex’ optimization problems. Such approaches, which “orchestrate an interaction between local improvement procedures and higher level strategies to create a process capable of escaping from local optima and performing a robust search of a solution space” are called meta-heuristic [63, p. vii]. They include the swarm-inspired stochastic PSO method, Ant Colony Optimisation (ACO), and other population-based search algorithms, such as GAs and Genetic Programming, all of which utilize various metaphors from natural systems.

Within the field of meta-heuristic approaches, the swarm intelligence paradigm consists of two dominant sub-fields: ACO and PSO [32, p. 229]. In PSO, each particle represents a candidate solution to the optimization problem and, using the flocking metaphor, its speed and direction of travel across the continuous search space are influenced by the best solution of its immediate neighbors as well as its own personal best solution that was previously achieved (ibid.).

ACO builds on the metaphor of stigmergy, that is ants using pheromone trails in order to find the shortest or ‘minimum cost’ path between a food source and the nest, and is used when the optimization problem can be transformed into the problem of finding the best path on a weighted graph as in the traveling salesman problem [50, Sec. 1]. Similar to their biological role models, these optimization approaches are inherently agent-based involving multiple interacting agents. However, there are important differences in the underlying assumptions and consequently in the use cases when compared to ABMS. First of all, one can argue that, while swarm algorithms are fully agent-based models, they are not technically simulations in the usual sense of the term, as no processes of activities of the real world are simulated [121, p. 151]. Furthermore, in swarm-based optimization, the practical focus is on harnessing the ‘power of the many’ in order to converge on a solution, an (optimal) end state [124, p. 152], whereas ABMS also allows for the investigation of the dynamics of a complex system [124, p. 151]. In applications of swarm optimization, systems are composed of homogeneous particles with static rules (see Brownlee [32]), such as idealized gas particles, whereas in ABMS, the population can be composed of heterogeneous agents with different and evolving behaviors at different times and in different contexts [23, p. 7287]. As mentioned above, the difference between Swarm Intelligence and ABMS does not preclude applications where both approaches are used together, as for example in the case of optimizing behavior parameters of agents in agent-based models as a form of adaptation (in which case an agent system constitutes an individual particle in PSO).

3.2.9 The role of simulation in ABMS

With ABMs describing systems that are out-of-reach of alternative approaches, critical aspects of modeling with agents are (1) the validation of the model and of the outcome of the simulation, (2) the role that the model can play in the scientific investigation, and (3) the question of what constitutes an agent-based explanation of an emergent phenomenon.

3.2.9.1 Validation

According to Heath, Hill and Ciarallo [75, Sec. 2.13], the purpose of simulation validation is “ensuring the model is an appropriate representation of the system of interest for a given set of objectives.” The authors subsequently propose a two-stage validation process: “The first round validates the conceptual model. The conceptual model is the abstracted model of the real system [...] and forms the basis for an ABM model” (ibid.). “The second round validates results of the simulation against results from the real system. For a model to be completely valid, it must be validated both conceptually and operationally” (ibid.). One significant and “alarming” finding of the survey by Heath, Hill and Ciarallo [75, Sec. 3.9] was that 65% of the models in the survey were incompletely validated; but it also showed that the ratio of validated models significantly increased over the survey period.

Metz [132, pp. 39–40] reframes the challenge of validation as a trade-off: on the one hand, a user of an agent-based model might not be able to refer to tests of model validity or significance; on the other hand, a well-documented and well-structured model, “permits a view on those elements of the conceptual model that would justify trust in the simulation results.” Again, the potential openness and the individual-based modeling perspective make ABMS easier to communicate to stakeholders than, for example, mathematical models of System Dynamics, which “suffer from the same validation questions but this has not proved a substantial barrier” [179, p. 210].

3.2.9.2 Different roles of simulation

With respect to the role that simulation and modeling play in science today, Heath, Hill and Ciarallo [75, Sec. 2.16] distinguish purposes of ABMs based upon the "level of understanding associated with the system of interest." Purposes range across a spectrum from models that are used as 'Generators' and 'Mediators' to models that can be used as 'Predictors' with corresponding expectations towards the simulation (Fig. 3.2):

- *Generator* models can be used to test a hypothesis where little is known about the functioning of the system. The simulation acts as a generator of hypotheses and theories about how the real system behaves [75, Sec. 2.17].
- In *Mediator* models, the system is moderately understood and a microscopic perspective is adopted. The simulation is used to gain insight into an aspect of the system behavior, but is not a complete representation of how the real system behaves. These models can be used to improve an existing conceptual model (ibid.).
- In *Predictor* models, the system is well understood and the model can be "used like a calculator". The simulation provides predictions about the system's behavior using an accepted conceptual model, for example a queuing system or an assembly line (ibid.).

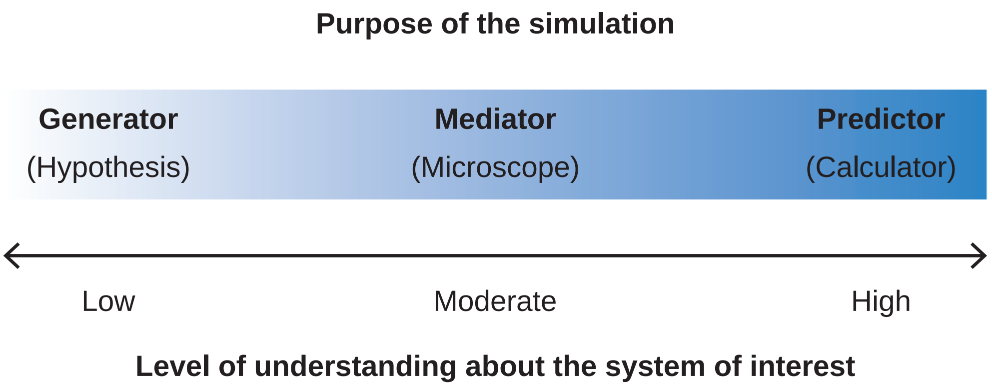


Figure 3.2: The role of simulation. Redrawn from [75, Fig. 4].

Particularly the role of models as Generators aligns well with the concept of 'exploratory modeling' championed by John Holland: probing the effects of hypothesized underlying mechanisms can generate an understanding of how systems work. According to Holland, "the key to science is to understand the mechanisms that cause a system to behave in a certain way" [57, p. 61].

In architectural design and engineering, a familiar example of the use of a simulation as predictor is Finite-Element Analysis (FEA), where once the conceptual model is validated by the structural engineer and correctly implemented in Structural Analysis software, the result of FEA constitutes a prediction of how the structural system under investigation will perform. Correspondingly, a scenario for the use of agent-based models as a Generator would be design exploration, as the divergent search process focussing on novelty and variation, where the designs can be developed by testing implications of hypothetical design decisions. This aspect of exploratory modeling is particularly relevant for early architectural design stages.

3.2.9.3 Agent-based explanation

Assuming ABMS in a typical application scenario, where the model represents a real system, with a validated conceptual model, correct implementation, and validated outcome, it still raises the question to what extent the model and the simulation constitute an explanation of the real system and what, in fact, constitutes an explanation of an emergent phenomenon [23, p. 7281]. In literature, these considerations are summarized under the term 'generative explanation'. Citing Epstein, Metz [132, p. 18] notes that those models, which are able to produce a correct macro-outcome, can be considered sufficient as an explanation insofar that they not only correctly predict the outcome, but also demonstrate in the constructive sense how the outcome was produced. Further, a sufficient condition does not exclude other conditions, such that a generative explanation can only be one among potentially many possible unless its validity has been decided on the micro-level.

3.3 State-of-the-art

The microscopic, individual-oriented perspective of ABMS lends itself to many different areas of application, as suggested by Bonabeau [23, p. 7280], which has led to a rising popularity of ABMS in the past 20 years. Heath, Hill and Ciarallo [75, Sec. 2.6] in their survey of 279 articles on agent-based modeling showed that in the time period from 1998-2008, the distribution of the number of articles per year has continuously increased and that ABMS has become a popular approach in the modeling and simulation field.

Using a similar approach, the number of articles per year in the following 10 years (2009 - 2018) was determined by searching for the term "agent-based" in publication titles stored in Clarivate's Web of Science database (formerly Thomson Reuter). The analysis focused on the 25 most relevant research areas according to number of publications totaling 6410 publications (covering 98.6% of all publications in this period logged in the database). This analysis excluded the research areas of "energy fuels”, “materials science”, “chemistry”, “biochemistry”, “molecular biology”, and “pharmacology pharmacy” due to their alternate use of the term ‘agent’ (Fig. 3.3).

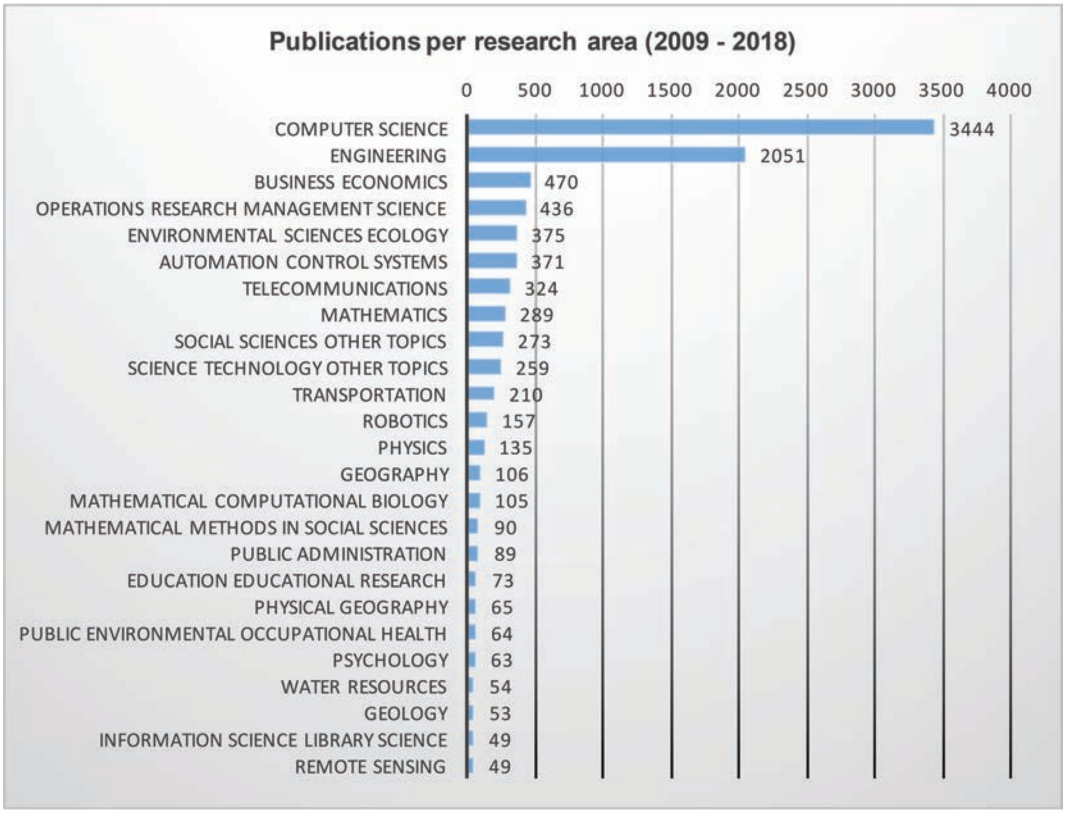


Figure 3.3: 25 most prominent research areas for “agent-based” research.

This first overview on the research areas highlights that ABM is actively used in a variety of scientific domains with the main disciplines being Computer Science, Engineering, Business Economics, Operations Research Management Science, and Environmental Sciences and Ecology, which still shows a general overlap with the disciplines identified by Heath, Hill and Ciarallo [75, Sec. 3.3]. The major difference is Computer Science (including Artificial Intelligence) now being the dominant discipline. Furthermore, 32 articles have been logged from the field of construction and building technology in the past ten years.

The yearly distribution of publications over the analysis period shows a general trend where the number of articles has continued to rise following Heath, Hill and Ciarallo’s survey from 2009 [75](Fig. 3.4).

The observation that Computer Science is the major discipline using ABM approaches coincides with Jennings [82, p. 293] who postulated that ABM “should be seen in its broader context as a general-purpose model of computation that naturally encompasses the major trends in software”: (1) distributed and concurrent systems as the norm rather than the exception; (2) the focus on flexible interactions between (independently developed) software systems; and (3) reflecting real-world relationships in computer systems.

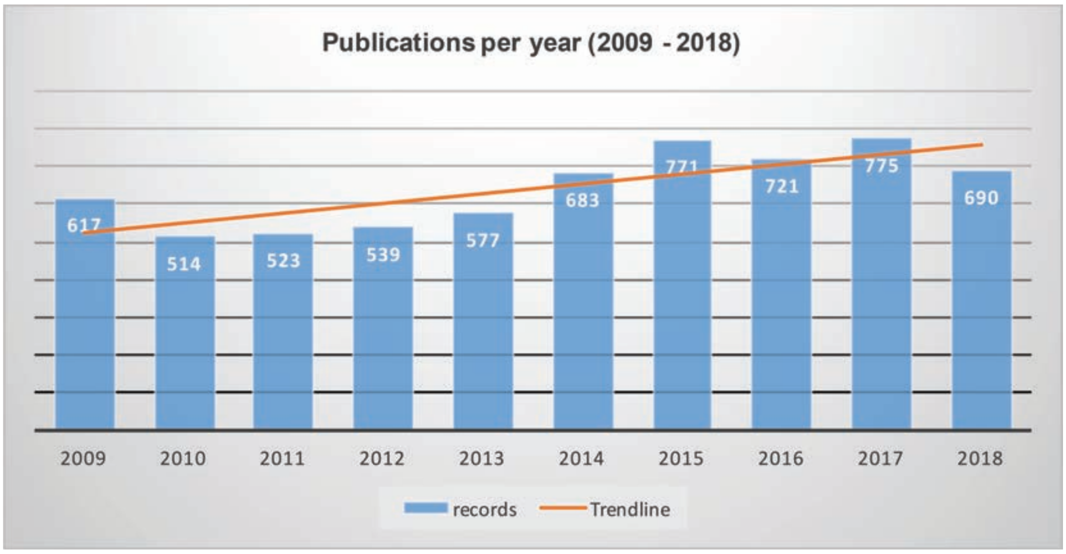


Figure 3.4: Histogram of number of agent-related articles from 2009-2018.

3.3.1 Applications of ABMS

The number of application scenarios for ABM is too large to enumerate, but a few authors have tried to summarize the main current applications in the areas of biology, ecology, social sciences, archaeology, engineering as well as in 'socio-technical' systems [23; 120; 121; 123]. This section will thus give a broad overview on example applications of ABMS in different fields. The common trait of these applications being that they investigate phenomena and hypothetical scenarios that would be difficult to model and analyze using alternative methods due to their complex nature. This section is then followed by the state-of-the-art in the field of architecture and construction (Section 3.4).

3.3.1.1 Archaeology and Anthropology

Archaeology and Anthropology are fields where ABM is actively used, as reported by Macal [121, p. 147] citing recent publications in the fields. One application scenario is the use of large-scale agent-based modeling in order to provide an experimental virtual laboratory for long-vanished civilizations [124, p. 157].

An example is the agent-based model by Griffin and Stanish, reported by Macal and North [122, p. 91], for the Lake Titicaca basin of Peru and Bolivia covering the late prehistoric period, 2500 BCE to 1000 CE. Agent behavior is modeled as a set of condition-action rules that are based on hypothesized causal factors affecting agricultural production, migration, competition, and trade. Through a series of simulation runs, "the model produced a range of alternative political pre-histories and the emergence of macro-level patterns that corresponded to observed patterns in the archaeological record" (ibid.).

3.3.1.2 Biology and Ecology

Applications in biology and ecology aim at modeling diversity and dynamics of populations of organisms and species [121, p. 147]. Examples include an ABM by Bryson et al. [34] to investigate dominance behavior between individuals in primate populations; or agent-based models of predator-prey relationships, such as the one by Mock and Testa, between transient killer whales and threatened marine mammal species (sea lions and sea otters) in Alaska, as reported by Macal and North [124, p. 156]. A further example is the use of ABMs to model food webs as complex adaptive systems, such as the model by Peacor, Riolo and Pascual, where agents are individuals or species representatives, as reported by Macal [120, p. 128]: "Adaptation and learning for agents in such food webs can be modeled to explore diversity, relative population sizes, and resiliency to environmental insult."

3.3.1.3 Epidemiology, Immunology and Computer network security

Another active area of research is the spread of diseases in populations as well as in the human body. For example, agent-based modeling and simulation of the former is, together with SD, one of the established approaches to modeling and simulating the spread of COVID-19 [197]. Agent-based models of the later have been developed to model the interactions between the cells of immune system. One example is the 'Basic Immune Simulator', which is based on a general agent-based framework (the *Repast* agent-based modeling toolkit) [124, p. 156]. Macal [120, p. 127] reports that approaches for modeling the immune system have also inspired agent-based models of intrusion detection for computer networks, and have also found use in modeling the development and spread of cancer.

3.3.1.4 Social Sciences and Economics

As outlined in section 3.1.3, modeling urban segregation using cellular automata was one of the first uses of an ABM. The quick adoption of ABMS in the Social Sciences and its use to model social processes as emergent processes and their emergence as the result of social interaction has led to the term 'Generative Social Science' [120, p. 126]. Epstein has argued that social processes are not fully understood unless one is able to theorize how they work at a deep level and have social processes emerge as part of a computational model (ibid.). Macal (ibid.) reports the use of ABMS in social network analysis to simulate dissipation as, for example, in the spread of political opinions, product adoption, or of 'memes' (as the smallest cultural unit).

Another dominant field of application is economics as illustrated by the emerging field of Agent-based Computational Economics. The failure of the established economic models to predict, or even suggest, the possibility of an impending economic crisis in 2008 was a crisis in itself for the field of economic forecasting [121, p. 147].

This caused a re-evaluation of the state of economic modeling, and lead to questions if ABMS could do better than conventional models [189].

Here, the particular potential of ABM is to be able to relax some of the standard assumptions of classical economics, for example with respect to the rationality of economic actors, their homogeneity, and the long-run equilibrium of the system [124, p. 157]. ABM can be used to understand the importance of each simplifying assumption, and the effect of dispensing with the assumption in producing more accurate economic forecasts can be tested [121, p. 147].

3.3.1.5 Politics and Public Policy

According to Metz [132, p. 15], the last 10-15 years have also shown innovative research in the political sciences that illustrate the potential of the approach particularly in the domains of election polling (psephology), party system research, conflict research and political economy.

While there are few widely reported documented cases of ABMS applications that have had direct policy impacts, or that are used on a regular basis as part of decision-making, better ways for ABMS to inform policymakers has become an area of research itself [121, p. 152].

Still, seminal studies using ABM in the realm of public policy include modeling and simulation of public transportation and traffic in the Denver/Fort Worth region (1996-1998) and Portland metropolitan area (2001) using the TRANSIMS software framework: developed by the Los Alamos National Laboratory (LANL) starting in the early 1990s, the ambitious TRANSIMS project was an “integrated traffic simulation approach aiming to provide transportation planners with complete information on traffic impacts, congestion and pollution” [39, p. 170] (see also [23, p. 7282]).

3.3.1.6 Business and Management

Bonabeau [23] gives an overview of applications of ABMS in business contexts such as modeling the flow of people through amusement parks, supermarkets and stores providing valuable information about customer behavior; agent-based modeling of stock markets and auctions; modeling operational risk in banking; and modeling product adoption. One conclusion of a study in the movie industry was that while “predicting hits might be the single most difficult thing to do; understanding how hits happen is a better use of the model” [23, p. 7286]. An example for an application of modeling and managing the flow of people is modeling pedestrian movement in urban spaces and inside buildings for example for modeling building egress using the reactive “social force model” [77].

3.3.1.7 Environmental and Urban Planning

As indicated by the TRANSIMS project, ABM has been used for transportation planning in regional and urban contexts including assessment of environmental impact. Chen [39, p. 169–70] lists more applications in the field of environmental planning, including water resource modeling, agriculture, forestry and the management of generic resource sharing. In this context, ABM is also actively used in conjunction with Geographic Information Systems (GIS) for modeling land use where agents move over a realistic geo-spatial landscape [124, p. 155] and urban planning including zoning and transportation [68].

3.3.1.8 Socio-technical systems

According to Behdani [17, p. 413:2] socio-technical systems are “systems that involve both complex physical-technical systems and networks of interdependent actors.” Typical examples are the electrical power grid, supply chains, assembly lines and flexible manufacturing, topics that fall into the domain of operational research (OR). In this context, Behdani [17, p. 413:3] discusses supply chains from the point of view of complex adaptive systems and compares ABM to DES and SD approaches (see Section 3.2.1). In a similar vein, Jennings [82, p. 284] illustrates his case for agent-based software development with an example from flexible manufacturing.

3.3.1.9 Engineering and manufacturing design

Agent-based approaches have been used in design applications in engineering and manufacturing either as assistance systems in the field of industrial and machine design in processes of high complexity as described by Kratzer, Binz and Watty [95] or in optimization as described in Section 3.2.8. An example of the Swarm Intelligence paradigm in engineering is the use of ACO for the design of space trusses [87].

3.3.1.10 Computer Science and AI

Finally, to conclude this overview, Computer Science is the discipline that not only provides the methods of implementation for ABM, most notably object-oriented programming, but it is itself applying agent-oriented programming. While agent-oriented programming is not the same as object-oriented programming, the object-oriented paradigm is a useful basis for ABMS, as an agent can be considered a autonomous, self-directed object with the capability of action choice [82, p. 280]. Jennings [82, p. 277] thus postulates that “agent-based computing represents an exciting new synthesis both for Artificial Intelligence (AI) and, more generally, Computer Science. It has the potential to significantly improve the theory and the practice of modeling, designing, and implementing computer systems.”

3.3.2 Spectrum of agent models and applications

Analogous to the wide range of applications, agent-based models range across a spectrum from small, elegant, minimalist models to large-scale decision support systems [124, p. 156]. Minimalist models are based on a set of idealized assumptions, and are designed to capture only the most salient features of a system. These models are exploratory 'digital laboratories' in which alternative assumptions can be varied over many simulations. Decision support models tend to be large-scale applications, designed to answer a broad range of real-world policy questions. These models are distinguished by the inclusion of real data and by having passed some level of validation to establish credibility (ibid.).

The range of model types aligns with the range of roles that models play in simulation ranging from Generator to Predictor models (see Section 3.2.9). Of the 279 publications on ABM that were surveyed by Heath, Hill and Ciarallo [75], 111 of the models were Generators, that is, the simulation acts as a generator of hypotheses and theories about how the real system behaves based on idealized assumptions; 168 were Mediators, that is, the simulation provides insight into the system, but is not a complete representation of how that system actually behaves; and 0 were Predictors, that is, the system is well understood with validated conceptual models and implementations. This finding confirmed the belief that agent-based models are used primarily to gain insight into the system of interest [75, Sec. 3.4]. Furthermore, this general characteristic of how agent-based models are used remained relatively constant over the 10 years of investigation (ibid.).

A further relevant finding of this study was that in some domains the majority of the models were generators: roughly 2/3 of the models in social science and in economics, whereas in business, public policy, and the military generator models were relatively uncommon. The authors thus speculate that the reason for the different role of models in those different domains is that "social science and economics are still relatively new and in the process of developing theories about how their systems of interest operate. Thus, using agent-based models as generators allows them to explore hypotheses and ideas that are not easily manipulated using other theory generating techniques" [75, Sec. 3.5].

3.3.3 Issues with ABM

Despite the benefits of ABM that have led to its wide acceptance across multiple fields, there are certain challenges that have to be addressed when using the agent-based approach towards modeling and simulation. These include determining the right level of abstraction in the description of the agents to serve the purpose of the simulation. According to Bonabeau [23, p. 7287], "this remains an art more than a science." A second issue also addressed by Bonabeau (ibid.) is determining the plausibility of the results, as part of the 'validation' of the model, which also relates to the role of the simulation and the associated expectations (see also [75] and Section 3.2.9.1).

A third issue, which, paradoxically is also one of strengths of the ABM approach, is emergence, which makes predicting the behavior of the overall system based on its constituent components extremely difficult (sometimes impossible) [82, p. 289]. Again, depending on the application of the model as generator or predictor of some other system or of the system itself, for example as a software-based application, this can also be a strength. It is also precisely the emergent property of the outcome, which makes ABMS valuable in terms of design exploration: the ability to test different outcomes by modifying rules, parameters and agent attributes and to generate 'novelty' with respect to the design decisions.

So, while there are techniques to reduce the system's unpredictability, such as "mechanism design from game theory," restrictions also tend to limit the power of the agent-based approach [82, p. 290]. Thus, a better understanding is needed of the link between behavior of the individual agents and that of the overall system (ibid.).

Finally, a forth challenge ABMS is facing is the computational cost of simulating the behavior of large systems, which can be time consuming [23, p. 7287]. However, following Moore's law of increasing computation power, this should become less of a concern. At the same time, as computation power increases, so does the scope of the systems that are being simulated. For example, the *Economist* [189] reports that in 2010 Doyne Farmer of the Santa Fe Institute and Robert Axtell of George Mason University explored the feasibility of constructing an immense agent-based model of the entire global economy.

3.4 ABMS in Architecture

3.4.1 Adoption in architecture

As described in Section 3.1, early experiments with ABM in the social sciences date back to the 1970s and 80s and the approach consolidated in various fields in the 1990s with the first general purpose toolkits (e.g., Swarm and Star Logo) emerging in the mid-1990s. First agent-based architectural design experiments using cellular automata were conducted by Frazer in the early 1990s [59]. First experiments using vector-based agents in continuous space weren't conducted until the second half of the 1990s [96]. These experiments were either conducted in those domain-independent toolkits (see [40]) or using general purpose languages such as C++ [96, p. 97]. It is worth noting that as of today a domain-specific software framework for the design and simulation of ABMs in architecture does not exist and ABMS is still far from being an established method in the architecture field.

One reason for the delay may have been the need for computation power to be able to iteratively manipulate geometric objects in Computer Aided Geometric Design (CAGD) as a precondition to make ABM a fruitful endeavor in architecture. In contrast, in other fields significant results could be achieved using 2-dimensional, discrete-space models of systems that allow for high-level abstraction and thus are significantly 'cheaper' from a computational point of view. For the same reason, early experiments in urban design were grid-based with a high level of abstraction, but the condition changed in the second half of the 1990s with computing power becoming more affordable and ubiquitous, which made experiments with CAGD agents possible (see Coates [40]).

Early experiments were characterized by a speculation about the possibilities of an agent-based approach in architecture and about methods of Swarm Intelligence to generate architectural form (see Miranda and Coates [135]).

The computation constraint is further enforced by different research cultures: natural science consists of traditional simulation fields where high performance computing (HPC) using large clusters is a common approach to run large-scale simulations, which can provide significant results. In comparison, architectural design and construction are considered 'new-to-simulation' fields where the standard is desktop computing due to its predominance in the professional field. While academic research could theoretically utilize existing HPC clusters, the relevance of the methods are currently predicated to a large extent upon their transferability to practice. This situation might change in the future with the advent of cloud computing, which could make high-performance computing available to architectural design practices.

By the end of the 1990s, complexity research was in full swing and first successful applications proved the validity of the individual-oriented ABMS method. In parallel, the concepts have leapt over into the discourse of architecture theory and design research. In the mid 2000s, the Architectural Design issue "Emergence: Morphogenetic Design Strategies" guest-edited by Weinstock, Hensel and Menges [79], made the topics of complexity, adaptation, emergence and self-organization accessible to a wider audience in the architecture field. The authors also provided a basic definition of the term 'emergence' in terms of its main applications, which is maintained here and largely corresponds to the distinctions between ALife and Cybernetics: "Emergence is both an explanation of how natural systems have evolved and maintained themselves, and a set of models and processes for the creation of artificial systems that are designed to produce forms and complex behavior, and perhaps even real intelligence" [79, p. 6].

Next to GA, one of the most popular approaches to instrumentalize emergence in the architecture field was the ABM proposed by Reynolds [156] involving locomotive agents where steering rules allow ‘vehicles’ to reposition and reorient based on internal and external stimuli (see Section 3.1.4.1). The collective ‘flocking’ behavior, which is based on the fundamental micro-level agent behaviors and group behaviors (cohesion, separation, alignment), was exploited as form-generating principle or as a principle for self-organization. This model became the basis for further developments in the field such as the one by von Mammen and Jacob [201], Shiffman [178], Snooks [181], and as a current implementation by Groenewolt et al. [72].

In recent years, with the advent of robotics in architecture research (starting in 2005 at the ETH in Zurich), hardware prototyping platforms, such as Arduino™ and Raspberry Pi™, availability of sensors, and first commercially available mobile robots, the agent-based concept has been extended beyond the digital domain into the domain of physical builders.

3.4.2 Classification of Agent-Based Systems in Architecture

The developments described above would suggest a classification of agent-based systems in architecture with respect to the *entities* that the agents in the model represent. Previous architecture-specific classifications were proposed, for example, based on categorizing designer intentions, based on application areas, or based on the stages of knowledge management, from retrieval to collaboration.

Snooks [182] proposed a classification of agent systems based on “the role of architectural decisions or intent within the [algorithmic] process” and based on the kind of evaluation that is applied to the emergent outcome. The author identified four categories distinguishing between ‘arbitrary’ behaviors (in the architectural sense) that “self-organise to create emergent patterns, organisations or form”, which are either (1) evaluated based on the pattern itself or (2) on architectural criteria, and specific behaviors, which are either evaluated purely based on (3) the global outcome or based both on (4) the local condition and the global outcome.

Baharlou [10] proposed a classification of agent-based systems based on “design agencies,” that is, discipline-specific drivers of design and distinguished between three main categories: (1) environmental and spatial effectiveness, (2) performative and structural approaches, and (3) fabrication approaches. The first category includes applications of spatial planning both on the urban and building scales that deal with perception and movement of people through man-made spaces both for design generation and evaluation. The second category covers applications of performance simulation and optimization, such as the self-organization of structural elements. The third category focuses on fabrication integration, such as by attributing agents with knowledge about fabrication tools.

Beetz et al. [16] proposed a broader classification consisting of three categories that, in addition to (1) MAS for simulation and performance of building designs, which corresponds to Baharlou's second category, also includes (2) MAS for knowledge capturing and recognition in drawings and sketches and (3) MAS in collaborative environments. The second category covers the category of today's software bots and crawlers, which are ubiquitous on the internet, but in the architectural knowledge capturing context seem to have never left academe. The third category covers design decision support systems where agents wrap legacy applications, exchange information in standardized formats, and capture domain knowledge such as in the previously mentioned Kratzer et al. [95].

The following classification, which is entity-oriented, identifies four distinct categories and is compatible with the previously introduced general classifications proposed by Macal [121] and Reynolds [158] (see 3.2.3) and also extends the previous classifications by Beetz et al. [16], Snooks [182], and Baharlou [10].

3.4.2.1 Agents representing design participants

First, from the perspective of the participants and stakeholders in the design process of the built environment, it is easy to see the built environment as the emergent outcome of the interactions of many different actors [40, p. 653]. ABMS is therefore the 'natural' way to model and simulate the network of interactions and the actors in the design process and their domain-specific intentions. The premise of modeling the collaborative design process using ABM is that it can provide insights into the dynamics of the design process. In this application, the goal might be to enable design integration, to make the design process more flexible, or to make it more resilient to unforeseen external influences.

Anumba et al. [8] describe the potential of ABMS in collaborative design and, as an example, present the key features of an agent-based system for the collaborative design of portal frame structures. Another example within the same paradigm constitutes the modeling (and design) of the supply chain in construction using ABMS (see [215]). This category is congruent with Beetz et al's third category.

3.4.2.2 Agents as building elements

Second, from the perspective of building elements, buildings are aggregate systems where individual building elements interface with their respective neighbors through joints and connections. This interfacing is predicated on the rules and constraints that are inherent to the building elements themselves. These locally defined rule sets can be utilized in a bottom-up approach for negotiating the shape and location of these elements within the larger context of an assembly according to well-defined performance criteria [172, p. 178]. This application is based on "embedding fragments of intelligence and design sensibility within architectural objects so that they might learn how to search through design possibilities autonomously" [96, p. 97]. The premise of this approach is to instrumentalize the concept of self-organization, for example to generate configurations of building elements that would be difficult to model in conventional ways.

Examples are the project 'Lamella Flock' by Tamke et al. [188] where a spatial structure is defined through self-organization of building elements; and the 'Landesgartenschau' project by Schwinn et al. [172] in the context of segmented shells (see chapter 7). The last example is also an example for hybrid approaches combining design integration and self-organization. Further examples of this category include the work by Gerber et al. [64] in facade and environmental design, and research into fabrication-informed design using ABM by Baharlou and Menges [11]. This category is compatible with Snooks' third and forth, and also overlaps with Baharlou's second and third category, while Beetz et al's corresponding category only covers the goal-oriented part of this category.

3.4.2.3 Agents as digital builders

Third, from the perspective of agents as digital builders, agents can either represent real fabrication and construction equipment or be virtual 'builders' (in Reynolds' sense). In both cases they modify their digital environment by depositing material or changing existing geometry in a CAGD environment. Digital representations of space can act as a learning environment for rule-based agents, who might represent design intentions. In this scenario, the interaction between the agent and the environment can be represented as "a form of architectural designing, as a set of simple parallel rules from many interacting virtual designers" [40, p. 651]. The premise of this approach is to use the emergent capacity of ABM to uncover novelty based on simple design rules and decisions, especially in the early stages of the architectural design process.

Early examples for this generative approach for design exploration are the works by Coates and Schmid [40]; and more recently by Leach [112] in urban design and by Snooks [181] in building design. This category is again compatible with Snooks' categories 2 to 4 and stretches across all categories proposed by Baharlou. An application where digital agents represent real construction workers would fit into Beetz et al's third category, while the notion of virtual agents as builders seems to be absent from Beetz's classification.

3.4.2.4 Agents as physical builders

The perspective of virtual agents as builders has recently been extended into the physical domain as an approach to adaptive fabrication using cyber-physical systems. In the fourth domain, numerically controlled machinery (such as industrial robot arms and mobile robots) deposit, produce and alter a construction while being in constant feedback with the as-built and the environment. The premise of this application is to allow for physical adaption during the materialization process including (1) path-correction and adaptation of predefined tool paths, (2) controlling of construction machinery without explicitly programming it, (3) swarms of autonomous builders, and eventually (4) (semi-) autonomous construction machinery navigating the unstructured environment of a construction site while collaborating with human workers as part of a socio-technical system.

Examples for these sub-categories are (1) the work by Vasey et al. [195] and by Yablonina and Menges [216][1]; (2) Johns et al. [84], Snooks and Gwyllim [183], and Brugnaro and Hanna [33]; and (3) the TERMES project by Werfel, Petersen and Nagpal [205]. Subcategory (4) is the subject of ongoing research in many fields at the intersection of robotics and production engineering, computer science and artificial intelligence, systems engineering and sociology. This category is absent from the designer-centered classification by Snooks, as well as the application- and knowledge-oriented classifications by Baharlou and Beetz et al.

3.5 Further Research

In this investigation agents are conceptualized as building elements and as virtual builders (categories 2 and 3 in the categorization above), thereby instrumentalizing self-organization and emergence.

The methodology for defining agent system constructs from real world segmented shells focuses on systems that are composed of building elements. In such an application, the premise is two-fold: first, ABMS can simulate and solve complex nonlinear design problems involving multiple locally interacting entities, which pursue multiple objectives, by instrumentalizing ‘self-organization’ as part of a Swarm Intelligence approach; second, ABMS can be used to explore the design space of hypothetical buildings systems by instrumentalizing ‘emergence’ as part of a bottom-up design approach.

The methodology can be extended to agent systems of virtual builders, with design principles representing real or imagined builders and their behaviors. In this case, the premise is that the impact of hypothetical design decisions can be explored using ABMS, again, by instrumentalizing ‘emergence’ as part of a bottom-up design approach.

In addition to discussing the relevance of these application domains for the production of architecture, challenges and open questions are outlined in the following.

---

[1] In the MoRFES series of projects “robots produced a three-dimensional architecture-scale artifact following a pre-calculated winding syntax and using a real-time path correction algorithm relying on a camera-based tracking system” [216, p. 65].

3.5.1 Self-Organization in Architecture

According to DeWolf and Holvoet [47, p. 7], organization is the arrangement of selected parts to fulfill a specific objective; and self-organization is this process unfolding without external control leading to increasing structure or order in a system. This process “restricts the behavior of the system in such a way as to confine it to a smaller volume of its state space [which] is called an attractor” (ibid.). Engineered examples of such systems are MAS, which are defined as loosely coupled networks of problem solvers that work together to solve problems that are beyond their individual capabilities [94, p. 3], such as ‘ad-hoc’ sensor networks and robot swarms. Koudari et al. [94] provide a review of applications.

The premise of self-organization to increase order and to solve problems is a promising prospect in architecture and particularly relevant in agent systems where agents represent building elements. Consequently, at least three possible application scenarios can be envisioned using ABMS:

1. To control geometric complexity, where traditional modeling of individual entities becomes impractical; instead, simple rules can be implemented to control how a building element should adjust given changes in its inputs, for example, its environment or parameters.
2. To find a steady-state equilibrium between multiple interacting entities each pursuing their own goals and attempting to satisfy their individual objectives. In this case, ABMS is used as part of a Swarm intelligence approach towards solving design and engineering problems where design objectives and constraints have been defined. In this scenario, the difference to PSO would be that each ‘particle’ corresponds to a building element as opposed to a candidate solution to the optimization problem.
3. Following from the previous scenario, in the third scenario the dynamic equilibrium state (attractor) of design decisions and constraints remains active across the design stages thereby dynamically incorporating design changes through adaptation and convergence to new equilibrium states. This scenario is based on the ‘robustness’ criterion of self-organizing systems in terms of adaptability in the presence of perturbations and change [47, p. 8].

The first two applications have been as demonstrated in the ‘Landesgartenschau’ project, where each segment in the segmented shell corresponded to an agent in the system. The behaviors of the agents were informed by multiple design objectives (constraints), in this case mostly related to geometry and producibility. Regarding geometry, the objective was to cover a double-curved surface (the shell geometry) with planar elements made from cross-laminated Beech veneer in a ‘water-tight’ manner. The geometric complexity of each segment, which was generated using the Tangent Plane Intersection (TPI) method, was controlled as part of the agent definition and interaction topology. Regarding producibility, objectives were to constrain the size of the segments to the available panel dimensions, as well as to constrain the diameter of the segments to the diameter of the available fabrication equipment, in this case a transportable industrial robot cell. These objectives could be met through the design of steering behaviors. Such a design process is more concerned with steering design as a dynamic process, similar to how "a shepherd would drive a herd" [88, p. 23], rather than achieving an optimal end state (see also Section 4.3.4).

A few open questions and challenges were identified in the Landesgartenschau project with respect to the first two applications, which form the starting points for the investigation pursued in the context of this research:

- How to define behaviors and interactions on the local level that 'converge' to a desired outcome?
- How to ensure that the solution space defined by design objectives and constraints overlaps with the design space of the building system?
- How to implement adaptation in agent behaviors over time?
- How to design steering behaviors based on structural evaluation?
- How to enable designer interactivity during the self-organization process?

The last point has been addressed in the meantime by Groenewolt et al. [72].

3.5.2 Emergence in architecture

Again starting with the definition by DeWolf and Holvoet [47, p. 3], a system exhibits emergence when its properties, behavior, structure, or patterns at the global level (emergents) dynamically arise from the interactions at the local level and when those emergents are novel; in other words, when they cannot be traced back to the individual parts of the system.

The characteristic property of emergence to produce novelty and the insight that ABMS can provide into the functioning of an emergent system emphasize two aspects of emergence that are particularly relevant for an application in architecture: to provide insight and to produce novelty:

1. Being the 'natural' way to model emergent systems, ABMS allows to heuristically use the computer to develop theories, as envisioned by von Neumann [74, p. 164], by hypothesizing about the inner workings and mechanisms of a complex system in a "virtual laboratory" [124, p. 156]. Holland termed this approach "exploratory modeling" [57, p. 61]. Using 'exploratory modeling' in architectural design, the impact of design decisions and the sensitivity of

the system can be tested thereby providing insight into the relation between micro-level design decisions and observed macro-level behaviors of the system. In such an application, instrumentalizing emergence can support the decision making process.

2. Next to being a necessary property of emergent systems, the production of novelty is a fundamental aspect for design in general. The corresponding concept is "design exploration", as the divergent part of the design process, which aims at uncovering novelty by changing the boundaries of the solution space before convergence methods can be applied (see 2.1). The emergent property of the outcome makes ABMS valuable in terms of design exploration: the ability to test alternative outcomes by modifying rules, parameters and agent attributes and to generate 'novelty' with respect to the design primitives.

It is this second aspect of emergence in architecture that is particularly relevant for early architectural design stages, where ABMS can be used for design exploration before any optimization methods can be applied due to the absence of optimization goals in early design.

Furthermore, in early design stages architectural models are minimalistic capturing only "the most salient features" of the design [124, p. 156]. These models are then iteratively refined over the course of the design process and become more and more detailed. This process thus runs exactly parallel to the development of an ABM starting from a minimalist description and subsequent iterative development [132, p. 24], which makes ABMS a suitable method in the architectural design process as the model scales parallel to the development process—it is co-developed.

In early architectural design, the model can be considered a 'Generator' model, where "little is known about the system of interest" [75, Sec. 2.18] translates as novelty in the model is explored with respect to design primitives. The model then proceeds through 'Mediator' models over the course of the design and engineering stages where "the system is only moderately understood" (ibid.) translates as iterative refinement and more detailed description during design development. The equivalent of a 'Predictor' model in the architectural design context would thus be a fully developed 'digital twin' that can predict how the building will perform structurally, environmentally, financially, etc.

However, a few challenges have been identified in literature, when it comes to instrumentalizing emergence in design and technical applications, which mostly revolve around the micro-macro-level relationship, which different authors have expressed in similar ways:

- Heath and Hill [74, p. 165] describe this challenge as "engineering a cell's logic at the local level in hopes that it will create the desired global behavior."

- Brooks [30, p. 10] concedes that "as with heuristics there is no a priori guarantee that this will always work". Therefore, systems with useful and interesting emergent properties require "careful design of simple behaviors and their interactions."
- Jennings and Wooldrige [82, p. 290] state that, in general, "a better understanding is needed of the link between behavior of the individual agents and that of the overall system."
- According to Werfel et al. [205, p. 754], "predicting high-level results given low-level rules is a key open challenge; the inverse problem, finding low-level rules that give specific outcomes, is in general still less understood."

One approach to address this important issue could be to genetically evolve behaviors, that can lead to meaningful associations between local rules and high-level behaviors as demonstrated by Vogel [198] and suggested by Nguyen et al. [141] in terms of genotype-phenotype mapping. Alternatively, machine learning approaches can be used not only to implement adaption but also to learn behavior parameters that lead to meaningful micro-macro relationships.

Further domain-independent challenges that also apply to architectural design are related to increasing transparency of models, reproducibility, and validation of models with real world data [121, p. 151]. These challenges can be addressed by implementing a domain-specific, yet flexible, modeling framework in an accessible software platform, as well as by developing models derived from real world data gathered, for example, through case studies.

Summarizing, within the context of an application in architectural design and engineering, the potential of ABMS can be seen as (1) an approach for achieving self-organization and for meta-heuristic optimization as part of a 'Swarm intelligence' approach; and (2) as an approach for exploratory modeling approach as well as for design exploration: by productively using emergence in architecture.

4

Shell Structures

4.1 Historical Overview

This first section provides an architect's view on the history of man-made vaulted structures starting from pre-historic and classical times to the 20th century. Following specific examples in the historical record, the expansion of form-defining methods from purely geometric methods to physical, mathematical/analytical, and numerical methods is outlined.

4.1.1 Pre-history, Antiquity and Middle Ages

The vaults and domes built by the Romans, such as the Temple of Mercury in Baiae, Italy (1st c. BCE) and the Pantheon in Rome (around 126 CE) are commonly considered the earliest ancestors of modern thin-shell structures [2, p. 33] (Fig. 4.1).

Figure 4.1: Examples of ancient shell structures. Left: Temple of Mercury in Baiae, Italy. From [152] by RaBoe (2010). Right: Pantheon in Rome. From [147] by Pirmann (2012).

The technological achievements of the Romans notwithstanding, the history of domed structures can certainly be traced back further to earliest human endeavours of building structures for ceremonial, burial, storage and shelter purposes. A striking example are the well-known megalithic domed structures in Sardinia called 'Nuraghe' that were built between 1900 and 730 BCE [48, p. 167]. They are double-layer, dry-stone structures with a rubble in-fill, where the exterior layer has the shape of a truncated cone and the interior layer is beehive-shaped forming a 'corbel vault'. Constructed by stacking minimally processed stones in an ashlar pattern and by offsetting consecutive courses of stones towards the centre until they meet at the apex of the dome, they reached hights of up to 20 m [48, p. 172]) (Fig. 4.2).

Figure 4.2: Megalithic vaults. Left: The nuraghe Goni, Sardinia. From [36] by Camedda (2008). Right: The corbel vault of the Nuraghe Is Paras. From [42] by Consorzio Turistico dei Laghi (2014).

These structures didn't require mortar and thus their stability (many are still standing after 3000 years) is based purely on self-weight, which might amount to several tons, and on friction between the stones. The characteristic features, which would allow to consider these vaulted spaces, and in particular the inner loadbearing layer, predecessors of the aforementioned Roman shell structures are:

1. Corbel vaults are surface structures, where the surface is, at the same time, the main load-bearing element and the enclosure.
2. Loads, predominantly the dead load due to self-weight and the additional weight of the surrounding earth or rubble, are transferred inside the thickness of the surface to the ground, essentially following a similar principle as contemporary unreinforced masonry structures designed using Thrust-Network-Analysis (see 4.3).
3. The loadbearing surface is double-curved thereby avoiding buckling of the surface.
4. The spaces that they create are similarly curved and vaulted.

These criteria can almost serve as a definition of shell structures, as they apply to the megalithic unreinforced shells of the late Bronze age and to many of the form-found shells of the 20th century. Furthermore, they also apply to the geometric (cylindrical and hemispherical) vaults from Roman times, which were built using *opus caementitium*, an early form of concrete; to the pointed arches and vaults in late medieval Gothic cathedrals, which were formed by two circular arcs in order to vary the height and span of a vault independently [2, p. 33]; and to the dome structures built during the Renaissance, most notably the double-layer dome of the Basilica di Santa Maria del Fiore in Florence by Filippo Brunelleschi (1377-1466).

4.1.2 Renaissance to 19th Century

Brunelleschi's dome was a model for Michelangelo (1475-1564) when he took over the building site of St. Peter's in 1547 and with it the design of its dome. In the original design, Donato Bramante (1444-1514) proposed a solid hemispherical dome similar to the Pantheon, but in contrast to the Pantheon, where the dome is supported all around its perimeter, it was supposed to rest only on the four columns at the intersection of the nave and the transept. Michelangelo, seemingly aware of the misalignment of form, forces and support conditions, tried to reduce the weight of the dome by following Brunelleschi's approach of a hundred years before and proposed a double-layered design. In the final design, Giambattista della Porta (1535–1615) increased the height of the dome in order to further reduce the outward thrusts at its base and he also introduced several iron chains to carry the tensile stresses in the lower part of the 41.9m diameter dome [2, p. 36].

In a parallel historical development, the Anglican Church tried to assert its independence from Rome under Henry VII with the act of Supremacy of 1534. After mistreatment and defacing of St. Paul's during the Civil War—part of the aftermath of the split from Rome—and finally after the Great Fire of London 1666, where many churches were destroyed, including St. Paul's, a redesign and reconstruction became necessary.

Sir Christopher Wren (1632-1723), a scientist by training and one of the founders of the Royal Society in London, was commissioned to design and oversee the execution of the many churches, 52 in total, including in 1669 the redesign of St. Pauls [146, p. 291]. Having been influenced by French representational architecture, such as Perraults Louvre-Facade for Louis XIV in Paris, and by Dutch classicist architecture through Vingboons' engraved publications, he designed a protestant cathedral, in a style blending the classical and the baroque [146, pp. 281 & 288].

Many comparisons have been made between St. Paul's and St. Peter's. While Wren's intention was not explicitly to rival the design of St. Peter's, it was certainly sporting one of the tallest domes at the time and was widely received as asserting the power and independence of the Anglican Church [163, p. 453].

Robert Hooke (1635-1703), a scientist and engineer, worked with Wren during the design of the 33m diameter dome of St. Paul's, which was topped out in 1708. Hooke is most known for his formulation of the law of elasticity, which states that the force needed to extend or compress an elastic material, such as a spring, is proportional to the distance. He published the law in 1676, which since bears his name, as part of his ten 'Inventions' in form of a Latin anagram, which when deciphered reads:

> "Ut tensio, sic vis."
> (As the extension, so the force.)

As part of the ten 'Inventions', he also published another important finding in the form of an anagram:

> "Ut pendet continuum flexile, sic stabit contiguum rigidum inversum."
> (As hangs the flexible line, so but inverted will stand the rigid arch.)

Based on his law, which is commonly called Hooke's law of inversion, he proposed the use of a hanging chain model to determine if the shape of St. Paul's dome was stable, that is if the shape of the catenary arch was inside the masonry, which is shown in one of Wren's sketches [2, pp. 35-36].

For a similar purpose, the method was later used by Giovanni Poleni (1638-1761) in the 1740s to assess the stability of the hundred-year-old dome of St. Peter's in Rome, which had developed cracks. However, Poleni added weights to the chain to more accurately represent the effect of the voussoirs' weight. Again a century later, Heinrich Hübsch (1795-1863) used the method as a design tool to determine the weights of voussoirs needed to achieve the desired shape of an arch or vault [2, p. 36].

The concern with the relationship between form and stability makes Hübsch's form-finding approach for a compressive vault more akin to the aforementioned nuraghe than to the realisation of geometric primitives as was prevalent from antiquity to the Renaissance. This relationship becomes apparent when comparing his sketches circa 1835 to a nuraghe section (Fig. 4.3).

Addis [2] retraces the use of physical form-finding methods from Hooke via Poloni and Hübsch to Antoni Gaudí (1852-1926) who used the law of inversion not only on two-dimensional but also on three-dimensional networks of strings using sand bags as weights; and on to Heinz Isler (1926-2009) who, as the "last of the great concrete shell builders of the twentieth century" further expanded the law by using sheets of cloth (rather than a chain) to make hanging models.

The characteristic feature of the method that each protagonist developed further, and which allowed them to use the law of inversion as a method for the design

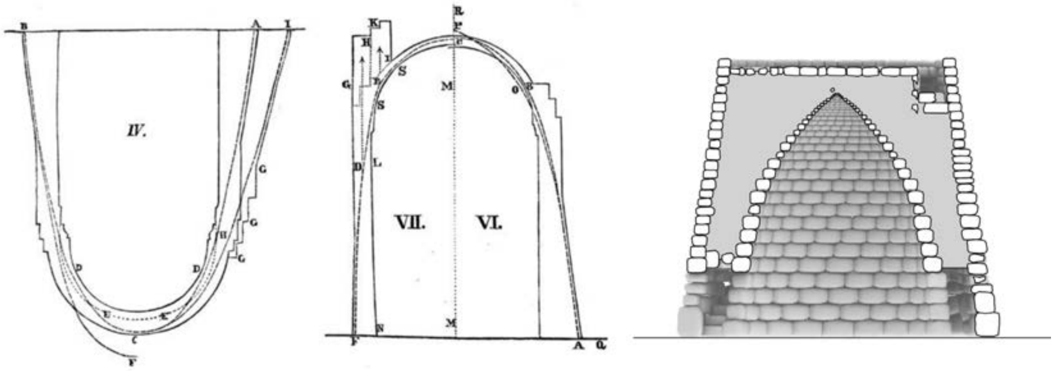


Figure 4.3: Juxtaposition of the sketches of Heinrich Hübsch investigating the form of vaults using the hanging chain method with a section of a nuraghe tower. Left and center images from [2, p. 37] by Hübsch (c. 1835).

of full-scale structures in the first place, is that certain aspects of the structure, such as its static equilibrium and the funicular shape, are scale-independent and thus independent of materials involved. Therefore, they can be scaled up linearly to predict structural behaviour of compression structures such as funicular arches, vault and domes at full-scale [2, p. 34]. This is in contrast to scale-dependent properties that cannot be scaled up linearly, such as the strength and elasticity of the material and the buckling behaviour of a column or thin shell (see Section 3.2).

Strikingly, both of Hooke's laws of elasticity and of inversion are still in use today and constitute fundamental principles in the numerical methods for the design of contemporary form-found shell structures (see Section 4.3.1).

4.1.3 20th Century

After a few generations of shell builders had proven the reliability of the law of inversion and had expanded on its methods, the advent of new materials in the early 20th century, most notably of reinforced concrete replacing voussoirs and stereotomy, made a new era of experimentation and a new category of shell structures possible: thin shells.

This era around the mid-20th century, which is sometimes called "Structural Expressionism" [164, p. 259], was characterised by the work of Eduardo Torroja (1899–1961), Pier Luigi Nervi (1891–1979), Eero Saarinen (1910–1961), Felix Candela (1910–1997), Eladio Dieste (1917–2000), who was particularly recognised for his 'Gaussian shells' made of bricks, and Heinz Isler.

A striking example for a form-found concrete shell is the roof for the Naturtheater Grötzingen in Aichtal near Stuttgart (1978) by Isler in collaboration with Michael Balz. It spans 42m with a shell thickness ranging from only 9cm at the top to 35 cm at the five support points [13, p. 31]. While Isler used physical form-finding methods to establish the shape of his shells, Torroja, Nervi, Candela and Dieste designed their shells using geometric primitives – not unlike the geometers of antiquity – that could be defined mathematically. This included shapes such as hemispheres, cylinders, cones, ellipsoids (Torroja and Nervi), and hyperbolic paraboloids (HP) (Candela and Dieste).

The ability to describe the shape of shells that are part of this category of mathematical or geometric shell structures was the pre-condition for analysing their behaviour before the advent of computers. Such shapes were also easier to set out on site as the dimensions could be easily calculated [2, p. 34]. Furthermore, the particular benefit of HP shells was that their formwork could be constructed completely from straight elements, which greatly facilitated construction as described in chapter 1.

Sasaki [164, p. 260] describes the distinctive changes of reinforced concrete shells during the 20th century:

> “...from the enclosed and semi-spherical shell covering the 1932 Algeciras Market Hall by Manuel Sánchez Arcas and Eduardo Torroja, to the dynamic and open Hyperbolic Paraboloid (HP) surface of the 1959 Bacardi Rum Factory by Félix Candela, to the formative and lively free-curved surface found in the 1961 TWA Flight Centre by Eero Saarinen.”

Next to the classical geometric and form-found shells, Saarinen serves as an example for the third category of shell structures: free-form shells, which do not exclusively follow principles of geometry or load adaptation, and continue to be the most prevalent category of shells to this day.

Concrete shells, however, have lost their appeal towards the last quarter of the 20th century due to rising manufacturing costs, lack of skilled workers, inefficiencies of on-site work, etc. [164, p. 259] (see also [114, p. 1]). The focus of the development of shell structures thus moved on to grid-shells, which offer the possibility of prefabrication as a way to lower the expenditures of on-site labor.

This development starts with the work of Frei Otto (1925-2015) in the 1960s and 70s at the Institute for Lightweight Structures (IL) at the University of Stuttgart who used physical form-finding methods, such as the hanging chain model or inflating balloons (similar to Isler), but also novel scale-independent methods such as soap-films to determine the equilibrium shape of minimal tension structures where gravity plays a minor part in establishing the shape [2, p. 39]. Most notable are the design developments for the roof of the Multihalle in Mannheim (completed in 1975), a strained gridshell, in collaboration with Ove Arup & Partners and the design for the roof of the Olympic Stadium in Munich in 1972, a tensioned cable net, in collaboration with Fritz Leonhardt.

In the Mannheim Multihalle, the load-bearing elements were initially laid out flat on site and then bent into shape and locked in position by locking the joints. It is a strained grid-shell where the load-bearing elements are stressed not only through self-weight, but also through the bending of the elements. The structural system of the roof of the Olympic stadium, in turn, is a tensioned cable-net clad in acrylic glass.

In contrast to the shells discussed so far, where the load-bearing elements are under compression and to a certain degree under bending, in the case of the Olympic stadium roof the load-bearing elements are under tension. This distinction is expressed in the terms 'form-active' for tensile surface structures such as tents and cable nets, which adjust their form based on changing load conditions, and 'form-passive' to describe compressive structures, which do not noticeably change their form due to changing load conditions [210, p. 22].

4.1.4 21st Century

The advent of novel materials, such as reinforced concrete and fibre-reinforced polymers (FRP), has characterised the evolution of shell structures during the 20th century by opening up a new range of possibilities - even a new category of shells - and resulting in the iconic funicular shell of the 20th century. Similarly, the advent of digital computation and numerically controlled fabrication tools towards the end of the 20th century characterised and continues to characterise the further development of shell structures in the 21st century by opening up new possibilities for shell design.

While physical form-finding is based on self-weight as the main design criterion, numerical form-finding methods, such as force density method, thrust network analysis, dynamic relaxation or particle spring method, now allow for design, simulation and optimisation of shell structures also with regards to other design criteria. For example, when minimising self-weight as part of lightweight design, other load cases such as wind or snow load become more relevant and bending stiffness will have to be introduced since there is no ideal shell geometry that reflects those dynamic loads.

Grid-shell have thus made a significant leap in the last 20 years as demonstrated by numerous contemporary shells such as the Roof of the Great Court of the British Museum in London (2000) by Fosters and Partner and Buro Happold Engineers, the Savill Building at the Windsor Great Park (2006) by Glenn Howells Architects with Buro Happold, the MyZeil shopping Mall in Frankfurt, Germany (2009) by Massimiliano Fuksas with Knippers Helbig Engineers, the Dutch National Maritime Museum (2011) by Ney & Partners or, more recently, the Chadstone Shopping Centre (2016) in Melbourne, Australia, by the Buchan Group, just to name a few.

What contemporary shells and historical examples have in common beyond their double-curvature, is that they are an efficient and sometimes even the only way to enclose a given spatial programme, while the design drivers, methods and materials, and with it the possible expression of shells, have obviously changed over time. Current technological developments, such as the continuing sophistication of numerically controlled fabrication tools, innovation in materials, the ubiquity of computation and thus the dissemination of computational design possibilities provide an evolving context for the design and materialisation of shell structures.

4.2 Categorisation of Shell Structures

With the broad outline of the history of shell structures in mind, the objective in this section is to establish a taxonomy of shell structures. In a first step, a definition of what constitutes a shell structure is proposed. In the following steps, categorisations based on stress-state, form, continuity, and material are introduced.

4.2.1 Shell definition

One commonly accepted definition is that shell structures are curved surface structures with two main characteristics [4, p. 1], [210, p. 21]:

1. the dimension perpendicular to the surface is significantly smaller than the other two, making them ‘thin’, without necessarily stating how thin they have to be;
2. loads are supported mainly through membrane stresses in the plane of the shell surface.

This definition, however, also applies to tensile structures such as membranes. Therefore a first categorisation is introduced that distinguishes membranes from shell structures.

4.2.2 Categorisation based on stress-state

As a generalisation, shell structures are part of the category of surface structures, which also include tensile structures. The distinction is that shell structures are mostly rigid, ‘form-passive’ structures, whereas tensile structures are “form-active” structures. Form-passive structures don’t significantly change shape when external loads are applied, and they mostly work in compression and bending. In classical shell design, self-weight is the predominant load case and thus physical and numerical form-finding methods based on the “law of inversion” are applicable. The ancestors of shells are geometrical stone vaults of the Renaissance and Romans, as well as the gravity-informed vernacular vaults from before (see Section 4.1.1).

Tensile structures, such as membranes and cable nets, on the other hand exclusively work in tension and they do significantly change and adapt their shape to external loading thereby establishing a new equilibrium state. In tensile structures, self-weight does not usually constitute a main forming factor and thus form-finding methods that produce minimal surfaces, such as soap-films or algorithms that minimise mean curvature, are applicable. The ancestors of membrane structures are lightweight tent structures that have been used in nomadic societies for millennia.

4.2.3 Categorisation based on form

In shells that are based on geometric primitives, from the classical shells of the Romans to the domes of the Renaissance up until the shell designs by Hübsch (see Section 4.1.2), the shell surface has to provide some bending stiffness in order to resist buckling. 'True' or 'ideal' shells, on the other hand, are considered those shells that are funicular, that is their form is based on the shape of the inverted weighted string or net, and, theoretically, do not require bending stiffness.

In addition to geometric primitives and form-finding based on the principle of inversion, shells can also be designed based on the principle of elastic deformation (buckling) resulting in elastic shapes; or they can be designed based on the principle of pressure difference resulting in pneumatic shapes. Similar to funicular shells, these shells have to be form-found using physical and numerical methods. Whereas in 'ideal' shells only compression forces occur under dead load, which usually is self-weight, in the alternative 'geometric' and 'free-form' shells also bending and possibly tension forces occur under dead load. It is important to note that a 'free-form' shell can also be considered ideal under certain circumstances: namely for those loads for which it is funicular [210, p. 26].

From these considerations follows a first categorisation that, similar to Adriaenssens et al. [4, p. 2], can be made based on the shape generation method of the shell surface:

1. 'Geometric' shells, also called mathematical or analytical shells, can be described using mathematical methods;
2. 'Form-found' shells of funicular, elastic or pneumatic shape can be designed using physical or numerical methods;
3. 'Free-form' or sculptural shells are the result of shell design approaches that, in addition to self-weight, integrate other design parameters.

As mentioned above, the more a shell is designed for lightweight, the less the dead load due to self-weight becomes a determinant factor and with it the funicular shape the ideal form of the shell. Similarly, the more efficient shells are with respect to the ideal membrane stress state, the more sensitive they become to buckling and to initial imperfections, and the more sudden is the buckling collapse [210, p. 31]. Thus 'real' shells that are subject to dynamic loads and to imperfections in construction have to deal with bending stresses, especially when designing for earthquake-prone regions [164, p. 260].

A term often used in the context of form finding is the catenary, as the shape of a string or chain under self-weight, that is under constant load per unit arclength. Williams [210, pp. 30-31] shows that the catenary shape for an arch or the catenoid shape for a shell of uniform thickness is not “particularly good” due to the stress concentrations at the supports. Rather, a shell of uniform stress but variable thickness is preferable which allow theoretical shells of revolution of up to 2.5 km span for concrete only considering its self-weight (ibid.).

4.2.4 Categorisation based on continuity

Shells can be constructed as a continuous surface as in the case of reinforced concrete shells, where the load-bearing layer is at the same time the spatial enclosure, but also as discrete lattice or grid shells where the material is concentrated in a relatively fine grid compared to the extents of the shell surface. In this case, the enclosure can be realized by an additional non-load bearing surface layer such as glass.

As part of the category of grid shells, a further distinction has to be introduced between strained and un-strained grid shells: in strained shells the curvature of the elements is achieved through bending, thereby introducing additional stresses in the loadbearing elements. This approach is particularly suited for on-site construction where elements can be laid out flat and then lifted into place, as in the case of the Mannheim Multihalle (1975), or lowered into place, as in the case of the Downland timber shell (2002). In this approach, the form of the shell is the result of the elastic forming capacity of the material used, usually timber. After the grid has found its shape the pin-jointed connections in the grid are locked; here, identical clamping mechanisms can usually be used. In strained grid-shells, part of the stress capacity of the elastic material is used up by the bending stress. In the case of wood, this is gradually released through creep and becomes available for accommodating membrane stresses [3, p. 98].

In un-strained grid shells, the members of the shell are usually prefabricated in their curved state or, in the case of faceted grid shells, as straight members. Other than the stresses due to dead-load and dynamic loading, no additional stresses occur. Here the connection of the members, for example in triangulated grid shells, is a bigger challenge than in strained grid shells and usually leads to a high amount of unique connector elements.

Similarly, within the category of surface shells a further distinction can be made between continuous vs. discrete or segmented shells: in continuous surface shells, as the name suggests, the surface is continuous, which usually corresponds to a continuity of material properties, mainly stiffness. Examples are reinforced concrete shells, such as the ones by Candela or Isler or monocoque shells made from fibre-reinforced polymers. In segmented shells, on the other hand, the surface is discontinuous, interrupted by the joints between the elements, and with it the stiffness continuity. This means that in segmented shells, the forces in the shell have to be transferred via the joints between the segments, which includes axial forces in the plane of the surface, but might also include bending and forces acting normal to the surface, such as in the case of suction forces under wind load.

While the distinction between strained and un-strained is usually used in the context of grid shells, it is worth noting that surface shells usually are un-strained. Exceptions are the “Self-strutted Geodesic Plydome” by Fuller (1959) and, more recently, the “ICD/ITKE Research Pavilion 2010” by University of Stuttgart, which is a segmented surface shell made from individual elastically bent plywood strips [56, p. 47]; and the prototype “Bend9” by La Magna and Schleicher (2016), which is a multi-layered arch built from bending-active plates [107, p. 182].

For disambiguation, while grid shells are also sometimes called ‘discrete’ shells, the ‘discreteness’ property also applies to masonry shells and to segmented timber shells, which are the focus of this research. In order to distinguish between discrete grid shells and discrete segmented shells, in the following the term ‘discrete’ is generally avoided unless both segmented and grid shells are addressed. Instead, the term ‘segmented’ is used in the context of surface shells and ‘grid’ in the context of grid shells.

Finally, within the category of segmented shells, a distinction can be made between plate shells that approximate the double-curvature of the shell surface through planar facets, such as the Landesgartenschau Exhibition Hall (2014) and non-plate shells, which consist of non-planar segments, that more accurately retrace the curvature of the shell surface, such as the Rosenstein Pavillon (2017), which was made from functionally graded concrete.

4.2.5 Categorisation based on material

A characterising feature of shell structures is the material predominantly used in their construction, such as stone, concrete, wood, or Fibre-Reinforced-Polymers (FRPs). To a certain extent, a correlation can be established between the type of shell, the shell geometry and the material used for its implementation. For example, the use of stone suggests the design of a segmented surface shell designed for self-weight resulting in a funicular form; reinforced concrete suggests the design of a continuous surface shell that can be form-found, geometric or free-form; and, finally, wood can be used for strained and un-strained grid-shells that can be form-found or free-form. Since wood is a relatively lightweight material, dead load due to self-weight won’t be the predominant load case [4, p. 2].

The above categorisations are summarised in Figures 4.4 and 4.5.

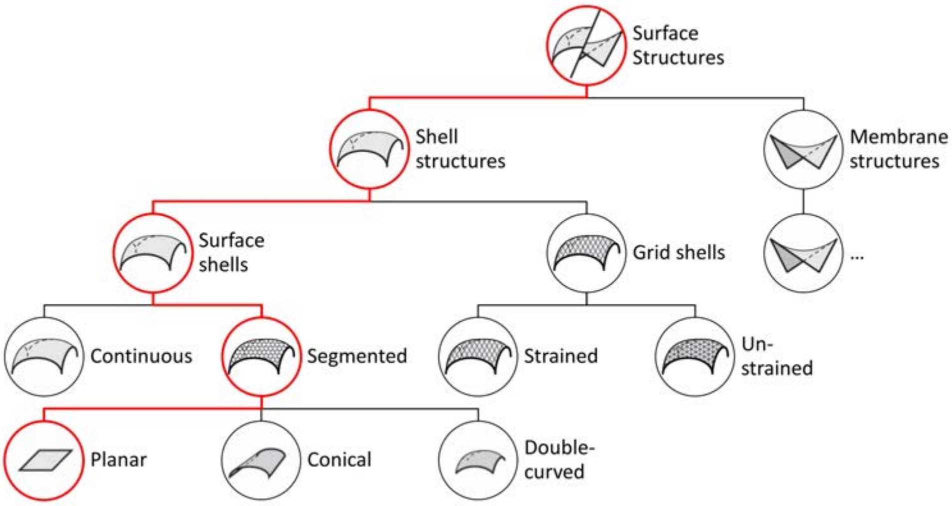


Figure 4.4: Taxonomy of shell structures based on stress-state and surface continuity.

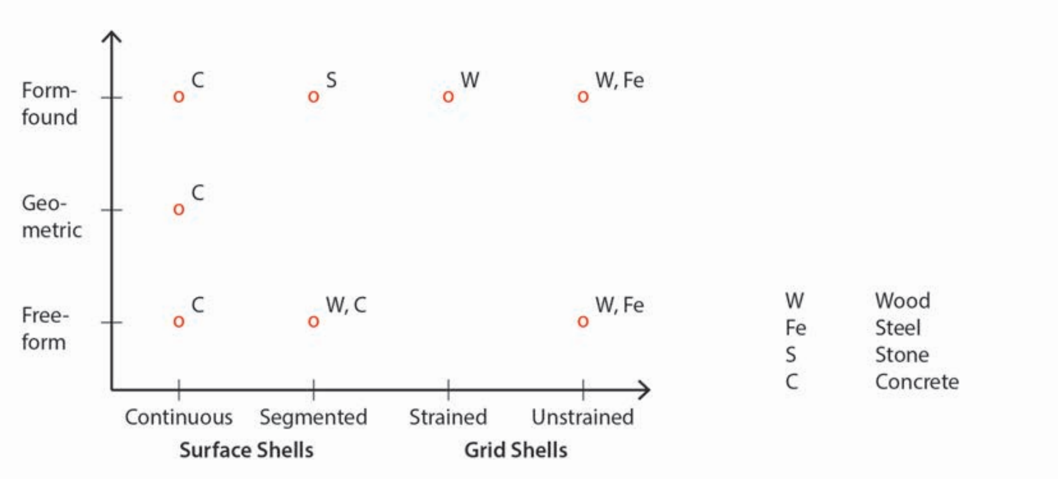


Figure 4.5: Material classification of shell structures based on stress-state and surface continuity.

4.3 State-of-the-art

The reciprocal relationship between form and structure is particularly strong in the context of thin shell structures, where the form depends on the flow of forces, and vice versa, therefore requiring methods of form-finding to determine the equilibrium shape of the shell [4, p. 1]. Consequently, shell design often is a two-step process where, in a first step, the equilibrium shape is found, and then material properties are added to the numerical model for analysis in the second step (ibid., p. 2).

The review of the state-of-the-art in shell design in the context of this research thus primarily concentrates on different numerical *form-finding methods* and only tangentially refers to *structural analysis methods*, the application of which is outside the scope of this research.

The third relevant topic is *optimisation*, since shell structures are usually designed to meet multiple design criteria in addition to self-weight. Therefore, optimisation methods are necessary in order to meet design objectives, such as minimisation of material and weight, minimisation of cost, or maximisation of stiffness, while being subject to design constraints [4, p. 3], which usually leads to multi-objective optimisation problems.

Finally, in view of the construction challenges historically associated with continuous shell structures, *segmentation* enabled by digital fabrication and associated joint design constitutes the fourth topic in the review of the relevant state-of-the-art.

4.3.1 Form finding

According to Adriaennsens et al. [4, p. 2], form-finding is a “forward process in which parameters are explicitly/directly controlled to find an ‘optimal’ geometry of a structure which is in static equilibrium with a design loading”.

Veenendaal and Block [196, pp. 115-16] divide form-finding methods into three main categories: *Stiffness Matrix* methods that are adapted from structural analysis, such as finite element analysis; *Static Equilibrium* methods with only a geometric stiffness that are material independent; and *Dynamic Equilibrium* methods solving for a steady-state solution that are material dependent.

The most relevant methods from an architectural design point of view are the ones from the second and third category, because they are computational methods that can be appropriated for shell design in an otherwise traditionally engineering-dominated design domain [89, p. 138]. These include Force-Density-Method (FDM) and Thrust-Network-Analysis (TNA), and, respectively, Dynamic Relaxation (DR) and the Particle Spring method (PS).

The respective models are composed of branches and nodes (edges and vertices) as a way for storing the connectivity between elements. External and internal forces then act on the nodes and the shape is considered to be in static, or steady-state, equilibrium if the forces add up to zero [196, p. 118]. Different solvers for solving the equilibrium equations exist, such as Newton–Rhapson’s method for static form-finding methods; and either explicit (for example leapfrog integration, classic fourth order Runge-Kutta), or implicit (for example backward Euler) integration methods are used for dynamic form-finding [196, pp. 116-17]. The particular solver, however, is considered to be of less interest as long as it ensures convergence towards equilibrium [196, p. 129]. The general premise is that the different methods will be able to find the same solution if the problem and its boundary conditions are identical [196, p. 116].

The foundations of the aforementioned methods have been laid in the last quarter of the 20th century, but, with the exception of the force density methods, have only been transferred to shell design around the turn of the this century or later. TNA and PS are further developments of previous methods: FDM and DR, respectively. The methods are well known and described in literature, therefore only the characteristic features are summarized here focusing on their relevance for early stage architectural design. For a more in-depth comparison see Veenendaal and Block [196].

Similar to the correlation between material, shell geometry and shell type that was mentioned above, a correlation can be established between form-finding method, material and shell type, which raises the question which of these methods is particularly suited for form-finding of segmented timber shells.

4.3.1.1 Force-Density Method

As part of the static equilibrium methods, FDM does not require the definition of material properties to solve static equilibrium. Instead, FDM introduces force-to-length ratios, or force densities, for the branches in the network [196, p. 118]. One advantage is that the actual force densities do not need to be known at an early stage in the design process since the density is a ratio which can be defined by the user. It is therefore particularly suitable for early design exploration.

A characteristic feature of this method is that the solution is entirely independent of the initial coordinates of the free nodes and only the location of the fixed nodes, that is the supports, needs to be known [196, p. 119]. Furthermore, Block and Veenendaal [196] showed that a wide variety of arches that are scaled versions of each other can be obtained by varying the force density. They also show that the resolution of the network influences the final result.

While higher force densities generally attract higher forces, the effect of individually setting the force densities for each branch on the final shape is difficult to anticipate, since the branch lengths also change [196, p. 120]. Therefore, in order to control the final shape, user interaction can be included to vary these parameters and assess their influence, or a constrained optimisation method can be used where additional constraints are introduced to control the shape such as a target height, or maximal length of branches [196, p. 120].

As demonstrated by Linkwitz [117], the method, which was originally developed for cable nets, can be used for the design of tensioned membrane roofs and, by introducing dead load, for the design of shell structures such as unstrained timber gridshells. Because the forces $\vec{f}$ and actual lengths $\vec{l}$ are known after form finding, one can calculate initial lengths $\vec{l}_0$ for any given axial stiffness EA. In this way, results from FDM are typically 'materialised' for subsequent structural analysis of different load cases [196, p. 128].

4.3.1.2 Thrust-Network-Analysis

As state before, it is difficult to anticipate the correlation between the final shape in FDM and the input parameters $\vec{q}$ and $\vec{p}$, force-densities and loads respectively. Especially, since force-density is an artificial, physically not meaningful value. TNA thus tries to simplify the problem, by indirectly controlling force densities through the manipulation of the horizontal force components, or thrusts, in the branches [196, p. 120]. The method is based on the duality of force- and form-diagram, where the force diagram is the dual of the horizontal projection of the design solution and vice versa. A characteristic feature is that a single force polygon representing the same horizontal thrust is equally valid for different results with different horizontal projections; conversely, a single horizontal projection is valid for different results with different force polygons (resulting from different horizontal thrusts in relation to different vertical displacements of the nodes).

Assuming constant loading, in the first case, the funicular shell is the result of the manipulation of the form diagram while the force diagram is given; in the second case, the shell is the result of manipulating the force diagram through a scaling factor while the form diagram is given. In the design of the same shell, both approaches can be used alternatingly. In both cases, force densities are obtained from the given horizontal projection and thrust rather than prescribing them at the beginning. Veenendaal and Block [196, pp. 122-23] demonstrate iterative implementations of shape-dependent loading, which take the changing self-weight of the branches into account for shell variations based on a constant form diagram. They also demonstrate implementations of constant-length chains, which take the changing horizontal projection into account for shell variations based on given constant length and thus self-weight of the branches of a catenary arch.

Similar to FDM, solving for static equilibrium does not require material specification in TNA, but the possibility of designing the final shape through the flow of forces and the flow of forces through the shape makes TNA particularly suited for early stage design exploration of compression-only shapes of static equilibrium. TNA is particularly suitable for the design of masonry shells where dead load due to self weight is the predominant load case, as demonstrated through multiple projects of the Block Research Group at ETH Zurich, for example the Texas pavilion project by Block, Lachauer and Rippmann [21]. In these applications, TNA ensures that the line of thrust follows the neutral axis of the shell, that is the middle-third of the masonry section [21, p. 74]. Similar to FDM, the results of TNA can be 'materialised' by updating the model with material specific parameters for subsequent structural analysis.

4.3.1.3 Dynamic Relaxation

Dynamic Relaxation is part of the category of dynamic equilibrium methods aiming for steady-state equilibrium, which, within given tolerances, is equivalent to static equilibrium. This method is similarly based on the definition of nodes and branches, which in this case are called links. Force densities are called tension coefficients and, in contrast to the previous definition of force densities, now contain axial stiffness with Young's modulus and initial links lengths to calculate strain [196, p. 123]. As a rule of thumb, increasing the axial stiffness EA, generally results in shallower arches [196, p. 124].

Each node in the network now accelerates as a function of the residual force and its mass with the initial velocity at the first iteration being zero. The resulting movement will oscillate around a steady-state solution unless some form of damping is introduced, such as viscous damping [196, p. 124].

When comparing characteristics of DR and FDM as stated in Veenendaal and Block [196], a few major differences become apparent:

1. In DR, form-finding now assumes a physically meaningful initial geometry determining the starting lengths of the members, whereas in FDM the solution is independent from the starting position of the free nodes.
2. In DR the general solution is independent of the resolution of the mesh, except for its impact on the accuracy of the solution, whereas in FDM the resolution affects the result as mentioned above.
3. In DR, loads are shape-independent and remain the same at each iteration step.
4. Finally, while the computational cost per iteration in DR is relatively low compared to FDM, DR generally requires more iterations to solve a problem.

Through the addition of spline elements and bending stiffness between branches, DR can be used for the design of strained grid-shells as demonstrated by Adriaennsens et al. [3, p. 93], and, in case real material values are available, DR can also be used for static analysis directly [196, p. 127].

4.3.1.4 Particle Spring method

The Particle Spring (PS) method is also part of the category of dynamic equilibrium methods and is in many respects similar to DR [196, p. 125]. The method implements Hooke's law (see Sec. 4.1.2) and originally came from the field of computer graphics and animation, in which iterations are a necessity to show any type of animated progression.

Nodes are called particles and links are springs, but it differs from DR in how the branch forces and masses are defined. Forces in the springs depend on a spring constant, or spring stiffness, and a rest length L_0. When comparing characteristics of PS and DR as stated in Veenendaal and Block [196, pp. 123-6], the following similarities and differences become apparent:

1. Similar to DR, PS relies on damping and on the definition of a convergence tolerance.
2. PS requires more advanced integration methods than DR and more iterations to converge on a solution.
3. In PS it is possible to define the rest length $L_0 = 0$, whereas in DR this would lead to a division by zero.
4. Similar to DR, Spline elements and therefore bending stiffness can also be implemented in PS.

With the ability to set the rest length $L_0 = 0$ and no dead load applied, PS can be used to model pure tension structures that adopt minimal surface shapes. At the same time, it can be used to model compression structures such as demonstrated by Bhooshan, Veenendaal and Block [18].

Summarising, design cases in which initial geometry and subsequent deformation have meaning and material properties are known, dynamic equilibrium methods are straightforward to use and appropriate [196, p. 129]. Static equilibrium methods, on the other hand, seem less suitable as design methods for segmented timber shells, in particular those made of timber: FDM is suitable for the design of unstrained gridshells; and TNA is particularly suitable for funicular structures where self-weight is the predominant load-case, which does not apply for lightweight timber shells. What makes dynamic equilibrium methods particularly useful for segmented shell design is their ability to incorporate additional design drivers that can be represented as forces and the active role that the designer can take in form finding (see 4.3.4). It is worth noting that in the context of shell design, nodes and links (particles and springs, respectively) can represent actual building elements as well as design surfaces that form the basis for a subsequent subdivision or segmentation, as in the case of TNA and masonry shells.

4.3.2 Structural Analysis

The state of the art in structural design is characterised by the combination of methods such as Computer-Aided Geometric Design (CAGD) for design parameterisation, Finite-Element (FE) analysis for response analysis, and Nonlinear Programming (NP) for optimisation [20, p. 50].

In terms of analysis, Finite-Element analysis is the de facto standard requiring the discretisation of the design geometry into triangular or quadrilateral finite elements as a precondition for establishing the data structure and stiffness matrix. This is an important task, that can have an effect on the plausibility of the results. At the same time, it is an intermediate step that will have to be repeated every time the design geometry changes. Therefore, numerical analysis methods are being further developed for example towards iso-geometric analysis where the analysis part also uses the B-Spline and NURBS parameterisation. The main advantage is that it provides a consistent concept for both parts of the design [20, p. 53].

4.3.3 Optimisation

Adriaenssens [4, p. 3] defines structural optimisation as the "inverse process in which parameters are implicitly/indirectly optimised to find the geometry of a structure such that an objective function or fitness criterion is minimised." Structural optimisation thus is needed as soon as additional objectives or constraints are introduced to form-finding other than dead load [4, p. 3].

Optimisation methods can generally be divided into two main categories: Local and global methods. Local methods search for a local optimum and offer no guarantee that the 'most optimal' solution has been found. Global methods include meta-heuristic methods that are based on principles observed in nature, such as evolution or annealing, and are essentially trial-and-error strategies [4, pp. 3-4]. In both cases, optimisation is an iterative process that approximates the optimal solution by minimising the 'cost' and thus requires evaluation at each iteration step. To determine the cost of the current solution can be a costly operation by itself in case computationally expensive FE-analysis is necessary or it can be a simple calculation to determine the difference between a current quantity and a target quantity.

Optimisation requires the definition of the optimisation problem, which, in turn, defines the solution space within which the 'optimal' solution is expected. This usually is a challenge in early architectural design stages, when significant parameters might still be unknown and the solution space is subject to change.

Optimisation can be based on a single objective function, which can be a function of various weighted quantities, in which case their sum is minimised. In case of competing objectives, which require multiple objective functions, the Pareto-front can be explored to assess the trade-offs between the objectives [4, p. 3]. Generally speaking, each quantity or parameter in the optimisation can be thought of as a dimension in an n-dimensional solution space, which is called the design space.

One example of a structural optimisation approach used in preliminary design is the shape design method proposed by Sasaki [164, p. 261]. There, shape generation is formulated as an optimisation problem and sensitivity analysis is used to optimise a given shape by minimising strain energy in the shell. The optimisation method searches for a "local optimum that is as close as possible to the initially conceived shape. [...] For each node in the design surface, the gradient of the strain energy with respect to change in the vertical coordinate z is computed. The value of z is revised in the direction that will reduce the strain energy. The new strain energy is computed in FE analysis, and the procedure continues until there is no appreciable change in the strain energy after an iteration". Used as a modification mechanism, the method is particularly suitable in preliminary design requiring a feedback loop between design and optimisation to achieve a high "design density" [164, p. 261].

It is important to note that in most real-world cases there is no clear optimum in the mathematical sense, since at least some objectives are typically conflicting – notoriously cost, speed and quality – and the same quantities might be weighted differently by different stakeholders. Rather, the concept of optimality can be introduced from the study of biological systems denoting that "the vast range of morphological diversity between organisms inhabiting the same environment suggests that a standard and optimal solution does not exist, but rather different strategies may perform optimally under certain circumstances" [109, p. 27].

4.3.4 Steering of Form

Kilian [89, p. 131] proposes the concept of 'steering of form' as an extension of form-finding "towards desired design outcomes that take into account a combination of the different design goals". The concept is distinct from form-finding, as in varying a given set of parameters in a hierarchical model, and from structural optimisation, which does not allow for the integration of other possibly non-structural constraints.

Instead, the proposed approach introduces 'steering principles' that aim at variation of parameters at all levels of the definition of the design in a non-hierarchical system of associations including (1) the support conditions; (2) the loads on, or masses of the nodes; (3) the length and strength of the springs; and (4) the topology of the network and the related discretisation of load [89, p. 133]. The goal is to use the power of computational methods to achieve more integrative design solutions, and to extend the canon of forms that was established through conventional analytical techniques (ibid. p. 138).

'Steering of form' thus implies a more active role of the designer in the form-finding process, which allows continuous and interactive manipulation of the various parameters, in order to achieve additional goals besides the structural optimum [89, p. 132]. Steering of form thus offers the chance for design discoveries that would neither be apparent in the optimised solution nor in the original design intent (ibid., p. 138).

4.4 Segmentation

In building practice, the form-finding, analysis and optimisation methods for shell structures discussed above are predominantly applied for the design of grid-shells made of steel and glass, such as the roof of the Great Court of the British Museum in London and the Dutch National Maritime Museum, or made of timber, such as the Savill building. In the academic context, TNA is used for the design of discrete un-reinforced masonry shells such as the work by the Block Research Group at the ETH in Zurich.

As stated before, continuous thin shells made from concrete, while having been a popular typology in architecture and engineering from the 1930s to the 1980s, have lost their appeal in recent decades – a notable exception being the Teshima Art Museum (2010) by Ryue Nishizawa with Mutsuro Sasaki. Sasaki [164, p. 259] also recognises that this trend was due to the comparatively high effort necessary for in-situ construction, which involves the manufacturing of complex formwork and extensive on-site manual labor. These factors still apply today, even when considering the high degree of automation in other fields (see also [173, p. 225]).

A possible approach to address the challenges that are raised by the construction of continuous shells is segmentation. Segmentation allows for pre-fabrication, and, subsequently, quality control, higher accuracy, lower tolerances, a higher degree of automation compared with on-site fabrication, and overall less time spent laboring on site. On the other hand, the double-curvature of thin shell also leads to geometric variation in the segments, which in turn requires effective means of engineering and fabrication.

Advancements in computational design and digital fabrication over the last 25 years now enable architects and engineers to handle the increased geometric complexity of segmented shells: numerical methods allow for digital design, simulation and analysis of increasingly complex shell structures; numerical methods also enable the digital fabrication of geometrically unique building elements within the constraints of the building industry (batch size one production).

The premise and relevance of segmentation thus is that shells can be composed of prefabricated individual segments that are joined together on site to become an integral structure. Based on this premise, the Institute for Computational Design and Construction (ICD) and the Institute of Building Structures and Structural Design (ITKE) at the University of Stuttgart have undertaken various studies into the segmentation of timber and FRP shells between 2010 and 2019. The background, approaches and findings of selected studies are described and discussed in detail as part of the case studies in Part II.

4.4.1 Reciprocity between segmentation and structure

What the form-finding methods for surface shells have in common is that the shell is considered a continuous surface during form-finding, regardless if the resultant shell is continuous or discrete. Even in the case of TNA, where the explicit goal is to design masonry shells that consist of discrete elements, the subdivision of the shell is a design step that occurs after the shape of the shell has been determined. It can be argued, though, that the segmentation affects the load transfer in the shell surface and thus should have an effect on the geometry of the shell.

In the case of unreinforced masonry shells, the segmentation strategy only has to ensure that the voussoirs do not slide against each other by providing enough friction or interlocking between the voussoirs [21, pp. 73], for example, by orienting the courses perpendicularly to the force flow (ibid., p. 83). In the majority of shells, however, which have to provide a minimum of bending stiffness and might even be able to partially accommodate tension forces, the segmentation is more relevant to the overall load-bearing capacity of the shell.

In other ongoing research, a particular focus is therefore placed on the arrangement of the segments in the shell and the joints between segments. This includes research into the simulation and arrangement of the joints between segments and into the optimal joint layout for segmented shells [9; 55; 26; 114], as well as the investigation of biological role models for segmented shells as part of the Transregional Collaborative Research Centre TRR141 at the University of Stuttgart. An overview on the state-of-the-art regarding the development of principles of shell design based on biological role models is given by Grun et al. [73]. The question of arrangement and connection is also the focus of research into design and fabrication of folded plate structures, including the development of integrated timber joints, as well as their assembly, which are investigated in the work of Robeller et al. [160].

4.4.2 Polyhedral Plate and Triangle Dualism

What is then needed for introducing segmentation to continuous surface shells is a segmentation strategy, which generates shell action and transfers external loads into membrane forces in the segmented shell [115, p. 123].

In the mid-1980s, Ture Wester (1941–2008) identified a duality between polyhedral plate structures and lattice structures, which defines criteria of geometry and topology for achieving stable plate shells that are based on the known properties of triangular lattices [206]. Duality in this case refers to the general notion of dual meshes where “the vertices of the first mesh are in 1-1 correspondence with the faces of the second one, and vice versa.” [149, p. 262].

4.4.2.1 Comparison

A comparison of lattice and plate structures illustrates the corresponding dual properties. Lattice structures are composed of nodes and bars, such as trusses, space frames and grid shells. Pure lattices then are fully triangulated systems, which only rely on bars and pin-joints for their stability. Plate structures, on the other hand, are composed of planar thin plates that are connected along their edges by shear-resistant hinges. Pure plate structures are systems that are fully composed of 3-way, trivalent vertices, which only depend on plates and hinges along the edges for their stability.

Regarding the criteria for stability, lattice structures require a minimum of three bars to fix an additional node in space, whereas plate structures require a minimum of three lines of support to stabilize an additional plate plane in space [207, p. 615]. This minimum requirement corresponds in both cases, to a tetrahedral configuration of either bars or plates which is known to be stable. Thus, the 3-way vertex has exactly the same meaning to the plate effect as has the well-known triangle to the lattice effect [206, p. 5]. In surface structures, triangular lattices where six bars meet in one node will therefore be just as stable as hexagonal plate structures where always only three lines of support meet in one vertex forming Y-shaped connections. Regarding forces, in pure lattice structures only axial forces (tension and compression) will occur, which are concentrated in lines and points; and in pure plate structures only shear forces will occur, which are distributed along the lines of support and in the plane of the plates [207, p. 614].

The challenge in realising lattice structures thus lies in the manufacture of the nodal point, which is subject to concentrated forces from many directions [206, p. 6]. In plate structures, on the other hand, this problem of forces concentration is not present, which allows for plane building elements that may be of moderate strength, e.g. plywood, concrete panels, etc. (ibid.).

4.4.2.2 Duality

The duality between lattice and plate systems that was observed by Wester thus includes both form and structure: From a *geometry* point of view, triangular lattices and polyhedral plate structures are duals of each other (every node in a lattice corresponds to a plate, and every bar in a lattice corresponds to a line of support in a plate structure) as for example in case of the cube (plate structure) and the octahedron (lattice).

From the equilibrium point of view, the force equilibrium of a point in space is described by the closed polygon of the force vectors acting on the point (for example six force vectors representing six bars that the node is connected to). This corresponds to the force equilibrium of a plate in space that can be described by the same force polygon, in this case representing moment vectors acting along the edges of a six sided plate around a point that is outside the plane of the plate effectively representing shear forces [207, p. 616] (see also La Magna et al. [108, p. 3]). The shear forces thus are kept in equilibrium by the plates [206, p. 4]. This means that if a triangular lattice, e.g. form-found or free-form, is in static equilibrium then its plate-dual will equally be in static equilibrium.

Finally, from the point of view of forces, the axial forces in the bars of the triangular lattice can be converted into shear forces in the edges of the plates and vice versa through a geometrical relation [207, p. 617]. The dualism of pure lattice and plate structures thus is that “nodes and plates may be interchanged if bars are changed to lines of support and axial forces to shear forces” [207, p. 617]. The dualism is therefore not limited to topology and geometrical features, but also includes the mechanical properties of the structure [108, p. 3] (see also [206, p. 7]).

An experiment by La Magna et al. [108, p. 3] comparing the forces in a simple plate structure (cube) and its dual lattice (octahedron) served as a proof of concept that the values for axial and shear forces of the two systems result equal, thus confirming the dual principle of forces. La Magna et al. [108, p. 3] thus propose the analysis of a plate structure based on its lattice-dual as a simple way to calculate the forces and reactions in a spatial plate structure. As of 2017, this can be solved much quicker in any truss calculation software than the shear in the plates using FE-analysis.

4.4.2.3 Pure plate structures vs. folded plate structures

Li and Knippers [114, p. 1] observed that one advantage of segmented plate shells over single-layer grid shells is that segmental plate shells can generate local bending stiffness without the help of a bending-stiff joint because each plate is constrained by the other two such that they cannot rotate around an edge. In contrast, single layer grid shells usually need bending-stiff joints to generate the required minimum bending stiffness to avoid buckling (see also [210, p. 22].

It is important to note that the stability criterion, which provides that a plate structure is as stable as (or even more than) its triangulated lattice dual, does not apply to foldable or origami-type plate structures, since their dual does not correspond to a fully triangulated lattice (see also [139, pp. 11-12]). Folded plate structures thus do not fall under the category of pure plate structures since they do not follow the 3-way vertex rule and thus rely on bending stiffness along the plate’s lines of support in order to ensure the plate system’s kinematic stability. Folded plate structures are essentially mechanisms that achieve stability through additional locking elements, for example through double-layering, or have a global shape, such as a barrel vault, with only one pre-dominant load-bearing direction [160].

4.5 Current Research Directions and Open Questions

The previous two sections gave an overview on the state-of-the-art in the fields of thin shell design and shell segmentation. This section describes current research avenues.

One of these avenues is the further development of *form-finding methods* (such as in the case of TNA and PS approaches) as pursued by Block et al. [21] and Bhooshan et al. [18]. A second avenue is the further development of *analysis methods* towards iso-geometric analysis as described by Bletzinger and Ramm [20]. The third research avenue is *segmentation* as pursued by Oval et al. [144] with a focus on discrete masonry shells; by Grun et al. [73] focussing on segmentation patterns based on biological principles; and by Robeller et al. [160] with a focus on foldable plate structures, the design of integrated timber joints and their assembly. A fourth recurring theme in current research is *materialisation* enabled by numerically controlled fabrication methods as demonstrated by the robotic fabrication approach of the BUGA Wood Pavilion in 2019 [7].

With segmentation of thin shells being a relatively novel research avenue that has only fairly recently been enabled by the advent of numerical methods for design, analysis and fabrication, only a few examples of segmented shells exist today mostly from the academic context. This includes previous work by the author as part of the previously mentioned ICD/ITKE Research Pavilions and building demonstrators that were realised between 2011 and 2016 at the University of Stuttgart.

In segmented shell design, form-finding, segmentation, analysis and design for fabrication are usually considered sequentially. An example is the design of unreinforced masonry shells, where segmentation is even only considered after analysis. The current state thus corresponds to a linear chain from form-finding and analysis to segmentation and fabrication.

However, particularly in the case of segmented shells that are not determined by their dead-load, the layout of the segments and the connection between the segments have an effect on the global structural performance. Therefore, segment design, which is driven by constraints of fabrication, should be considered earlier in the design process. The proposal at the core of this research consequently is to consider segmentation, fabrication, form-finding and analysis concurrently in the design process of segmented shells.

While the integration of design and materialisation was a focus in the previous work at the University of Stuttgart, the solutions and findings remained specific to the projects and, in the absence of generalisation, are thus of limited applicability and transferability to building practice. No research is known at this point that focuses on a systematic approach for integrating principles of segmentation and fabrication with principles of form-finding and analysis.

Such a holistic approach is by definition interdisciplinary and requires not only methods for integrating various disciplinary principles but also the ability to maintain an adaptive digital model throughout the various stages of design and fabrication. A particular challenge in this context is embedding such an integrative approach into the architectural design process starting with preliminary design.

Architects, as the originators of the architectural design process, lack a systematic approach, methods, and tools that would allow the integration of design principles for segmented shells into architectural design. Furthermore, architectural designs that are uninformed by principles of shell design can hardly be reengineered in the later stages of the design process without requiring substantial re-design. Consequently, if architects don't have adequate methods at their disposal, light-weight design, and the design of thin shells in particular, will remain a niche endeavour.

The aim of this research therefore is to establish a systematic approach for identifying, defining, and integrating design principles for segmented shells into the architectural design process. Given the advantages of pure plate structures, such as local bending stiffness, the investigation will focus on segmented shells where the plate-lattice dualism applies.

PART II

Case Studies

“The most powerful abstractions are those that minimise the semantic distance between the units of analysis that are intuitively used to conceptualise the problem and the constructs present in the solution paradigm.”

—Nicholas R. Jennings [82, p. 286]

5

Introduction to the Case Studies

5.1 Research Demonstrators

Starting in 2010, the Institute for Computational Design and Construction (ICD) and the Institute for Building Structures and Structural Design (ITKE) at the University of Stuttgart have realized an annual research pavilion at the intersection of academic research and teaching. In parallel, various other building prototypes and demonstrators have been developed in the context of funded research. The construction of these full-scale pavilions, building prototypes, and prototypical buildings (collectively called research demonstrators), allowed the validation of the various employed computational methods and fabrication processes and ultimately provided the opportunity for synthesis of different research developments in one demonstrator.

Most of these projects investigated the use of timber or FRPs for lightweight construction in architecture. While segmentation has also been applied to FRP shells, most notably the ICD/ITKE Research Pavilion 2013-14, the Elytra Pavilion at the Victoria & Albert Museum in London (see Prado et al. [151] and [150] respectively), and the BUGA Fibre Pavilion at the Bundesgartenschau 2019 in Heilbronn, Germany, this investigation focuses on timber plate shells only.

Taking into account the period until 2019, research demonstrators that are relevant for this investigation are the ICD/ITKE Research Pavilion 2011 (Chapter 6), the Landesgartenschau Exhibition Hall (2014) (Chapter 7), the ICD/ITKE Research Pavilion 2015-16, the Rosenstein Timber Shell (2017), and the BUGA Wood Pavilion (2019). These research demonstrators are part of the state-of-the-art in the field of segmented shells in terms of research and application of computational design methods, robotic fabrication processes, and development of material systems. They have been frequently referenced in literature.

5.2 Case Studies

As stated at the end of Chapter 3, ABMS generally tends to suffer from a lack of validation, transparency, and reproducibility [121, p. 151] despite its specific potential for architectural design and engineering. One approach to address the issue of validation of conceptual models is to derive conceptual models for agents and behaviors from the analysis of real-world examples through case studies as those models have already been validated. The complementary approach to increase transparency and reproducibility is to implement models in an open-source modeling framework as part of an accessible software platform, which is pursued in the development part of the dissertation.

At the core of the research thus lie the case studies. Following an inductive research approach from the analysis of specific examples to generalization, the findings of the case studies form the basis for an abstraction of design principles and, in the further development, for implementation aiming for a systematic approach for the design of agents and behaviors in multi-agent systems for segmented timber shells.

In the context of this research, the purpose of a case study is the analysis of the research projects, the real-world cases, with regards to defined criteria. The case studies consequently follow the format of case study reports consisting of an introduction outlining the motivation and aim of the case study, a background chapter outlining the context of the case, and the main body of the case study describing analysis approach, identified functional principles and their associations. A summary of the findings and their discussion constitutes the final section of each case study.[1] The case studies form the methodological template for identifying functional principles from real-world segmented shells and thus are the pre-condition for translating principles into the agent-based design context in the further development.

Two cases, ICD/ITKE Research Pavilion 2011 and the Landesgartenschau Exhibition Hall, have been selected for analysis according to the following criteria: (1) their relevance in the field of segmented timber shells; (2) their typicality that warrants a case study; and (3) the contribution of the author to the original research projects as a research associate at the ICD. Particularly, the ICD/ITKE Research Pavilion 2011 is the first manifestation of research into segmented timber shells and is based on previous research into robotic fabrication of timber joints. Based on

---

[1] The case studies follow the recommendations by UNSW Sydney for writing case study reports in engineering (https://student.unsw.edu.au/writing-case-study-report-engineering, accessed October 21, 2018).

the findings of this first demonstrator, the Landesgartenschau Exhibition Hall is the outcome of a research and development project with the objective to further develop segmented timber shells towards an application in building practice and to evaluate the relevance of segmented timber shells in the context of construction. It is also the first robotically pre-fabricated Beech plywood shell.

Furthermore, the close familiarity of the author with the cases, their functional mechanisms and design history, allows for a detailed and accurate report and also ensures that no important principles are overlooked. Each project was conducted in an interdisciplinary team of researchers consisting of architects, engineers and, in the case of the research pavilions and Rosenstein Timber Shell, with biologists and a dedicated team of graduate students in architecture. The contributions of the author generally included research at the intersection of

1. *robotic fabrication* focussing on the development of methods and approaches for the integration of fabrication requirements into the design stage as well as the implementation of the robotic fabrication process;
2. *computational design and modeling* including particle-spring based and agent-based modeling; and
3. *segmented shell design* focussing on segmentation strategies.

The specific contributions are outlined in the introduction sections of the case studies. Beyond the personal continuity between the projects, the sequence of projects from 2011 to 2017 allowed for a research continuity where the outlook of one project led to the research questions of the following project. In this way, research questions could be asked, hypotheses formulated, and experiments designed that could not have been otherwise [177, p. 116].

5.3 Functional Principles and Design Principles

The aim of the case studies is the identification of functional principles of morphology, organization and process that were used in the design, fabrication and construction of the projects under investigation.

For clarification, a working definition of what constitutes a principle is introduced. A principle, according to its Latin root *principium*, is that which stands at the beginning or origin of something else, for example of an effect or a behavior. Interpretations of the term depend to a large extent on the specific context such as jurisdiction, natural science, philosophy or colloquial use where synonyms might be tenet, rule, law, percept, guideline, etc. An example for the definition of the term 'principle' is as "a progressive abstraction generalized from a series of data and particular cases" [6, p. 1]—a definition that is particularly relevant in the context of this research.

One way to differentiate the term would be to distinguish between axiomatic, methodic, and systemic principles. *Axiomatic principles* declare a law, which can form the basis of other laws: for example, a natural law, such as gravity (as one of the four fundamental forces) or a fundamental law, such as the connection between cause and effect, which forms the basis of the scientific method. Gobrecht [65, pp. 90] provides different descriptions of the quality of this connection in Philosophy, of which the following are applicable in this context: “a relation (Locke, Kant), a necessary linkage (Kant), a law or rule (Kant, Russel), a relation of content (Descartes), a temporal relation (Kant), or an energy-preserving, physical relation (Quine).”

An example of a *methodic principle* would be the principle of induction, which “describes the transition from the specific (e.g. of past events) to the general (e.g. future and, consequently, all events) and produces an extension of content through empirical generalization” [65, p. 192]. This needs to be seen in contrast to mathematical “complete” induction, which can be arrived at through a mathematical proof. In the context of this investigation, induction is understood as empirical generalization, i.e. applying a principle that was derived from past experience to future instances. The principle can therefore only be used to predict similar future events, which may occur with a certain probability [65, p. 195].

The third class, *systematic principles*, can be used to relate cause and effect. The principles in this class are based on the axiomatic principle of a rule-based, causal relation between cause and effect and, in the absence of a mathematical proof, on the methodic principle of induction (empirical generalization). A systematic principle can therefore be defined when an effect can be associated with a constellation of specific input factors, that always produce the observed effect. Such a link can be established even when the underlying rule or mechanism is unclear (in which case it is arrived at through “incomplete” induction). An example is the principle of adaptation in evolution, which has been declared long before the underlying genetic mechanisms have been understood. A second example is the use of the steam engine, before the laws of thermodynamics have been formulated. Therefore, principles often are named after the effect they produce, not after the actual mechanism. In this investigation, however, principles are named such that they are as intelligible as possible. This means that, while most are named after the effect they produce, some are named after the mechanism.

In the investigative chapters 6 and 7, the term principle is used in its systematic sense as functional principle, retrospectively identifying the set of functioning rules and mechanisms that relate a certain outcome to its input factors. Such a mechanism can also be considered a function, which computes an output value based on a set of variables, e.g. x, y and input parameters, e.g. a, b, c etc., and which could be derived through empirical generalization.

The question that is asked when defining systematic principles is “What causes are necessary to produce the observed effect?” Furthermore, the question “*How* are the effects produced?” is asked to identify the mechanisms that are at the core of the principles. For example, in the case of the observed shear- and compression-load bearing capacity (effect) of finger joints (cause), loads are transferred via interlocking teeth in edge-wise connected plates and in-plane forces (mechanism). As a systematic principle in the context of timber plate structures, this relation can be called the ‘Principle of Shear-load Transfer’ named after the most prominent effect produced (Fig. 5.1).

The identified functional principles then form the basis for the further development (Chapter 9). In the development chapter, the term principle is used in its axiomatic sense as design principle, meaning that it doesn’t need to be proved again when it is used. A design principle is forward-looking, that is, in order to achieve a desired outcome, certain guidelines should be followed with respect to necessary input factors and mechanisms. The question at the core of defining design principles is therefore “Which input factors and mechanisms are necessary in order to produce the desired outcome?” Again using the example of finger joints, if the goal is to transfer shear and compression forces between planar elements (input), then finger-joints can be used (output) by connecting the elements in a form- and force-fitting manner (mechanism). Connecting planar elements through interlocking teeth in a form- and force-fitting joint with the objective to transfer shear and compression forces can consequently be called the ‘Finger-Joint Principle’.

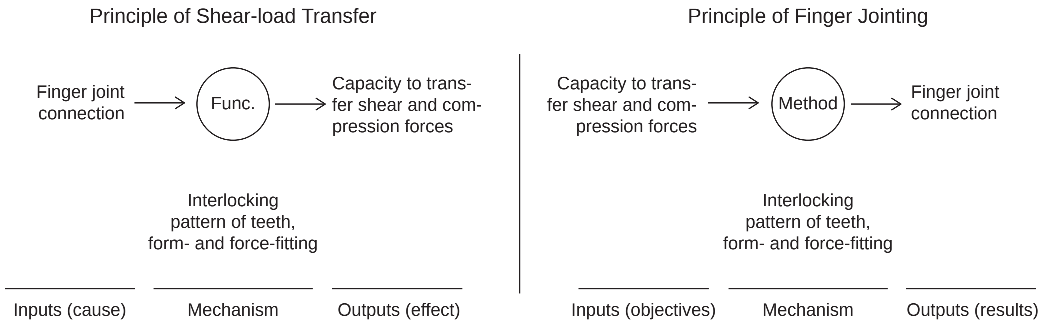


Figure 5.1: Comparison of systematic vs. axiomatic principles using the example of finger jointing.

Using this definition, systematic functional principles will be identified through the case studies and subsequently be abstracted and transferred as axiomatic design principles which then allows the activation of these principles within the agent-based framework. In both cases, combining several principles into a ‘inference chain’ allows the definition of high-level principles. In the field of Logic, such a chain is typically called a 'Sorites', meaning that multiple propositions are chained together in such a way that the conclusion of one proposition becomes the subject of the next premise thereby relating the final conclusion to the initial premises [190, p. 390].

The systematic approach for the abstraction and transfer of design principles into the behavior-based modeling framework will ultimately be independent from the projects in this investigation and can be applied in other contexts. Whenever new design principles are to be defined as part of the conceptual modeling stage of an agent system for segmented shells, a case study, either of an existing "real world" case or of a hypothetical scenario, will form the basis for the identification of functional principles which, subsequently, can be transferred into the framework. In the second case of the hypothetical scenario, the validation of the principles will logically occur 'a posteriori' through implementation in a research demonstrator.

5.4 Aim and Scope of the Case Studies

The aim of the case studies is, first, to identify functional principles that were applied during the stages of design, development, fabrication, construction and evaluation; and, second, to develop recommendations as to which of these principles have the potential to be transferred into an agent-based modeling framework (see Section 9.3). With fibre-placement being an additive fabrication process employed as part of the research into FRP shells, the selected cases employed subtractive fabrication processes, mainly CNC milling. While on-site fabrication has been investigated and demonstrated in the context of the research into FRP shells, one of the main affordances of segmentation is off-site pre-fabrication, which applies to both FRP and timber shells. In terms of robot control, the cases followed the offline programming paradigm where tool paths are generated 'offline' and then executed by the robot without additional sensor integration. With respect to robotic setup, they used multi-axes setups consisting of 6-axis industrial robot arms with flange-mounted effectors and one additional external axis as vertical positioners, which allowed reorienting the work piece. With subtractive fabrication and pre-fabrication, offline robot control and multi-axis fabrication, the selected cases thus cover a significant part of the spectrum of principal fabrication and construction strategies.

While the two cases follow distinct approaches and result in unique outcomes, they share further similarities beyond their fabrication contexts, which allows for a consistent comparison of the results: they also share a common research agenda, including shared hypothesis and research aims, and a common research context, which includes previous work. Both, shared context and objectives, and shared previous work, are introduced in the next few sections. In the following two chapters, the specific goals and objectives, approaches, results and findings of the two cases are investigated and discussed with a focus on identifying and categorizing the main functional principles. This categorization then provides the basis for the further research development towards a behavioral approach for the design, fabrication and construction of segmented shells (see Chapter 9).

5.5 Shared Background and Context

The two projects under investigation are situated in a shared research context defined by, first and foremost, the resurgence of timber construction in architecture. Contemporary timber construction entails 21st century post-industrial production logic and CNC-milling as the pre-dominant method for machining timber, from which integrated timber joints re-emerge as a viable connection technique. Furthermore, biomimetic design was proposed in both cases as a strategy for filtering the expanding solution space of timber-based material systems in architecture.

5.5.1 Timber Construction and Wood

The starting point of the research conducted in the two projects is the observation that timber as a construction material has recently experienced significant innovation in the areas of fabrication technology, building code and regulations, and structural applications: for example, innovative new building products allow explorations in high-rise construction (see [58]). Given that wood was the predominant construction material in the pre-industrial era but has lost relevance during industrialization with the advent of steel and concrete, this resurgence of interest in the construction industry is recognized as a “timber renaissance” [111].

The renewed interest in wood is typically attributed to aspects of sustainability, but also to its characteristic performance. From the point of view of sustainability, the relevance of building with wood lies in its regional availability in areas such as Central Europe, Scandinavia and North America – allowing for short transportation routes and a decentralized regional economy. Krieg, Schwinn and Menges [102] highlight the relevance of regional added value chains made possible through timber construction. The second aspect is its renewability: in Germany alone 122 million m^3 of wood grow annually, of which only two thirds are harvested, effectively creating an annual surplus of wood [49]. A third aspect is the use of wood as long-term CO_2 storage: the CO_2 that is captured during the growth of a tree is effectively stored in the material for the lifetime of a building. A comparison with conventional energy-intensive materials shows that timber structures reduce the amount of CO_2 emissions during production and operation. In certain applications even a negative carbon footprint of timber structures is possible, effectively storing more CO_2 than what is released during the production of the building materials [93, p. 19].

A second reason for the renewed interest in timber can be identified based on its performance characteristics: from a mechanical point of view, wood has a higher structural performance in relation to its weight than steel [111]. It can absorb high amounts of tension in its fibre direction and still have a significantly better "weight-cost" ratio of carrying a compressive load relative to its density than concrete or steel [66, p. 321-22]; at the same time, wood is highly machinable, in the past obviously through manual tooling, and therefore especially suitable for CNC-fabrication. These performance characteristics in addition to its sustainable credentials make wood a relevant candidate material for lightweight construction.

Additionally, principles of lightweight construction, in particular increasing the load-bearing to self-weight ratio, become relevant in the context of timber construction. Timber is a renewable, albeit limited, resource and there are competing interests in the use of forests as they fulfill a number of conflicting socio-economic and ecologic roles: natural habitats and biodiversity, sources of water and fresh air, and invaluable spaces for recreation on the one hand, and economic considerations of investment and return, on the other, can place a considerable pressure on forests. Lightweight design is consequently proposed as a strategy to arbitrate the conflicting goals of generally increasing timber construction while minimizing the volume of wood exploitation on the other [172, p. 178].

The presented cases are therefore part of an overarching and ongoing effort to design and develop resource-efficient lightweight structures and to demonstrate their performance through prototype buildings. This entails the research and development of corresponding design and fabrication procedures.

5.5.2 Post-industrial Production

The second aspect at the basis of the research conducted in the two projects relates to the field of production economics with the observation that the geometric differentiation of building systems has traditionally been limited due to manufacturing constraints imposed by the serial production paradigm of the 20th century. Primarily because of economic factors, systems with a large number of similar construction elements have been given preference [97, p. 259].

The advent of CNC-technology and industrial robot arms, particularly in combination with automated programming approaches, now offer the opportunity to overcome the previous limitations and allow efficient one-off (batch size 1) production. This has led to the overall research question of how the expanded design solution space that is provided by computational design and robotic fabrication can be meaningfully explored and activated in areas of particular performance (see [14; 98; 103; 171; 173; 174]).

5.5.3 CNC-Milling

The third topic is CNC-milling as an established approach to machining timber in the context of contemporary timber structures. Next to sawing and planing, milling is one of the most common subtractive methods for wood working. Originally operated manually, these methods have been computerized over the last few decades. In these instances, formerly manual processes have been digitized by replacing manual with numerical control with a focus on increasing the efficiency, repeatability and quality of the results. Schindler describes this development as the transition of material- and information processing from human to machine, which defines the 'turning point' between machine-tool-technology and information-tool-technology periods [166, p. 86].

CNC-Milling automates subtractive fabrication strategies and, correspondingly, different standardized routines are typically included in CAM software packages such as roughing, finishing, pocket milling, drilling and contour cutting. Depending on the kinematics of the machine, routines with 3-, 4- and 5 degrees of freedom can be executed. Given that contemporary timber structures are increasingly non-standard, consisting of many unique building elements, CNC-Milling is also used for producing free-form building elements. However, due to its inherent material waste and the time consumed by the roughing and finishing fabrication steps, approximating free-form surfaces through milling is usually considered inefficient.

Consequently, alternative milling strategies have been explored that limit material waste, such as 5-axis flank milling, where a finished surface can be produced while omitting the rough cut resulting in ruled-surface geometry [28, p. 325] (see also [168]). CNC-Milling is used in both discussed projects for 5-axis contour cutting, effectively flank milling the edge surfaces of the building elements.

When cutting elements from planar sheets of material such as plywood, the overall efficiency of material usage is determined to the largest extent by the nesting efficiency of the elements on the stock sheet. While custom job cycles have been developed in both cases for milling of the joints, for nesting the commercially available solution RhinoNest™, a software plugin to Rhino has been used (see 5.5.7).

5.5.4 Integral Timber Joints

Contrary to conventional timber construction, in integral timber joints the joining mechanism is integrated into the joint in the form of specific geometric features, which transfer loads between building parts [161, p. 75]. This often leads to complex joint geometry and therefore requires elaborate fabrication methods. The numerical control of the milling process and particularly robotic milling re-introduces the possibility of producing integrated timber joints and offers the opportunity for enabling the efficient production of these complex joints.

As mentioned before, wood has been one of predominant building materials throughout the pre-industrial era, and consequently wood jointing techniques were developed and refined over a long period in the face of limited material supply and the exigencies of manual labor. The joints were tailored to the local geometric requirements of each specific connection resulting in highly adaptive, albeit labor-intensive, mono-material joints [167, p. 187].

Finger joints are an example of integral timber joints. Finger jointing is an ancient timber connection method that is more than 3500 years old [90, p. 11] and is still a commonly used corner joint for connecting planar or linear elements. Finger joints are characterized by their ability to connect planar elements in a force- and form-fitting manner through multiple interlocking teeth with a straight or tapered shape, resulting in high shear-load bearing structural capacity. They allow for connections without any additional fastener and can reduce warping effects during dimensional changes (i.e. swelling or shrinking of wood) [97, p. 263] (Fig. 5.2).

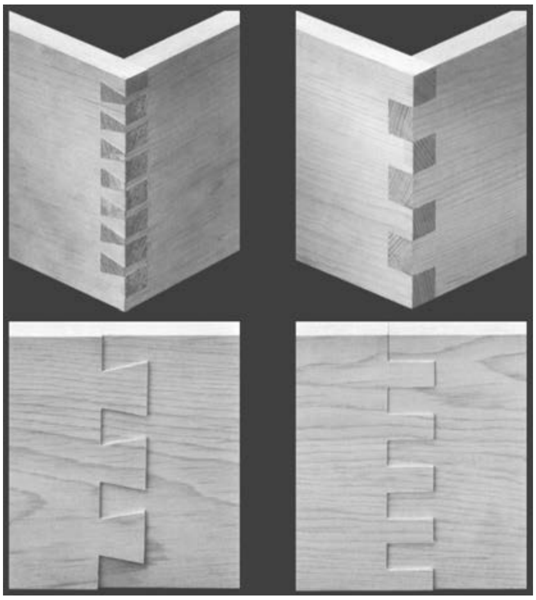


Figure 5.2: Form- and force-fitting timber connection. From [67, pp. 156-58] by Grunder (1986).

In traditional wood working, the structural and aesthetic quality is achieved mostly through manual fabrication. During industrialization, the focus was set on mass-production and the reduction of manual labor for economic reasons and on standardization of timber connections for reasons of calculability. This has led to the preference of metal fasteners over geometrically complex mono-material connections [100, p. 523]. Therefore, even if methods of numerical calculation have significantly improved and now allow the simulation and analysis of integrated joints and non-standard connections, manually fabricated joints still cannot be taken into account in contemporary timber construction [166, p. 163]. Today, finger joints are mainly used in furniture design due to their aesthetic qualities and, in their wedge-shaped variation, for linearly extending timber slats within the industrial fabrication process of other timber products such as cross-laminated timber or glue-laminated beams [136, p. 272]. At the outset of the research related to the first case, CNC-milling for producing finger joints was still limited to rectangular or planar plate connections due to the finger joint's complex geometry. An economically feasible fabrication method for joints connecting plates of varying angles did not yet exist. This observation consequently led to the research question of how the expanding solution space of robotic fabrication can be utilized specifically for expanding the range of application of finger joints (see [109; 174]).

5.5.5 Machinic Morphospace

The increased potential of morphological differentiation of building elements based on the possibilities and constraints of the machine corresponds to an expanding solution space—the space within which lies what can theoretically be produced given the parameters of a specific configuration of tool and machine [100, p. 522] (see also Menges and Schwinn [130]).

In order to systematically denote what can theoretically be and has empirically been produced with respect to the specific parameters of a given machinic configuration, the concept of 'morphospaces' has been transferred from the field of theoretical morphology in biology [129, p. 33]: "There, the term morphospace is used to describe concepts that see the morphological features of a specimen as actuations within a solution space" [130, p. 121].

Factors such as connection angle and properties, tool and effector dimensions, workspace area, fastening possibilities, etc., define the spatial domain of plate, finger joint, and module geometries. In order to ensure that the construction elements' geometry satisfies the production constraints, these aspects directly influence the form-generation process as part of the computational design approach [174, p. 163]. This led to the ensuing research question of how particularly promising and performative areas of the solution space could be identified.

5.5.6 Biomimetics

In order to address the previous research question, biology was proposed as a 'filter' for constraining the search space to areas where performative solutions have been found in nature [97, p. 260]. Consequently, a biomimetic design approach was pursued in the two cases that are discussed here, as well as in the other research pavilions completed in the same period, for abstracting functional principles of structural morphology in biology and transferring them to technology. The goal being to strategically explore the solution space of finger-jointed plate structures in areas of specific performance [98, p. 576]. The corresponding hypothesis was that functional principles of structural morphology found in biology can serve as role models for the development of a "bio-informed material system [98, p. 574]." Relevant principles were identified in Echinoids, which include sea urchins and sand dollars, that govern features such as local and global plate arrangements, local load transfer, and edge formulation [97, p. 264]. The resultant material system would take the specific affordances of robotic fabrication into account in addition to established architectural requirements such as structure, spatial configuration, lighting, insulation, etc. [97, p. 261].

Starting in the fall of 2014, the approach towards research and abstraction of biological principles for segmented shells was formalized as part of the research project A07 "The sand dollar as a biological role model for segmented shells", in the Transregional Collaborative Research Centre 141 (TRR141).[2] This research project also produced a demonstrator—the Rosenstein Timber Shell, which largely builds on the technological findings discussed in the two case studies.

5.5.7 Software used

The geometric models described in the case studies were created, unless otherwise noted, using Rhinoceros®(Rhino) for 3-dimensional modeling and Grasshopper® (Grasshopper) for visual programming and as user interface.

Rhino is a 3d-modeling software based on the Non-Uniform-Rational-Basis-Spline (NURBS) model and is developed by Robert McNeel & Associates (RMN). The software is commonly used in the field of architecture for Computer Aided Geometric Design (CAGD). Through the openNURBS initiative, RMN provides the source code to read and write the .3dm file format under the MIT License[3] thereby ensuring access to the data and transferability between applications[4]. Its Application Programming Interface (API) allows the communication and interaction of Rhino with other software packages. The cross-platform Software Development Kit (SDK) RhinoCommon provides a framework for the development of custom algorithms and plug-ins for the Rhino modeling environment.

One such plug-in is Grasshopper by RMN, which allows the development of algorithms inside Rhino using visual programming. Visual programming, also known as "data-flow programming" (Davis 2013, p. 101) is the process whereby the logical flow of an application is modeled by visually inter-connecting predefined function blocks (components) on the screen resulting in a Directed Acyclic Graph.

---

[2] http://www.trr141.de, accessed September 11, 2017.

[3] https://opensource.org/licenses/MIT, accessed May 8, 2017.

[4] https://www.rhino3d.com/en/opennurbs, accessed May 8, 2017.

In such a graph, "a node represents an operation and a directed edge represents a connection (a flow of data between two operations)" (ibid.).

In the context of Rhino, Grasshopper allows the creation of hierarchical parametric models that capture the designer's intent in addition to the representation in the 3-dimensional model space. Consequently, the definition of input parameters allows for variation within the defined parameter domains. The functionality of Grasshopper can be extended through the development of third-party add-ons, for example, for advanced manipulation of surfaces and meshes, triangulation and spatial segmentation algorithms, physics simulation, structural and environmental analysis, machine learning, optimization, GA, and data wrangling. At the time of writing, Grasshopper was in beta development stage and available free of charge.

5.6 Shared Research Goals

In addition to the overall goal of designing and building lightweight and resource-effective timber structures, the goals of the research on segmented timber shells can be summarized as relating to design integration, timber fabrication, and design exploration. Regarding *design integration*, the goal was to establish coherent design processes that can take specific constraints of fabrication into account during the early stages of the design with the aim of developing fabrication-informed designs that do not require post-rationalization.

Regarding *timber fabrication*, the goal was to address the need for geometrically differentiated building components and intricate joint geometry in lightweight design by applying robotic fabrication as an approach towards expanding the solution space of integrated joints.

Regarding *design exploration*, the goal was to find more performative design solutions for timber constructions in architecture than those produced using conventional methods by exploring the expanding solution space that is provided by the potential synergy between novel computational design and fabrication technologies.

Finally, in order to test the specific hypotheses that are stated in the research projects and as a vehicle for interdisciplinary integration, the goal was *to validate* and *evaluate* novel material systems through the design and construction of full-scale architectural prototypes. The resulting research demonstrators consequently form the subjects of this investigation.

6

Case Study I: ICD/ITKE Research Pavilion 2011

6.1 Introduction

Preliminary research on producing geometrically differentiated integral timber joints, which was conducted at the University of Stuttgart in 2010, established that robotic fabrication is a viable method for producing finger joints [97, p. 263]. Furthermore, this research showed that robotic fabrication provided an increased solution space compared to established CNC processes by allowing non-standard connections geometries (ibid.) (Fig. 6.1).

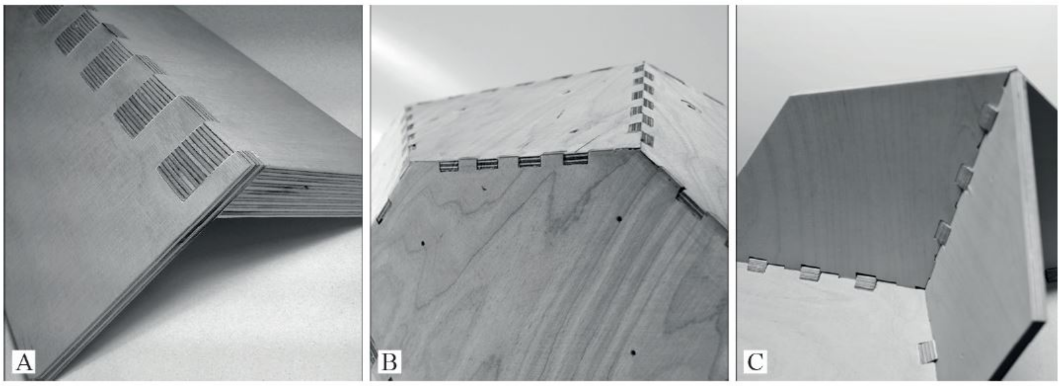


Figure 6.1: Finger-joint plate topologies. From [97] by Krieg (2011). A. Two plates with varying angles and thicknesses. B. Closed convex polyhedrons. C. Three plates joined along one edge.

The ensuing research project, which led to the ICD/ITKE Research Pavilion 2011 (“Case A”), was based on the findings of this preliminary work and provided further development towards design, technical development, and robotic fabrication of a segmented timber shell. The case represents the first segmented timber shell consisting of planar plates that was designed and built at the ICD. It demonstrates the architectural and structural possibilities associated with such an increased solution space and is therefore the subject of this first case study.

In the following sections, Case A is first contextualized with respect to its specific background, in addition to the general background given in Section 5.5. The main body of the case study then focuses on the identification of functional principles, which were used in the design, fabrication and construction of the pavilion. The identified functional principles are then summarized and discussed with respect to their role in the design and realization of the segmented shell thereby identifying the most relevant principles.

6.1.1 Main Characteristics and Relevance of Case A

Based on the previously established taxonomy (see 4.2), the ICD/ITKE Research Pavilion 2011 can be categorized as a freeform segmented timber shell, consisting of polyhedral segments that are composed of planar convex plates. Made from Birch plywood with a material thickness of 6.5 mm, a material thickness-to-span ratio of 1/580,[1] and a weight-per-floor area ratio of 16.9 kg m^{-2}, the shell can be considered extremely lightweight according to criteria of lightweight construction.[2]

Figure 6.2: Night-time view of ICD/ITKE Research Pavilion 2011 (Busch, 2011).

The pavilion was designed and built by graduate students and research associates as a temporary structure on the city campus of the University of Stuttgart. It was opened to the public in summer of 2011 and was exhibited for three months until the

---

[1] The structural span of 7.54 m was calculated from the centerline of the shell's footprint; the maximum clear span was 6.47 m.

[2] This characterization intentionally avoids the term "ultra-lightweight" as it is associated with adaptively manipulating stress fields during operation (see [184]).

end of November 2011.

The dome-shaped pavilion featured two interior spaces: a taller main space and a smaller interstitial space framed by the separating outer and inner layers of the shell. The interstitial space was accessible and allowed experiencing the constructional logic of the structure. The main space was characterized by the gradually changing openings of the interior layer, which not only allowed accessing the connection details between the prefabricated cells during assembly, but also visualized the changing effective depth of the double-layer structural system. Finally, they also served as indirect light sources at night (Fig. 6.2). The main space of the pavilion was oriented towards the park through a large circular opening and towards the university buildings through an arch-shaped opening. The latter served as main entrance with its open edge exhibiting the double-layer structure. The pavilion demonstrated the capacity of the building system to not only incorporate geometric differentiation but also to enable differentiated spatial and programmatic experiences [109, p. 38] .

Case A is an example for what can be produced using 7-axis robotic milling in order to access an area of the plate and joint morphospace that lies outside that of traditional CNC-machinery [100, p. 522]: 855 geometrically unique Birch plywood plates with a combined surface area of 275 m^2 and more than 100.000 individual finger joints along their edges were robotically pre-fabricated and pre-assembled off-site into 59 modules. The modules were assembled on-site using bolted, shear resisting connections. Approximately 1.8 m^3 of plywood were used to enclose 200 m^3 of gross volume while covering a gross floor area of 72 m^2. The material system's performance capacity for lightweight construction was demonstrated by the fact that, despite its considerable size, the pavilion could be built almost exclusively out of 6.5 mm thin sheets of plywood [171, p. 59].[3]

The following main contributions of Case A to the field of segmented shell design define its relevance to the field:

1. the further development of the robotic fabrication approach towards automated fabrication programming of three-dimensional finger joints for plates with varying angles and material thicknesses;
2. the integration of fabrication-specific constraints into the design phase through the systematic definition of the machine's morphospace (see 5.5.5);
3. the development of a structural model for finger-joint plate structures;
4. based on 1-3, the expansion of the solution space for segmented shells made of finger-jointed plates; and

---

[3] The exception being vertical 'fins' of 9 mm Birch plywood that were used in the bottom row of plate cells to connect the cells with the raised pavilion base as a means of anchoring the pavilion on site.

5. the establishment of a biomimetic design approach using Biology as a "selection filter" for identifying performative areas of the expanded solution space [100, p. 522].

6.1.2 Scope of the Case Study

This case study investigates design integration of architectural, structural, and fabricational requirements in the context of joint, segment, and global design of segmented timber shells. The focus of the investigation specifically is on identifying discrete steps in the design development, fabrication and construction processes, on the methods and approaches (mechanisms) used in those steps, on the relationships between steps, and on their role in the design process, e.g. as design drivers or constraints.

The case study does not focus on the biomimetic transfer process of functional biological principles. Where mechanisms are based on or relate to biological role models as part of the biomimetic design approach, reference is given to the corresponding literature. Finally, methods and approaches that were not immediately affecting the development process, as, for example, in structural simulation and analysis or in the evaluation of the built structure through 3D laser scanning, are outside the scope of this case study.

6.1.3 Research Team and Contributions of the Author

Conceived and realized at the intersection of research and teaching, the pavilion is the result of an interdisciplinary research and development project between architecture and engineering researchers of the University of Stuttgart, biologists of the University of Tübingen and graduate students of architecture at the University of Stuttgart.

Research partners included the Institute for Computational Design and Construction (Prof. Achim Menges, Steffen Reichert, Tobias Schwinn) and the Institute of Building Structures and Structural Design (Prof. Jan Knippers, Markus Gabler, Riccardo La Magna, Frederic Waimer) at the University of Stuttgart, and the Paleontology of Invertebrates Group (Prof. James Nebelsick) at the University of Tübingen. For the full credit list see Appendix A.1.

The author's contributions included the co-conceptualization and co-direction of the pavilion studio in the summer of 2011, in which most of the design and technical development were pursued, project management, integration of the robotic fabrication constraints into the design process, the further development of the robotic finger joint fabrication process including fabrication data modeling, as well as the evaluation of the as-built using 3D laser scan data.[4]

---

[4] The 3D laser scans have been performed by Stephan Laatsch as part of a Master thesis at the IIGS (Institute for Engineering Geodesy, University of Stuttgart) under the supervision of Prof. Volker Schwieger and Annette Scheider.

The research related to Case A has been reviewed and published in several conference papers and scientific journals. In particular, the case study is based on descriptions in Krieg et al. [100], La Magna et al. [109], Menges and Schwinn [130], Schwinn, Krieg and Menges [171] and Schwinn et al. [174]. Additional facts that were not included in any of the previous publications are highlighted using footnotes.

6.2 Specific Background and Objectives of Case A

6.2.1 Previous work: Performative Morphology

Initial research on the affordances of robotic fabrication for the design and realization of structures that consist of integral timber connections was conducted in the fall semester of 2010 within the context of a design studio titled "Performative Morphology."

The studio's hypothesis was based on the premise that the industrial robot's kinematic degrees of freedom can form the basis for the fabrication of a geometrically differentiated material system: similar to biological structures, which show a high degree of morphological variation and of functionality, artificial material systems can adjust to system-external and system-internal conditions through morphological adaptation while utilizing a minimal material and energy input [97, p. 260] (see also [78]). Beyond demonstrating the feasibility of implementing a traditional timber connection through robotic fabrication, the project showed that industrial robot arms can broaden the range of manufacturing possibilities for finger-joints by enabling the joining of plates of varying thickness at varying angles. This, in turn, allowed the re-interpretation and re-appropriation of a traditional finger-jointing process in view of contemporary robotic fabrication [98, p. 521]. The following paragraphs summarize the approach toward robotic finger-joint fabrication.

6.2.1.1 Fabrication programming

Following the state-of-technology in offline-programming of industrial robots, a geometric approach was chosen where, in a first step, the geometry of the joint was modeled using Rhino as a CAD software tool and subsequently exported to a dedicated CAM software tool, in this case OpenMind's hyperMILL™. In the following step, tool paths for finger-joint milling were modeled using standard CAM features, where each finger was modeled individually using a notching job cycle. Next, NC-Code according to ISO 6983 (G-Code) was exported.

In the fourth step, G-Code was imported and simulated in a dedicated post-processor, in this case QDesign's Robomove™, in order to visualize the movement of the robot arm. This step is typically necessary in order to check for out-of-reach positions, collisions and singularities. In this step, proprietary code was generated and exported by the post-processor and transferred to the robot controller.

Finally, QDesign software was used on the robot controller to interpret the proprietary code and to feed tool positions and orientations to the KUKA's robot control. The workflow from 3-dimensional geometry modeling to robotic fabrication is illustrated in Figure 6.3.

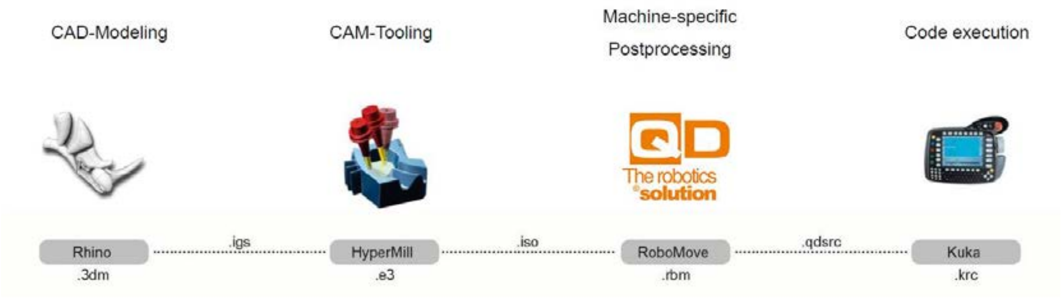


Figure 6.3: Robotic milling workflow: From CAD to CAM to fabrication (Schwinn, 2014).

6.2.1.2 Robotic fabrication

In order to produce a finger-jointed plate, a sheet of 6.5 mmBirch plywood was first mounted on a horizontal base and cut out with oversize to form a work piece. Second, the work piece was mounted to an inclined base frame such that the edge that was to be processed cantilevered and was accessible to the milling tool (Fig. 6.4). Third, this edge was then milled at right angles with the finger joint connection in order to avoid unnecessary gaps between the fingers. Fourth, after completion of the edge, the work piece was repositioned if necessary, and the next edge was milled. The fabrication process proceeded by repeating steps 2 to 5 for each plate until all edges of a plate had been milled. This process allowed the fabrication of closed, convex polyhedrons with varying angles between the plates (Fig. 6.1).

6.2.1.3 Finger joints

The proposed robotic fabrication approach was a proof-of-concept for the fabrication of integral timber joints that lie outside the kinematic range of established process-specific CNC-machinery (see 5.5.4).

During the development process, the angle limitations for the joints were identified as ranging from 15° to 165° as a function of the geometry of the spindle, the length of the tool, and the geometry of the finger joints. Joint angles of less than 15° and more than 165° result in unreasonably long "fingers" that compromise the integrity of the joint [174, p. 162] (Fig. 6.5).

In addition to demonstrating the ability to join plates at varying angles and plate

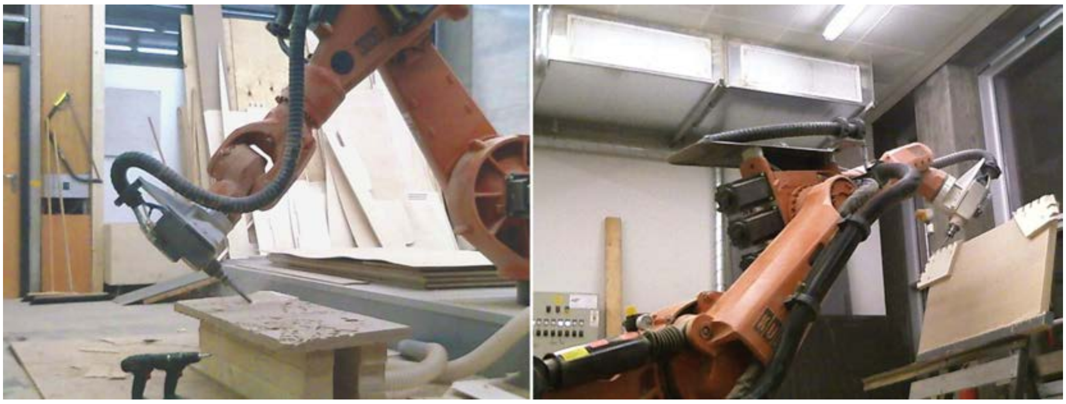


Figure 6.4: Fabrication strategy in previous work (Krieg, 2010). Left: The work piece is mounted on a horizontal base. Right: The work piece is mounted on an inclined base and cantilevering.

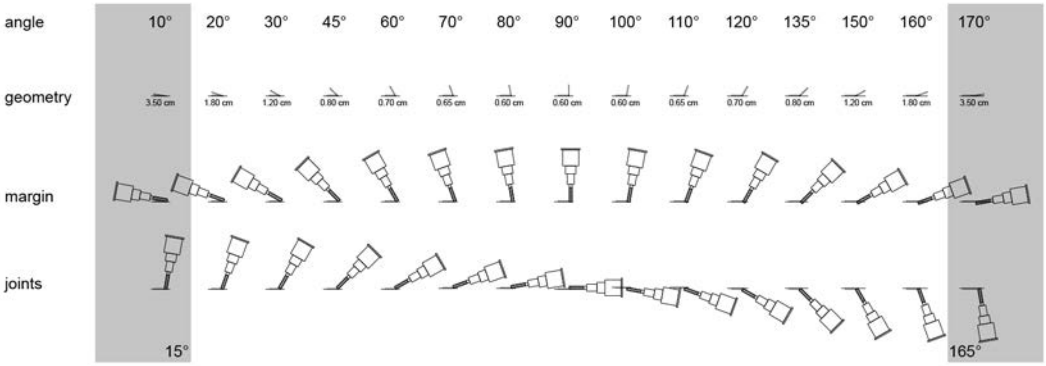


Figure 6.5: Fabrication limits for finger joints. From [174] by Krieg (2012).

thicknesses using finger joints and miters, the possibility for "connecting three or more plates along one margin" was shown [97, p. 264] (Fig. 6.1). The developed milling technique for individual finger joints thus led to a novel finger jointing system for connecting thin timber plates.

The findings of this preliminary research, published and discussed in the conference papers Krieg et al. [97] and [98], can be summarized as follows:

1. The integration of robotic production into the digital chain from design to fabrication provides the possibility of designing and fabricating geometrically differentiated timber joints.
2. The efficient fabrication of finger-joint connections and the further expansion of their solution space require a fabrication setup consisting of an industrial robot arm equipped with an additional turn table for handling the work piece.
3. As part of the proposed biomimetic design approach, it was shown that the definition of performance catalogues allows for the goal-oriented abstraction and transfer of functional principles for the design of plate systems.
4. It was shown that the conventional approach for fabrication programming, i.e., tool path modeling and robotic control code generation, was not suited in the

case of geometrically differentiated finger-joint production.

These findings regarding finger-joint and fabrication modeling, robotic setup and fabrication formed the basis for the development towards a segmented timber shell.

6.2.2 Expanding Solution Space

As discussed, preliminary research had shown that robotic fabrication could open up the design space for plate structures through the ability to efficiently join geometrically differentiated plywood plates by means of three-dimensional finger joints.

The increase in fabricational flexibility of industrial robot arms, compared to established process-specific CNC machines, is due to their higher degrees of kinematic freedom and due to the ability to customize fabrication processes with different effectors, additional axes and multiple arms.

Correspondingly, preliminary research had also shown that a seven-axis robotic setup was necessary in order to expedite robotic fabrication and avoid manual repositioning of the workpiece. In this setup, the seventh axis automatically repositioned the work piece so that it can be accessed from all sides by the milling tool.

Research on digital fabrication in architecture is thus characterized by a shift from CNC machinery designed for a specific task toward more generic fabrication equipment, such as industrial robot arms, with the aim to increase the design space and to develop more performative design solutions [174, p. 158].

Based on these findings, the premise at the outset of Case A was that robotic fabrication in timber construction allows for new design and fabrication strategies, as well as the reinterpretation and re-appropriation of existing techniques—both of which have the potential to enable novel architectural systems [100, p. 521].

The question that these initial findings then raised was, how the expanding solution space could be accessed in order to realize novel and performative load-bearing structures that maximize ecological benefits and performative capacities of building with timber. The hypothesis was that wood combined with integrative computational design and novel digital fabrication methods holds the potential for more adaptive and highly differentiated material systems and consequently offers the opportunity for novel performance-based architectural design. In order to investigate this hypothesis, the following research objectives were pursued:

i. to open up the solution space of the traditional finger jointing technique, while maintaining its inherent advantages [109, p. 38];
ii. to develop computational design methods that activate a part of the solution space for finger joints, which can exclusively be accessed by 7-axis robotic milling [100, p. 522].

6.2.3 Design-to-Fabrication Process

The research context is also defined by challenges that integrative design approaches are facing in architecture. One challenge is posed by the linearity and top-down nature of the conventional design process, with distinguishes between *upstream* and *down-stream* processes; a second challenge is posed by the separation of information and responsibility between design, engineering, fabrication, and construction. Together, they result in a lack of feedback between the major realization stages.

In a conventional CAD-CAM process, as of 2011 [171] and reconfirmed in 2015 [170], the design model is disconnected from fabrication modeling, which is typically performed downstream by a contractor in different, application-specific CAM software environments (see Section 6.2.1). This means that fabrication-relevant information can hardly be incorporated into architectural design, in turn requiring costly post-rationalization, which then affects all stakeholders in the development process.

The research conducted within Case A consequently investigated approaches to increase the integration between disparate development stages by pursuing the following objectives:

iii. to develop computational methods for integrating architectural and structural demands, and fabrication parameters [100, p. 512] ;
iv. to “design a design process” by fostering collaboration and information sharing between architects and engineers around a common information model that encapsulates project-specific programmatic, environmental, and context-related constraints and maintains the validity of those constraints during the design development [100, p. 522].

6.2.4 Standardization of Features

Preliminary research described above had shown that using established modeling approaches and data pipelines alone do not allow for the production of batch size 1 populations of building elements as they significantly constrain the accessible areas of the theoretical machinic morphospace. A further challenge is therefore the standardization of most commonly used features and routines in CAM packages, which limit the development of novel or alternative joint fabrication strategies.

At the outset of the project, the particular challenge was that no routine was available in the previously used, state-of-the-art CAM software package Openmind hyperMILL®, that would allow for the automated programming of thousands of individual finger joints. In the preliminary research described above (see 6.2.1) this meant that the fabrication data for finger-joints had to be modeled manually, one-by-one, based on a detailed, 3-dimensional digital model of the finger joints.

Using such an approach, design changes that occur ‘upstream’, possibly triggered by changed design requirements or by post-rationalization in other domains, as an effect of the separation of stages, would pose a major challenge for the viability of integral finger-joints and geometric variation in the context of an industrial application. In other words, the potential of robotic fabrication for lightweight construction could only be activated by developing new, generative approaches towards fabrication modeling. This confirmed the statement by Bechthold [15, p. 121] that non-standard fabrication, consisting of geometrically unique, one-off (batch size 1) building elements, requires automated fabrication modeling procedures.

With regard to the standardization of features, the research thus pursued the following research objective:

v. to parametrically model tool paths for finger-joint fabrication and to automate machine code generation directly from the design information model in order to allow for digital fabrication of geometrically differentiated elements [171, p. 56-58].

6.2.5 Exploration of Solution Space

The possibility of geometrically differentiating finger-joint plate structures that is enabled by automating fabrication modeling and the expanded solution space provided by robotic fabrication then raised a second research question: What is a suitable approach for exploring the related design space and a suitable heuristic for identifying areas of high performative capacity with respect to architectural and structural criteria? The corresponding hypothesis was that this could be achieved through a biomimetic design approach, which lead to the continuing investigation of biological principles for shell design that had started in the preliminary research in 2010.

With regard to the potential for the identification of performative material systems, the research thus pursued the following objectives:

vi. to activate the respective advantages and potentials in robotic fabrication (such as the extended solution space of possible form) and traditional wood jointing techniques as part of a new lightweight material system in architecture [171, p. 49];
vii. to use biology, as part of a biomimetic design approach, as a selection filter for the expanded solution space in the search for performative architectural morphologies [100, p. 522];
viii. to develop a modular, bio-informed material system that allows a high degree of adaptability and performance due to the geometric differentiation of its plate components [174, p. 160];

ix. to apply the developed material system in combination with a biologically inspired, performative plate arrangement into a pavilion while respecting specific architectural requirements and constraints [109, p. 29].

6.2.6 Topology and form finding

The computational design approach that was used in Case A is based on the PS method where each edge in a triangular mesh is converted into an elastic spring with equal rest length as a way to relax the mesh into equal edge lengths (see 4.3.1). In this process, the topology of the mesh plays a crucial role. The Schläfli notation, which was introduced by the Swiss mathematician Ludwig Schläfli, is a way to describe such topology. It takes the form $\{p, q\}$ where p represents the number of sides of a polygon and q represents the number of polygons meeting at a specific vertex.

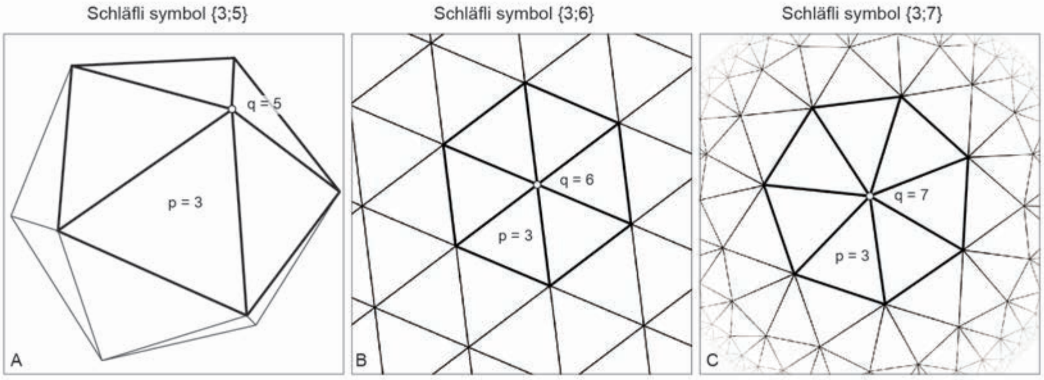


Figure 6.6: Shape and curvature change due to changing topology (Krieg, 2011). A: Spherical, positive curvature. B: Flat, zero curvature. C: Hyperbolic, negative curvature.

This topological indicator can be used to steer the system's form finding process (see 4.3.4): with the springs relaxing to approximate an equal rest length and being assigned the same internal spring strength, changing the Schläfli symbol triggers a three-dimensional change of the overall mesh shape towards positive, negative or zero Gaussian curvature [100, p. 527] (Figure 6.6).

More specifically, relaxation of regular triangular meshes where always six faces meet in one vertex (Schläfli-symbol $\{3, 6\}$), results in flat Euclidean surfaces, where no virtual 'stress' appears among the equalized mesh edges. In areas of a mesh where five triangles meet in one point (Schläfli-symbol $\{3, 5\}$), the mesh tries to compensate for the missing triangle by pushing the central vertex out of plane and contracting towards a locally spherical, synclastic surface, resulting from the tendency to form equilateral triangles. Similarly, in areas of a mesh where seven or more triangles meet in one point (Schläfli-symbol $\{3, 7\}$ or more), the mesh tries to accommodate the extra triangles by expanding towards a locally anticlastic surface, where the tendency to form equilateral triangles pushes the vertices at the perimeter of the triangle group out of plane.

Regarding form finding, the objectives thus were

x. to open up new possibilities for strategically manipulating mesh geometries in interactive form-finding processes using dynamic equilibrium methods, such as particle spring simulation [100, p. 527];
xi. to materialize the resulting shell surface, which can consist of positive and negative Gaussian curvature areas, using planar plates [109, p. 33].

6.2.7 Structural Design

As stated by Nachtigall [139, pp. 11-12] the structural advantage of rigid plate structures in comparison to foldable structures, such as origami, is based on their topological rule of joining no more than three plates in one point (Y-folds vs. X-folds). By following this principle, no bending forces should occur along the plate's perimeter, thus ensuring the system's stability (see also Wester [208]).

From the point of view of structural design, the objectives thus were

xii. to test the hypothesis of achieving bending-free structure by following the topological rule of always joining three plates in one point (see 4.4.2);
xiii. to develop a structural model of finger-joint connections and bending-bearing plate structures;
xiv. to find an effective way to build spatial structures by assembling panels which transfer only normal and shear forces between the edges.

6.3 First-level Functional Principles of Case A

The investigation occurs on three levels of abstraction, whereby on the first, lowest level functional principles are established by identifying discrete design and development steps. Each step is defined by its outcome, by the mechanism that produces the outcome, and by the corresponding minimum number of input factors that are required by the mechanism (see 5.3). Furthermore, the low-level principles are grouped according to the area of development in which they occur. Areas are the development of joints, segments (in this case called plate cells), form-definition, structural design, robotic fabrication, and construction.

While structural design principles are also included here in order to highlight their role in the design process, principles relating to the evaluation of the as-built using 3d-scanning, which constituted a considerable contribution of the author to the research project at the time, are not part of the investigation due to the missing link to the design process.

The definition of the individual low-level principles is described in the following sections with reference to the corresponding literature. These sections (labelled A.1 to A.6) thus constitute the fundamental body of work of the investigation into the functional principles of Case A.

A.1 Joint Development Principles

A.1.1 Principle of Finger Jointing

The Principle of Finger Jointing associates the fabrication sequence of a form-fitting finger joint with the geometric logic of an integrated plate connection and the inherent logic of subtractive fabrication via flank milling and spot facing using a customized milling tool (Fig. 6.7).

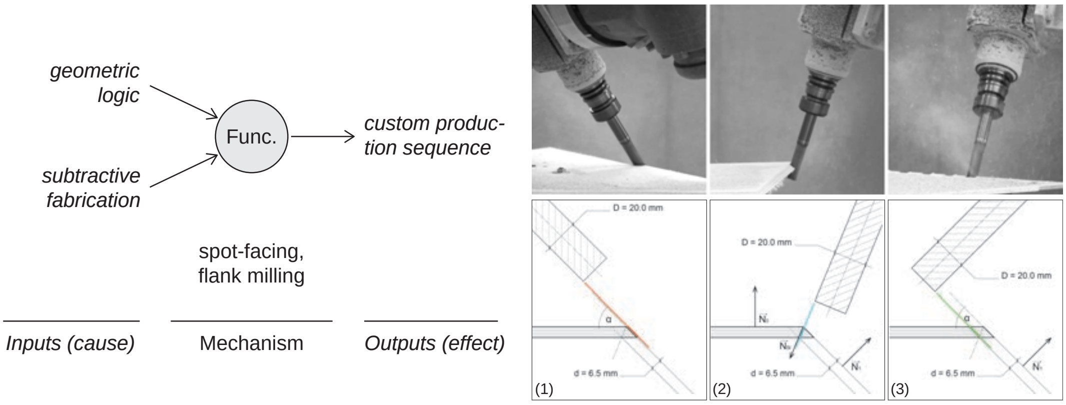


Figure 6.7: Finger Jointing principle. Right: Fabrication sequence. From [176] by Krieg (2012).

Inputs (cause) – Joint geometry and, consequently, cutting angles and the length of the teeth / depth of the indentations are a function of the angle between plates. CNC-cutting processes using milling tools are typically performed with the tool shaft, which results in rounded concave corners on the work piece.

Mechanism (function) – The fabrication sequence is based on the preliminary research described in Section 6.2.1 and uses a custom tool developed in cooperation with the industrial partner Leitz. It combines flank milling with the tool shaft and spot facing with its tip in one tool. The sequence consists of three steps for each plate based on its geometric relationship with neighboring plates: (1) contour-cutting of the plate's edges using the tool shaft to be co-planar with the adjacent plates; (2) flank milling a mitered edge at the start and end segment of each edge by aligning the tool axis with the plate angle's bisector; (3) indenting the finger joints into the plate's edge in the direction to the adjacent plate's normal vector through spot facing with the front end of the milling tool [174, p. 162].

Outputs (effect) – A sequence for producing accurately shaped force- and form-fitting finger joints as a function of plate angles.

Significance – As opposed to the widely used flank milling in CNC contour cutting, which typically results in the rounded corners for concave tool paths, the spot facing strategy produces a form-fitting finger joint connection [171, p. 58].

Furthermore, this principle is providing inputs to the principles of Finger Joint Production (A.1.2), Connecting Angles (A.1.3), and Finger Joint Fabrication Modeling (A.5.1).

A.1.2 Principle of Finger Joint Production

The Principle of Finger Joint Production associates the production of an integral joint type for edge-wise connecting timber plates with a custom fabrication sequence via numerically controlled machining (Fig. 6.8).

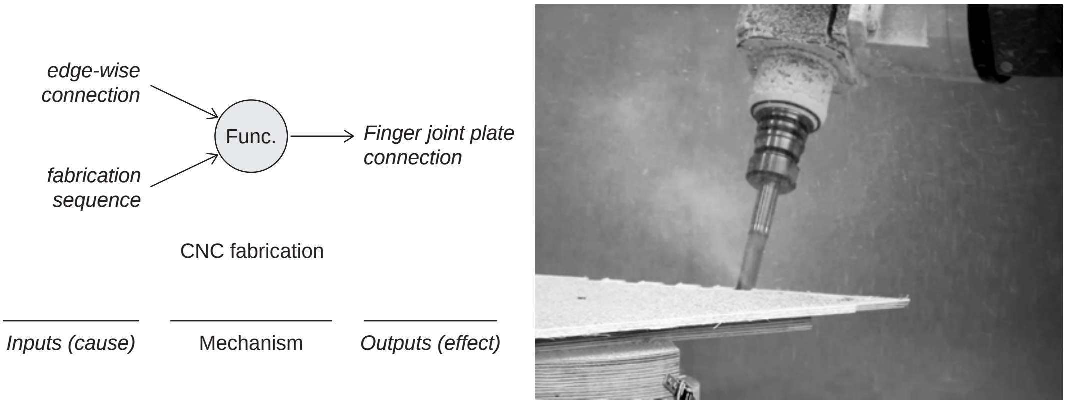


Figure 6.8: Principle of Finger Joint Production. Right: Close-up view of finger joint fabrication. From [100] by Krieg (2011).

Inputs (cause) – Prerequisites for the CNC-production of geometrically complex finger joints are, first of all, the objective to edge-wise connect planar plates without additional fasteners and, second, a viable fabrication strategy as described in A.1.1.

Methods (mechanism) – Using the milling strategy described in the principle of Finger Jointing (A.1.1), tool paths were initially modeled on the basis of a digital model that represented the target geometry of the work piece. This CAM step was facilitated by parameterized standard milling routines, such as contour cutting and notching, in a dedicated software environment. Machine code for numerical control of machine tools was then exported, post-processed, and executed as described in the preliminary research (see 6.2.1). The advantages of CNC-wood working compared to manual work are the precision and repeatability of the process, and, assuming an adequate programming strategy and robotics, the geometric flexibility that are prerequisites for finger-joint fabrication.

Outputs (effect) – An integral, form- and force-fitting joint for timber plates, in this case 6.5 mm Birch plywood, that are connected along their edges in a specific angle through an alternating pattern of interlocking fingers.

Significance – The principle is the precondition for the principles of Shear-load Transfer (A.1.4), Structural Validation (A.4.2), Scalability (A.5.2), and Assemble-ability (A.6.2). The capacity to produce such a joint in a numerically controlled way is the starting point for the ensuing research in the project.

A.1.3 Principle of Connecting Angles

The Principle of Connecting Angles associates a range of valid connection angles (15 to 165 °) with a given fabrication setup via load-bearing and producibility criteria that are defined by the geometric characteristics of the joint (Fig. 6.9).

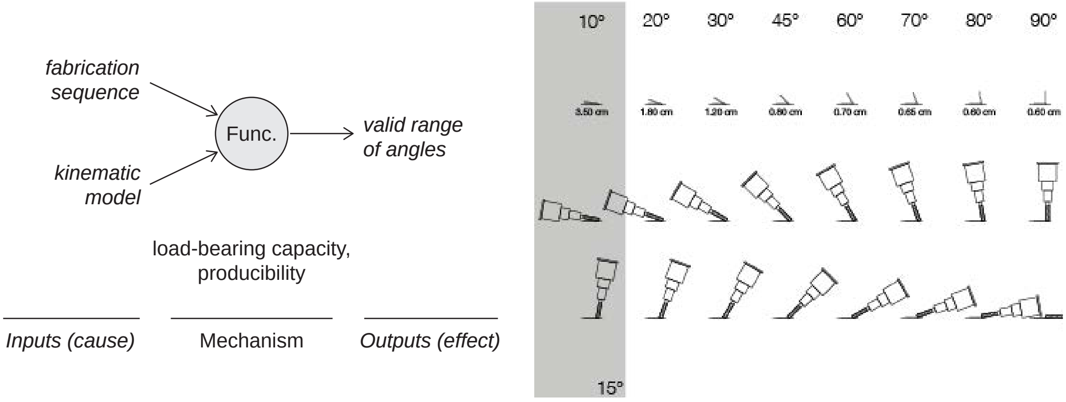


Figure 6.9: Principle of Connecting Angles. Right: Angle limitation due to machine constraints. From [174] by Krieg (2012).

Inputs (cause) – Fabrication sequence defined in Finger Jointing (A.1.1) and machine constraints determined by 7-axis Robotic Fabrication setup (A.5.5).

Methods (mechanism) – Geometric constraints are determined by the relationship between plate angle and depth of the teeth indentation: on the one hand, the depth of the teeth and, correspondingly, the contact surface between plates decreases for angles approaching 90 degrees, which reduces load transfer capacity; on the other hand, the length of the indentation of the finger joints increases towards 0 and 180 degrees resulting in extremely long and sharply angled finger joints that compromise accuracy of fabrication and structural stability.

Fabrication constraints are determined by the Degrees of Freedom (DOFs) of the fabrication machinery, in this case the 6-axis robot arm, and by the end-effector geometry. In order to ensure a collision-free fabrication process, the spindle casing, chuck and milling tool restrict feasible joint angles [174, p. 162].

Outputs (effect) – The structural capacity to transfer loads across the joint and the producibility of the joint are functions of the connection angles. This correlation allowed the definition of a valid range of connection angles of approximately between 15 to 165 degrees.

Significance – The principle is a precondition for the principles of Shear-Load Transfer (A.1.4) and Machinic Morphospace (A.5.8).

A.1.4 Principle of Shear-load Transfer

The Principle of Shear-load Transfer associates the structural capacity of a joint to bear normal and shear forces with plates that are connected along their edges within the valid range of angles via interlocking teeth in a force- and form-fitting manner (Fig. 6.10).

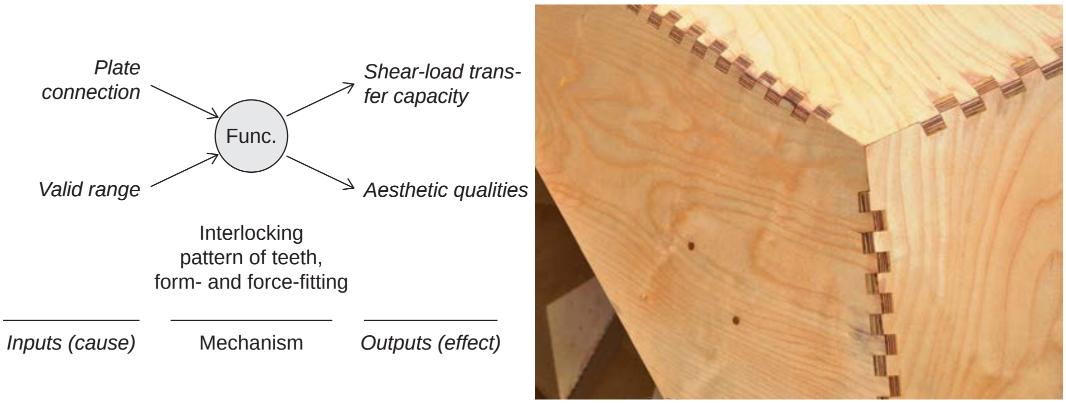


Figure 6.10: Principle of Shear-load Transfer. Right: Detail view of finger joints. From [100] by Menges (2011).

Inputs (cause) – Timber plates that are connected along their edges through integral joints such as the ones produced according to the principle of Finger Joints (A.1.2); producibility based on angular constraints according to the principle of Connecting Angles (A.1.3).

Methods (mechanism) – The performance of the integral joint is due to embedded geometric features, in particular the multiple interlocking straight-shaped teeth arranged in an alternating pattern, which connect two plates along their shared edge in a force- and form-fitting manner [174, p. 158]. In order to test the joint's load-bearing capacity, different setups were tested on their lateral, normal and shear force transmission and on their stiffness behavior [109, p. 33].

Outputs (effect) – From a structural point of view, the hinged connection is characterized by high structural capacity to withstand normal compressive and shear forces and by low bending stiffness of the rotational DOF [109, p. 33]. Furthermore, the aesthetic qualities of the form-and force-fitting connection are typically only achieved in furniture making.

Significance – No additional fasteners are required to achieve the structural capacity of the joint. The principle is a precondition for the principle of Bending-Free Edges (A.1.5).

A.1.5 Principle of Bending-free Edges

The Principle of Bending-free edges associates bending-free edges in a bending-bearing structure with a strictly trivalent plate arrangement ('pure' plate structure) and a shear-load bearing connection via geometric locking of the plates and resisting in-plane plate forces (Fig. 6.11).

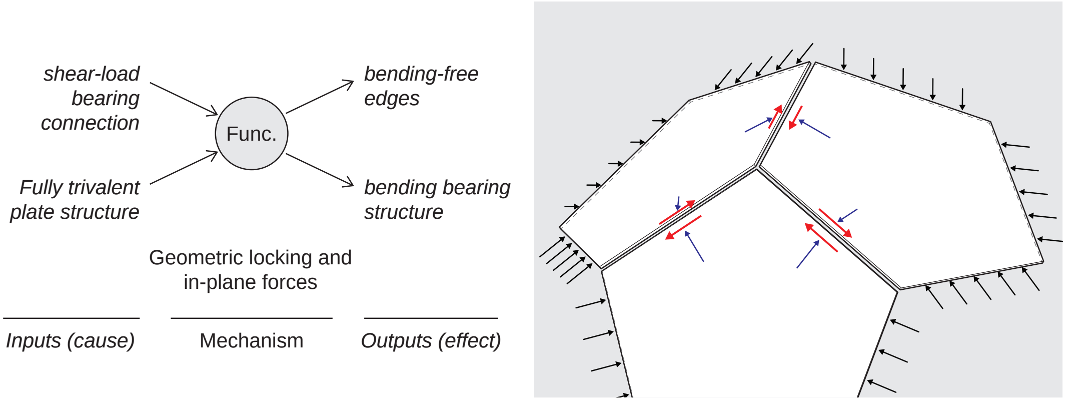


Figure 6.11: Principle of Bending-free Edges.

Inputs (cause) – 'Pure' plate structures, composed of just plates and hinges, where always three plates meet in one point (valency 3) [208; 108] as defined in the principle of Trivalence (A.4.1); shear-load bearing connection such as finger joints as described in the principle of Shear-load Transfer (A.1.4).

Mechanism (function) – Geometric locking: each plate is constrained by the other two. The plates are stabilized by resisting internal forces which lie in the plate itself. Each face of a 'pure' plate structure carries plate forces only, such that plates do not need to rely on carrying bending moments and torsion for their own stability [109, p. 31]. The mechanical behavior of trivalent plate structures was confirmed through FE-Analysis (ibid.).

Outputs (effect) – Bending-free edges in the plate structure are a function of the fully trivalent arrangement of plates and the shear-load bearing connection between the plates. This allows for plate structures that are bending-bearing on the global level, but are composed of hinged connections between the plates.

Significance – This functional principle allows for the definition of an axiomatic principle: any 'pure' plate structure will be inherently stable, whereas any variation from this arrangement pattern will result in a deformable and kinematic structure similar to origami [109, p. 31].

The principle is the precondition for cell development, in particular for the principle of Cellular Morphology (A.2.1) and Frustum (A.2.3).

A.2 Cell Development Principles

A.2.1 Principle of Cellular Morphology

The Principle of Cellular Morphology associates a 2-dimensional subdivision of the plane and a corresponding triangular mesh with user defined input points and logistic requirements via the computation of the Voronoi diagram (Fig. 6.12).

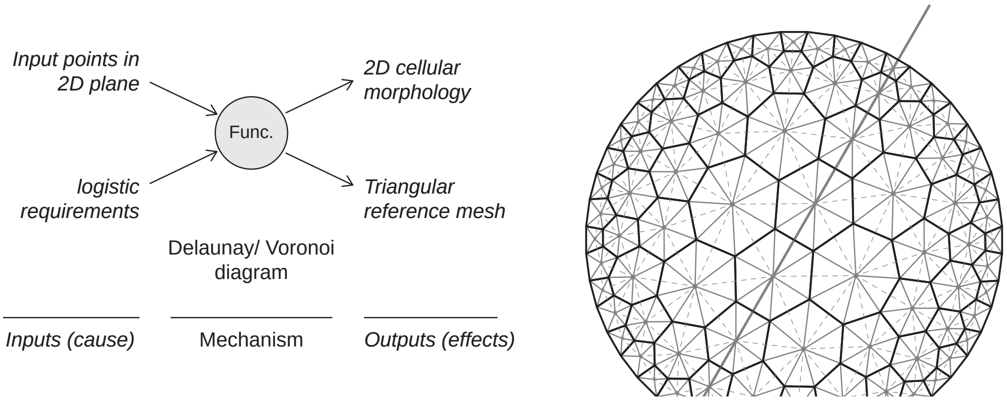


Figure 6.12: Principle of Cellular Morphology. Right image adapted from [100] by Krieg (2012).

Inputs (cause) – The inputs required for the computation of the cellular configuration are a set of non-coincident points on a 2-dimensional plane as well as logistic requirements dictating the number and sizes of plate cells. Implicit in the selection of the Voronoi algorithm is also the structural requirement of a trivalent topology given Bending-free Edges (A.1.5).

Mechanism (function) – The computation of the Voronoi diagram is a well-known algorithm to partition space based on proximity to chosen input points: every point within a resultant cell is closer to its input point than to any other input point in the set. The Voronoi diagram's dual is a triangulation of the input points according to Delaunay's method which provides a network representation of cell connectivity. In the second step, each cell in the 2-dimensional cell layout is triangulated by connecting each polygon vertex with the polygon's centroid.

Outputs (effect) – The result is a cellular layout of trivalent topology, where each cell corresponds to a 2-dimensional polygonal boundary of a plate cell. The adjacency between the cells is described by the connectivity diagram. The triangulation of the polygonal cells is stored as a mesh object, which functions as a reference mesh for the subsequent modeling steps.

Significance – This principle allows implementing the biomimetic principle derived from the plate morphology of Echinoids (see Zachos [217]). The principle is the precondition for the principles of Parametric Proxy (A.2.2) and Assembleability (A.6.2).

A.2.2 Principle of Parametric Proxy

The Principle of Parametric Proxy associates a trivalent 3-dimensional cellular model with a 2-dimensional cellular configuration of the same topology via a dynamic equilibrium method (Fig. 6.13).

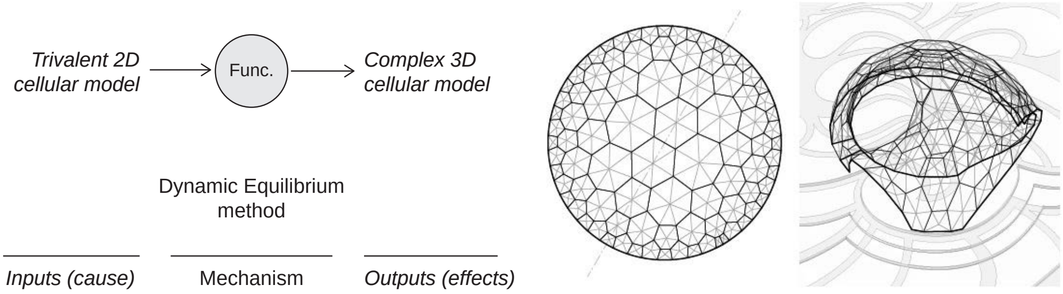


Figure 6.13: Principle of Parametric Proxy. Right images from [100] by Krieg (2012).

Inputs (cause) – A 2-dimensional configuration of polygonal cells including a triangulated reference mesh as described in the principle of Cellular Morphology (A.2.1); a method such as the one defined in the principle of Dynamic Equilibrium (A.3.3).

Mechanism (function) – The mechanism described in the Principle of Dynamic Equilibrium (A.3.3), in which triangle edges are converted to springs and assigned spring stiffness values, is used to transform the 2-dimensional triangular reference mesh into a 3-dimensional configuration of triangulated polygonal cells [109, p. 31].

Outputs (effect) – A complex 3-dimensional model consisting of triangulated polygonal cells of the same topology as the simple proxy model.

Significance – The simple model is used to control the more complex model, similar to the parametric design pattern “Controller" introduced by Woodbury [212, p. 191]). This principle is a precondition for the generation of a geometrically differentiated Modular Plate Cell (A.2.4) and Hierarchy (A.2.5).

A.2.3 Principle of Frustum

The principle of Frustum associates a trivalent plate configuration within the plate cell with the bending-free edges principle and producibility constraints via a custom mechanism for generating a truncated pyramid (frustum) based on a cell polygonal outline (Fig. 6.14).

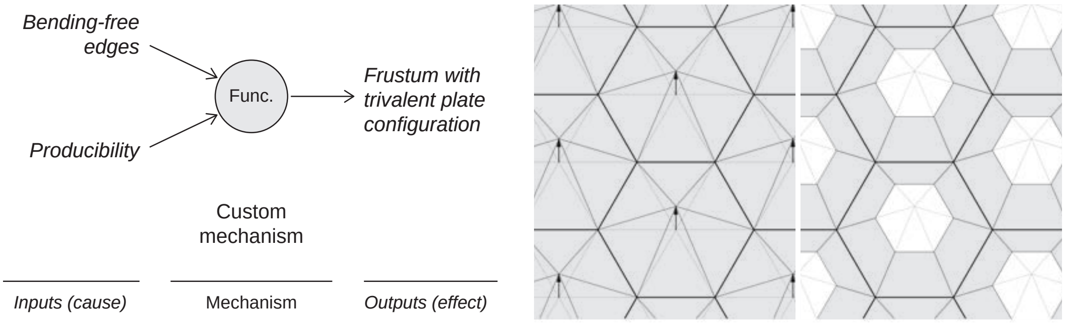


Figure 6.14: Principle of Frustum. Right images adapted from [100] by Krieg (2012).

Inputs (cause) – The requirement for a trivalent layout of convex plates as described in the principles of Bending-free Edges (A.1.5) and Stress Reduction (A.4.5). The range of valid plate angles defined by producibility constraints as described in the principle of Machinic Morphospace (A.5.8).

Mechanism (function) – In the first step of the custom mechanism, the central vertex in the cell is moved in its normal direction to form a pyramid with the cell outline as a base. In the second step, the pyramid is truncated at a variable height by a plane defined by the vertex normal direction [100, p. 525].

Outputs (effect) – The result is a frustum whose height and upper plane can vary in order to react to the different input factors [100, p. 525].

Achieving convex polygonal planar surfaces and its beneficial structural behavior is typically limited to geometries with positive Gaussian curvature. By raising the level of each individual cell, it is possible to achieve the desired design freedom in each curvature direction [109, p. 33].

Significance – Two aspects of this principle thus are particularly relevant: (1) the resultant frustum has a trivalent plate layout; and (2) no concave plates are generated even in anticlastic surface regions. This principle is the precondition for the principles of Modular Plate Cell (A.2.4), Hierarchy (A.2.5), and of Assemble-ability (A.6.2).

A.2.4 Principle of Modular Plate Cell

The Principle of Modular Plate Cell associates a geometrically differentiated polyhedral cell composed of planar polygonal plates with a non-planar closed polyline via the Frustum mechanism (Fig. 6.15).

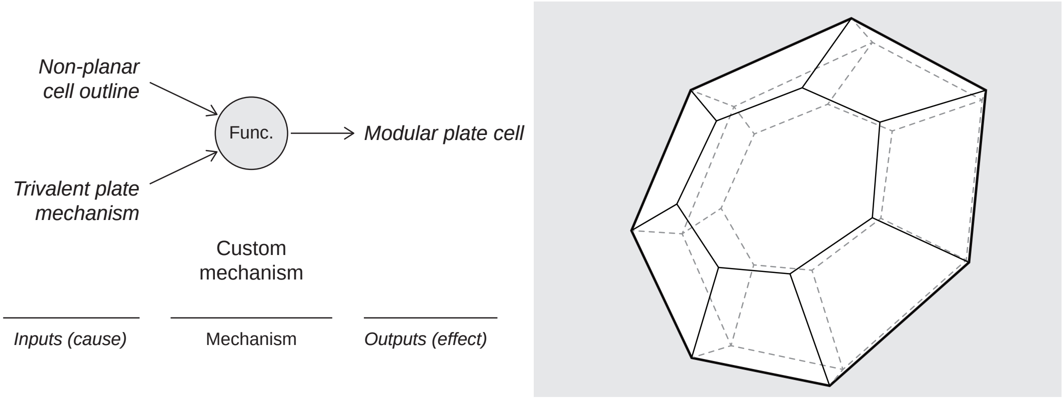


Figure 6.15: Principle of Modular Plate Cell.

Inputs (cause) – Geometrically differentiated cell boundaries as part of a reference model consisting of irregular and non-planar polygons that are the result of the Principle of Parametric Proxy (A.2.2); a mechanism for generating trivalent plate configurations as described in the Principle of Frustum (A.2.3).

Mechanism (function) – In a first step, cell outlines are aligned spatially as well as in terms of number of vertices and edges to form a double-layer configuration of cells. In this configuration, opposing cells that are within a certain proximity to each other are combined into a polyhedral cell by connecting their outlines thus generating a set of quadrilateral surfaces, or flanges, between the corresponding cell edges [100, p. 527]. Second, the flange surfaces are planarized using the PS method and a planarity goal. The flange surfaces then allow connections between neighboring cells. Finally, the Principle of Frustum (A.2.3) is applied to the top and bottom cell outlines to form two nested frusta.

Outputs (effect) – This principle results in a closed polyhedral plate cell composed of planar polygonal plates. For example, following this mechanism, a plate cell based on a hexagonal outline consists of 20 individual plates. The cell's modularity is based on the capacity to adapt to varying boundary conditions that determine the cell's shape while ensuring planarity of all of its components independent of the reference polygon's shape [100, p. 525].

Significance – The cell can locally respond to curvature and shell boundary conditions through the cell's size and extrusion height. The geometric differentiation of its plate components allows for a high degree of adaptability and performance via variable structural depth and a high geometrical and functional robustness [100, p. 525]. The resultant 3-dimensional plate model is the precondition for the principles of Global Cell Arrangement (A.3.4) and Finger Joint Fabrication Modeling (A.5.1). The principle is also a component of the high-level principle of Hierarchy (A.2.5).

It is worth noting that the extrusion of the cell outline topologically results in cells where, at the cell's perimeter, four plates meet in one point. Without noticeably affecting its stability, the plate cells thereby locally deviate from a 'pure' plate structure.

A.2.5 Principle of Hierarchy

The high-level principle of hierarchy associates a plate system with bending-free edges on two levels of hierarchy with a trivalent configuration of the plates within the cell (A.2.3) and the trivalent arrangement of cells within the segmented shell (A.2.1) by subsuming the corresponding principles.

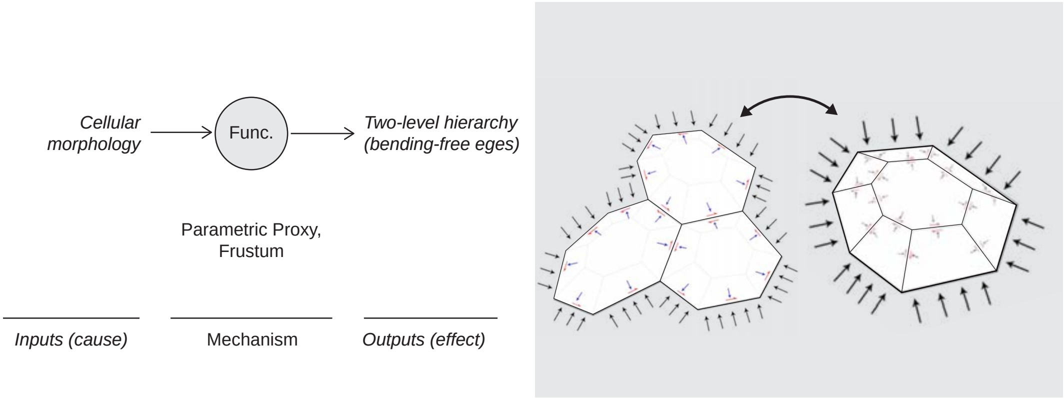


Figure 6.16: Principle of Hierarchy. Right image adapted from [109] by La Magna (2012).

Inputs (cause) – The principle combines the principles of Parametric Proxy (A.2.2), Frustum (A.2.3), and Modular Plate Cell (A.2.4).

Mechanism (function) – According to the Principle of Parametric Proxy (A.2.2), the Voronoi diagram controls the 3-dimensional configuration of cell outlines. By combining the resulting geometrically differentiated cellular configuration with the mechanism described in the principle of Frustum (A.2.3), adaptive polyhedral cells are constructed according to the Principle of Modular Plate Cell (A.2.4).

Outputs (effect) – The result is a hierarchical plate system consisting of two levels [174, p. 162]: on the lower, first level, plywood sheets are joined at the edges through finger joints and glued together to form a plate cell. On the higher, second level, a simple bolt connection joins the cells together (forming a hinged connection), which allows for assembly and disassembly of the pavilion.

Within each hierarchical level only three plates - respectively three cells – meet at one point, therefore assuring bending-free edges on both levels.

Significance – The mechanism implements Bending-free Edges (A.1.5) on the level of the individual plates that constitute the cells and on the level of the cells that constitute the structure. The described principle thus implements a corresponding high-level biomimetic principle, which is based on the observation that in biology, hierarchical organization in the structural morphology of an organism is the basis for its particular performance (see also [92]).

The Principle is a precondition for the Increased Stiffness principle (A.4.4).

A.3 Form-Defining Principles

A.3.1 Principle of Topological Control

The Principle of Topological Control associates the topological programming of a mesh with user-defined inputs points via the application of the mechanism of the Cellular Morphology principle (Fig. 6.17).

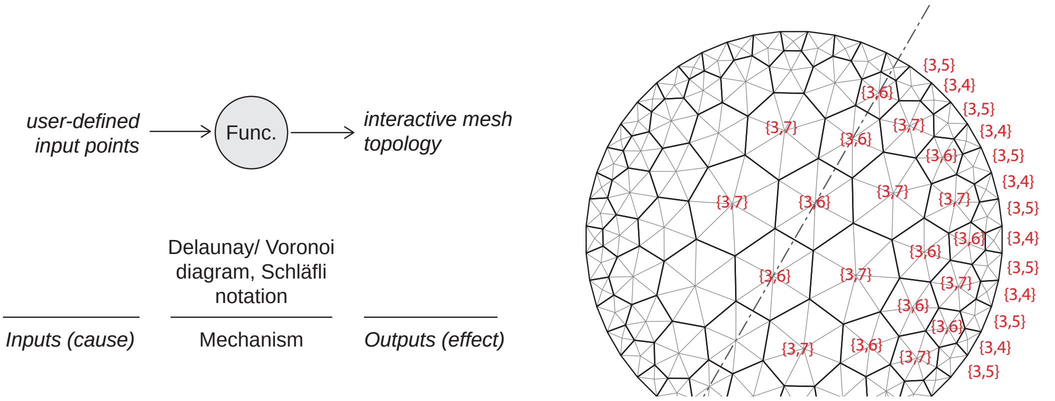


Figure 6.17: Principle of Topological Control. Right image adapted from [100] by Krieg (2012).

Inputs (cause) – A grid of user-defined input points in the 2D plane in concentric circular layout with rising concentration of points towards the perimeter; design drivers mapped to point positions as part of Steering of Form (A.3.8).

Mechanism (function) – The topological programming of the mesh is carried out in two steps: in the first step, following the principle of Cellular Morphology (A.2.1) a triangulated 2-dimensional layout of cells is generated. In the second step, additional input points are strategically inserted by the designer inside the point grid, which allows for interactive control of the number of edges of polygonal Voronoi cells and, correspondingly, of the number of triangles located around the cell centroid. The topology in the 2D model is defined with the aim of aligning polygonal cells in 3D, which leads to a symmetrical layout of input points in order to ensure similar vertex count in the opposing cells. The Schläfli notation thereby gives an indication of the resulting curvature at a corresponding vertex in 3D (see Section 6.2.6).

Outputs (effect) – The result is a configuration of Voronoi cells and corresponding reference mesh whose topology can be interactively manipulated in order to pre-program the form-defining effect of each cell.

Significance – The resulting mesh is the precondition for form-finding, which unfolds via the principle of Dynamic Equilibrium (A.3.3). Its local topology, i.e. the face count of the polygonal cells, has direct influence on the global form-finding behavior of the system. Topological pre-programming thus provides low-level control of the form-defining effect of each cell (synclastic, anticlastic or flat).

A.3.2 Principle of Interactive Edge Control

The Principle of Interactive Edge Control associates the control of the Reference Mesh boundary with architectural and programmatic requirements via interactive manipulation of a 3-dimensional boundary curve and a custom PS-based mapping mechanism from circular outline to 3D boundary curve (Fig. 6.18).

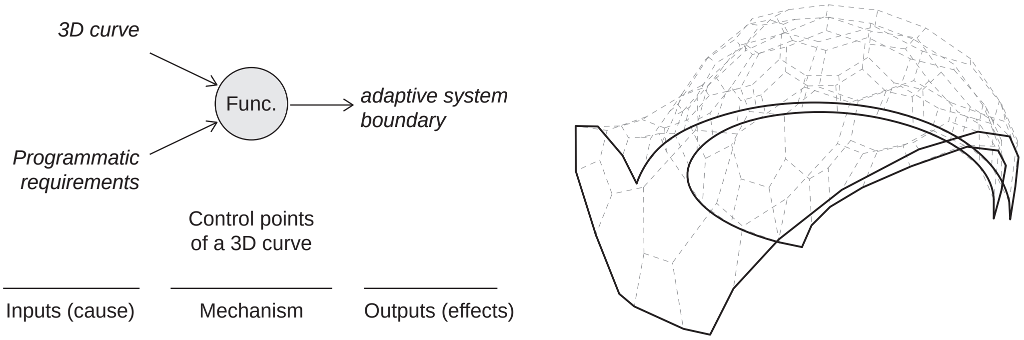


Figure 6.18: Principle of Interactive Edge Control.

Inputs (cause) – Design drivers, such as site, views, and program, that can be mapped to a user-defined 3D curve acting as a guide for the boundary of the reference mesh during form-finding as part of Steering of Form (A.3.8).

Mechanism (function) – By interactively adjusting the control points of the boundary curve, the designer can control the spatial outcome of the form-finding process [100, p. 527].

Particles located at the perimeter of the 2-dimensional cellular layout are mapped to the 3D boundary curve via PS-based mapping. The mapping occurs by modeling springs with zero rest-length and high stiffness between 'naked' vertices on the boundary of the circular 2-dimensional mesh and target points on the user-defined boundary curve. The points on the boundary curve are allowed to slides along the curve to equalize strain in the springs along the mesh boundary.

Outputs (effect) – An adaptive boundary curve that allows the reference mesh, and subsequently the structure, to adapt to architectural and programmatic requirements while remaining in the context of the morpho-spatial capacity of the plate system [100, p. 527].

Significance – This principle defines the boundary condition for the principle of Dynamic Equilibrium (A.3.3). It enables the designer to have intuitive high-level control of the form-finding process.

A.3.3 Principle of Dynamic Equilibrium

The Principle of Dynamic Equilibrium associates a 3-dimensional model in steady-state equilibrium with a triangular mesh via the application of a dynamic equilibrium method such as the PS method (see Section 4.3.1.4) (Fig. 6.19).

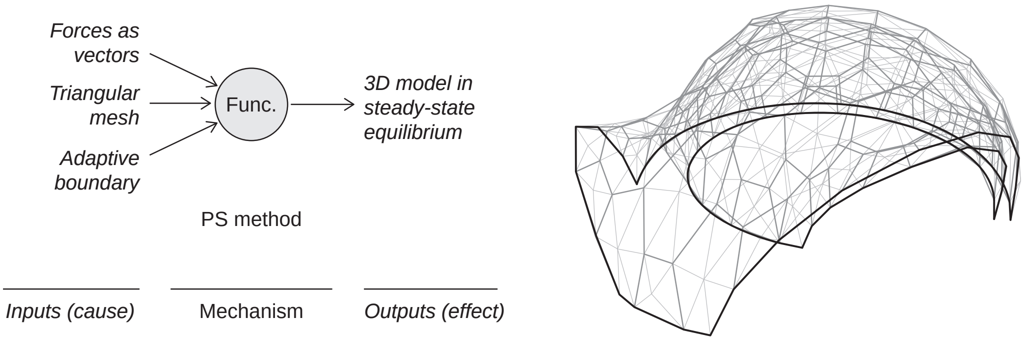


Figure 6.19: Principle of Dynamic Equilibrium.

Inputs (cause) – A triangular mesh as generated by the Principle of Topological Control (A.3.1); an edge curve defining the boundary of the resultant mesh (A.3.2); design drivers that can be mapped to force vectors, such as negative gravity, as part of Steering of Form (A.3.8).

Mechanism (function) – Every edge in the mesh is converted into a spring and assigned a stiffness value and minimum rest length such that all edges relax to approximately the same edge length and, in turn, approximate equilateral triangles. With the application of a nominal mass of the nodes (particles) and a negative gravity force, the relaxation process transforms the 2D mesh into a form-found 3D shape. By combining the dynamic equilibrium and interactive edge control principles, cells can be aligned in the 3D model such that they are facing each other.

Outputs (effect) – A 3-dimensional mesh model in steady-state equilibrium.

Significance – This principle is used as the mechanism in the principle of Parametric Proxy (A.2.2). This principle is the precondition for the principles of Global Cell Arrangement (A.3.4).

A.3.4 Principle of Global Cell Arrangement

The Principle of Global Cell Arrangement associates the distribution and arrangement of polyhedral cells in the 3D model with a 3-dimensional cellular model in steady-state equilibrium via the application of the principle of Modular Plate Cell (Fig. 6.20).

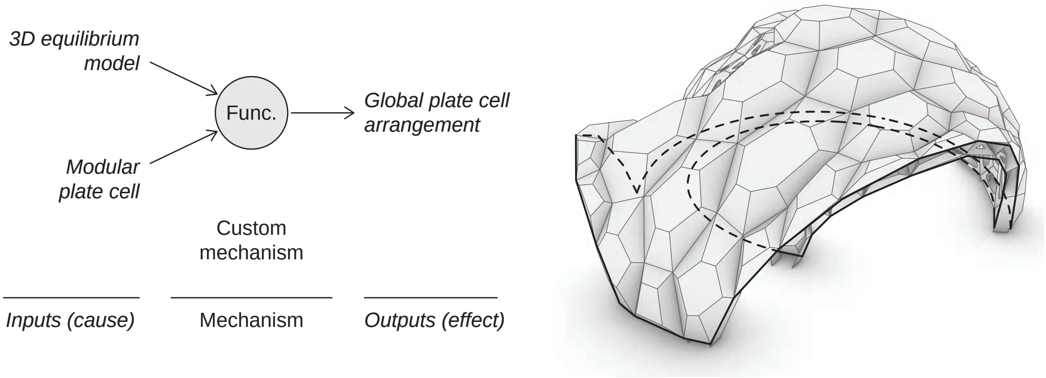


Figure 6.20: Principle of Global Cell Arrangement.

Inputs (cause) – A 3-dimensional model of cellular morphology in steady-state equilibrium as the output of the principle of Dynamic Equilibrium (A.3.3); an adaptive plate cell model as described by the principle of Modular Plate Cell (A.2.4).

Mechanism (function) – In the first step, cells that are aligned and within a certain proximity to each other are merged into a double layer configuration as described in the principle of Modular Plate Cell (A.2.4) by joining edges of opposing cells to form planar quadrilateral flange surfaces. In the second step, the mechanism described in the Frustum Principle (A.2.3) is applied.

Outputs (effect) – A 3-dimensional model of a plate structure composed of polyhedral cells that consist of planar plates.

Significance – The 3-dimensional plate model is the precondition for global structural analysis (A.4.3) and fabrication programming (A.5.3) of the full-scale pavilion, as well as for goal-oriented steering of the plate structure described in the principles of Anisotropy (A.3.5), Variable Plate Size (A.3.6) and Plate Coverage (A.3.7).

A.3.5 Principle of Anisotropy

The Principle of Anisotropy associates the morphologically adapted plate cells with the structure's main load-bearing directions via the principle of Steering of Form (A.3.8) (Fig. 6.21).

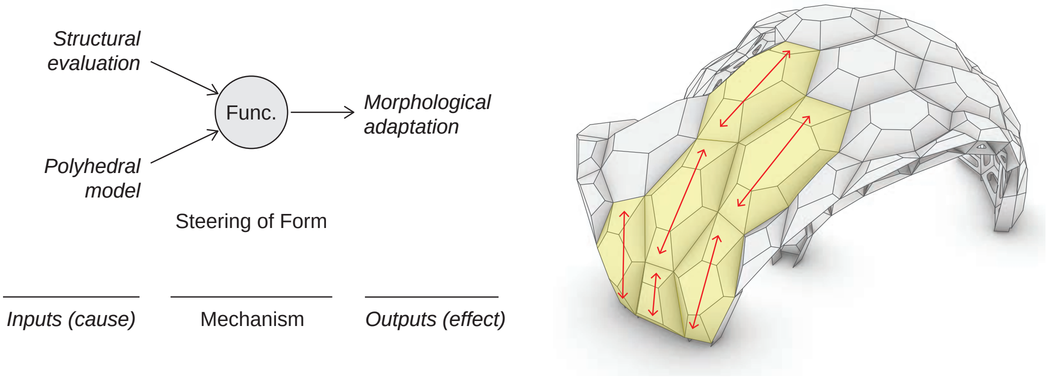


Figure 6.21: Principle of Anisotropy.

Inputs (cause) – This principle operates on the principle stress directions derived from structural analysis (A.4.3) of the full polyhedral plate model (A.3.4).

Mechanism (function) – This principle is an example for the application of the high-level principle of Steering of Form (A.3.8). The designer interactively steers the outcome of the form-finding process by empirically modifying mesh topology, location and strength of design forces that act on the particle spring system and by modifying the edge condition of the system with the objective to minimize stress concentrations in the structure.

Outputs (effect) – The plate cells iteratively adapt to main load-bearing directions for example through stretching of the cells along the main load-paths thereby causing anisotropic load-bearing behavior [100, p. 527]. The result is a morphological adaptation of plate cells according to load-bearing direction, reduced stress concentrations, and the system's boundary conditions.

Significance – The principle implements a closed control loop with the principle of Steering of Form (A.3.8). It allows for an intuitive response to local stress concentrations in the shell through morphological adaptation at the local level of the cell, while meeting architectural and programmatic requirements on the global level.

A.3.6 Principle of Variable Cell Sizes

The Principle of Variable Cell Sizes associates sizes of plate cells with the distance to the boundary of the plate system and with the local curvature of the reference mesh via an adjustment of the density of the input points, i.e. a change in the relative distances between the input points that are the basis for the cellular morphology of the system (Fig. 6.22).

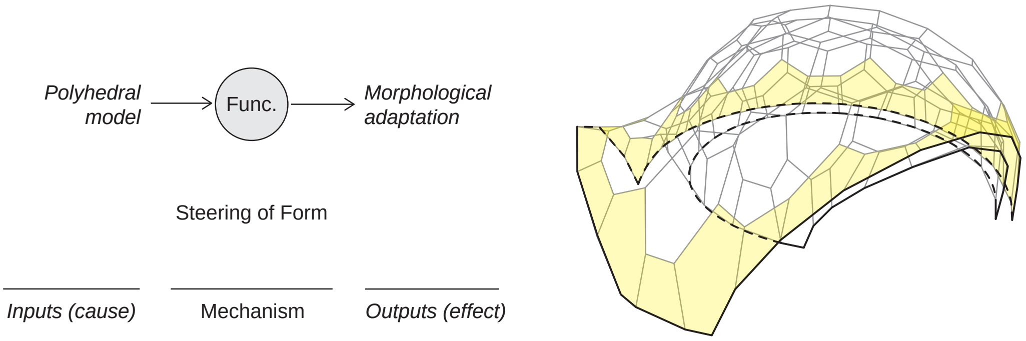


Figure 6.22: Principle of Variable Cell Sizes.

Inputs (cause) – This principle evaluates the relative cell sizes of the full polyhedral plate model (A.3.4).

Mechanism (function) – The principle is an example for the application of the principle of Steering of Form (A.3.8). It mainly operates on the inputs of the principle of Topological Control (A.3.1) in order to adjust cell sizes: (1) the target size of a plate cell is directly proportional to the distance of the cell centroid to the edge of the plate shell within the minimum and maximum limits of allowable cell sizes, meaning that cells are smaller the closer they are to open edges of the system and larger the further away they are located. (2) based on the same user-defined range of minimum and maximum cell sizes, the size of a cell is indirectly proportional to the curvature of the reference mesh at the location of the cell centroid: cells are smaller the higher the curvature is and vice versa. In both cases the adaptation in cell sizes is achieved by locally changing the density of / the relative distance between input points as described in the principle of Cellular Morphology (A.2.1).

The effects of these two low-level functional principles have the potential to cancel each other out, e.g. in the case of high curvature away from the system edge. This effectively means that one effect has to be prioritized over the other: in this case, the prioritization is in favor of smaller cell sizes such that areas of small curvature (large cell sizes) at a small distance to the edge (small cell sizes) leads to small cells and, conversely, areas of high curvature (small cell sizes) at a larger distance to the edge (large cell sizes) also lead to small cell sizes.

Outputs (effect) – The result is an adaptation of cell sizes as a function of the described criteria of distance and curvature. These criteria are analyzed and evaluated individually at the level of each cell.

Significance – The principle incorporates two lower-level functional principles with potentially opposite effects and thus provides an example for a prioritization / trade-off mechanism. The principle implements a closed control loop through the principle of Steering of Form (A.3.8). This feedback rule was empirical and parameter adjustments were implemented manually.

A.3.7 Principle of Plate Coverage

The Principle of Plate Coverage associates required resources, and the fulfillment of programmatic and logistical requirements with the number and sizes of building elements (cells and plates) of the full polyhedral model via a trade-off mechanism (Fig. 6.23)

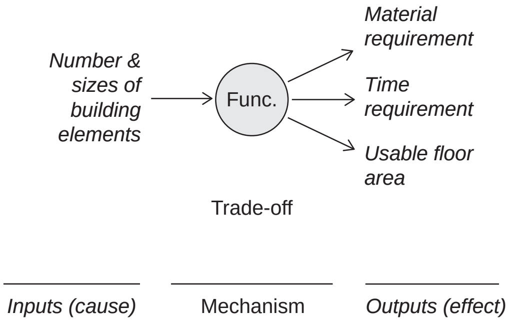


Figure 6.23: Principle of Plate Coverage.

Inputs (cause) – This principle evaluates (1) the number and sizes of building element (plates and cells) given as the outcome of the principle of Global Cell Arrangement (A.3.4) with respect to specific contextual parameters such as available resources (material, time, cost, etc.) and programmatic requirements, e.g., in terms of usable floor area to be covered.

Mechanism (function) – A trade-off mechanism is defined between, on the one hand, minimizing plate sizes for ease of handling and increased geometric flexibility, which leads to an increase in number of plates; and, on the other, minimizing the overall number of plates, which leads to increasing plate sizes and reduced geometric flexibility.

For example, a single hexagonal plate cell consisting of 20 plates with a given average production time per plate effectively places an upper limit on the total number of plates and, in turn, on the number of plate cells being producible within a fixed production timeframe. Changing pavilion size (usable floor area), e.g., by tweaking the boundary curve using the principle of Interactive Edge Control (A.3.2), and adjusting the number of plate cells through the number of input points using the principle of Cellular Morphology (A.2.1), allows approximating an 'optimal' relationship between average plate size and their total number. The principle thus is a further example of the application of the Principle of Steering of Form (A.3.8).

The feedback loop involves logistical requirements (material, time, cost, transportation) and the fulfillment of programmatic requirements by assigning a score at each iteration, which can be used to steer the design into a configuration that sits within the boundaries defined by the contextual parameters.

Outputs (effect) – The results of this principle can be evaluated from logistical, programmatic, and design point of views. (1) logistical, in terms of available resources, production time, transportation, and handling; (2) programmatic, in terms of usable floor area that can be covered with respect to the design brief; and (3) design-oriented, in terms of demonstrating the flexibility of the material system through a global shape of sufficient geometric complexity.

The number of resultant plates (855) and plate cells (59) is based on the described feedback-driven trade-off mechanism defined by the limits of available resources and production time.

Significance – Controlling the number and sizes of plate cells thus ensures suitable plate dimensions for transportation and fabrication (turntable positioning), while matching the number of elements to the available resources and production time, and meeting the design objective [174, p. 163].

The principle implements a closed control loop through the principle of Steering of Form (A.3.8).

A.3.8 Principle of Steering of Form

The high-level Principle of Steering of Form associates a model in steady-state equilibrium with programmatic, structural and economic design drivers represented as geometry (e.g. input points and edge curve) and as force vectors via a design environment implementing dynamic equilibrium methods (Fig. 6.24).

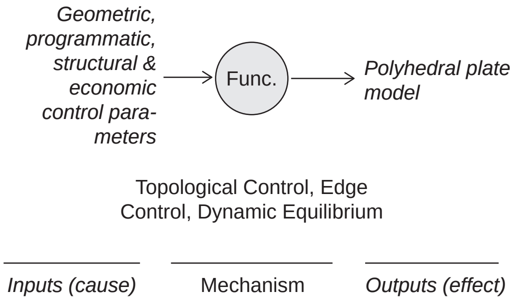


Figure 6.24: Principle of Steering of Form.

Inputs (cause) – Inputs are goal-oriented design drivers such as load-bearing capacity (A.3.5), cell sizes (A.3.6), available resources (A.3.7), and producibility (A.5.7).

Mechanism (function) – Design drivers are mapped to parameters, either in the form of numbers and geometry, as in the case of the principles of Topological Control (A.3.1) and Interactive Edge Control (A.3.2), or as forces as in the case of Dynamic Equilibrium (A.3.3). While the mesh is responding to default load-cases such as self-weight and wind during the form-finding process, the designer can adjust the mesh topology and thereby the polygonal representations of each cell in order to find a structurally performative shape. The boundary condition of the particle system can be adjusted in 3D space as a means to incorporate architectural and programmatic requirements.

In a physically driven design environment gravity and additional design drivers, such as program or views, can be represented as forces that act on the nodes/ particles in the system in order to steer the resultant configuration.

Outputs (effect) – The result is a model in steady-state equilibrium of forces where design drivers are implemented and 'played out' in addition to structural criteria such as self-weight and standard load cases.

Significance – The addition of forces as design drivers in the dynamic equilibrium approach facilitates the response to structural as well as architectural demands within the same modeling approach. In this process the architect acts as a 'mediator' trying to accommodate potentially conflicting goals by controlling topology, force components and boundary condition as the dynamic equilibrium process unfolds until the model converges to a desirable solution. This approach enables a high-level, interactive control of the form-steering process (see also [100]). This principle thus combines the main lever points for the architect to steer the design process.

Through the principle of Dynamic Equilibrium (A.3.3), this principle affects the Global Cell Arrangement (A.3.4).

A.4 Structural Design Principles

A.4.1 Principle of Trivalence

The Principle of Trivalence associates an abstract model of a trivalent plate configuration (where always three plates meet in one point) with the analysis of biological segmented shells, such as sea urchins, as part of a biomimetic transfer process (Fig. 6.25).

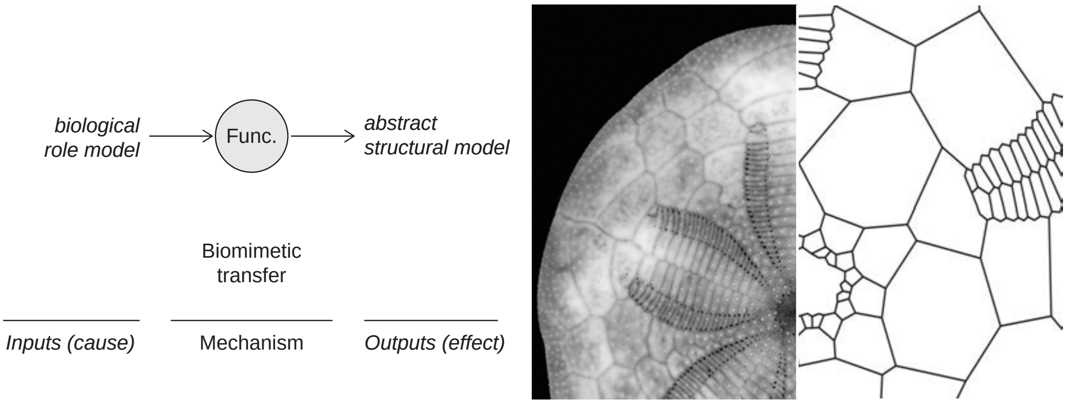


Figure 6.25: Principle of Trivalence. Center-right: Top view of sea urchin skeleton. From [130] by van der Steld (2012). Right: Schematic view of plate arrangement. From [100] by Krieg (2012).

Inputs (cause) – The principle is based on the analysis of biological role models, in this case the functional morphology of sea urchins.

Mechanism (function) – Analysis of a biological role model, abstraction of principle and transfer to a structural model following a biomimetic design methodology (see [138, p. 192]); Structural analysis of the plate-lattice dualism as described in [108].

Outputs (effect) – A structural model of plate-lattice dualism stating that any structure complying with the 3-plate principle is inherently stable [108, p. 6]

Significance – The most important functional aspect of the sea urchin's plate morphology, from a structural point of view, is its compliance to the 3-plate principle, meaning that it is fully trivalent and can thus be regarded as a pure plate structure [108, p. 4] (see also [208]).

This output of this principle is the precondition for the principles of Bending-free edges (A.1.5) and Structural Validation (A.4.2).

A.4.2 Principle of Structural Validation

The Principle of Experimental Validation associates a validated structural model of finger joint connections with selected finger joint samples and an abstract structural model via physical load testing (Fig. 6.26).

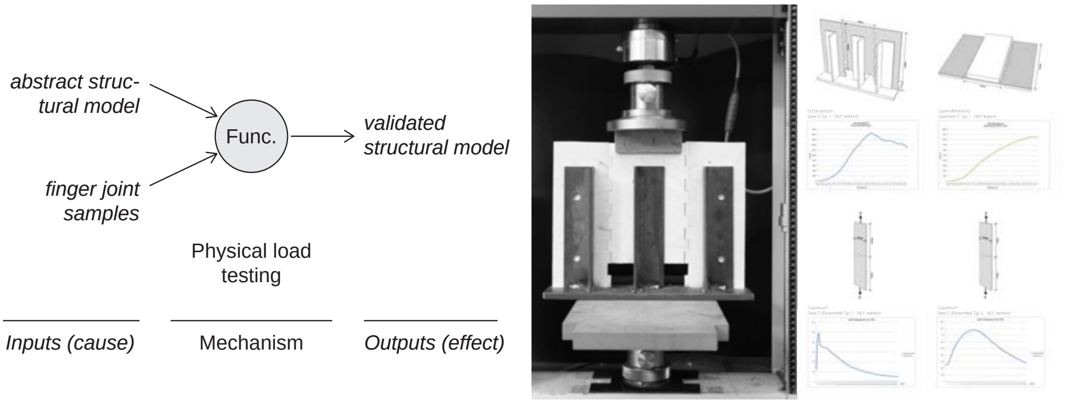


Figure 6.26: Principle of Structural Validation. Right: Load testing of finger joints. From [108] by ICD/ITKE University of Stuttgart (2011).

Inputs (cause) – Robotically fabricated finger jointed plate samples with selected angles including co-planar configuration (A.1.2); an abstract structural model (A.4.1) for which the characteristic values/ parameters, such as bending stiffness, shear capacity and failure criteria need to be defined.

Mechanism (function) – Physical load tests of the connection samples were performed in different setups for lateral, normal and shear force transmission and for stiffness behavior [109, p. 33].

Outputs (effect) – Validated structural model of the connection and its characteristic values. Very good results were achieved for the shear strength and low bending stiffness of the rotational DOFs [109, p. 33].

Significance – Defining parameters of the structural model including the failure criteria of the joints is the precondition for incorporating the characteristic values of the connection in the Finite Element model of the full plate structure [109, p. 33]. Consequently, this principle is the precondition for the principles of Automated FEM (A.4.3).

A.4.3 Principle of Automated FEM

The Principle of Automated Finite-Element-Modelling (FEM) associates a complex and intricate FE-model with a validated structural model of finger joints and a variable, geometrically differentiated design model via a computational approach towards structural modeling (Fig. 6.27).

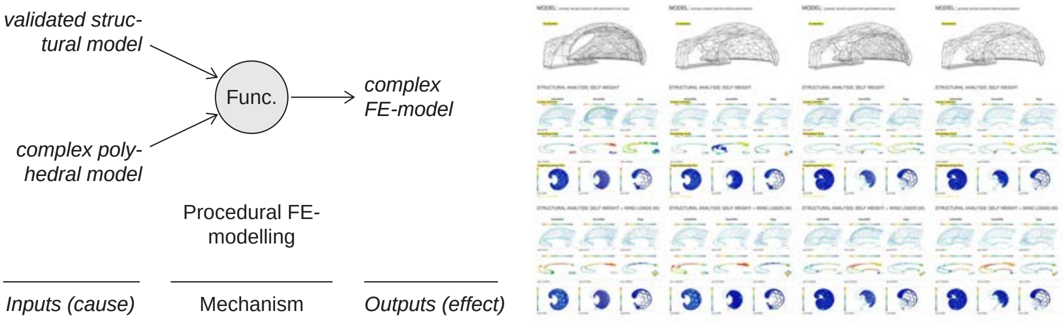


Figure 6.27: Principle of Automated FE-Modeling. Right images from [109] by ICD/ITKE University of Stuttgart (2011).

Inputs (cause) – A geometrically differentiated design model, such as the one generated by the principle of Global Cell Arrangement (A.3.4), that is subject to design changes due to feedback mechanisms (see A.3.8). A validated structural model of finger joints, as produced by the Principle of Experimental Validation (A.4.2), defining characteristic values of the joints.

Mechanism (function) – Custom tools for the FE-Solver were developed and implemented in order to automate meshing and definition of plane shell elements followed by modeling the finger joint connections as axial springs in the direction of the edges of connection between the plates [109, p. 34]. The stiffness values of the linear springs were obtained from the laboratory tests on the shear strength of the finger joints [108, p. 6].

Outputs (effect) – A FE-Model with more than 10.000 spring elements that allows simulating the structural behavior of finger-joints in the cell elements as well as the global structural behavior [109, p. 34].

Significance – Automated FEM enabled by custom tools and approaches allows for design exploration and iterative optimization of shell design. The principle is a precondition for form-finding based on structural criteria (A.3.5), as well as for the iterative studies in A.4.4 and A.4.5. Finally, it is also a component in the high-level principle of Free-form Segmented Shell (A.4.6).

A.4.4 Principle of Increased Stiffness

The Principle of Increased Stiffness associates the increase of stiffness in a double-layer plate shell, when compared to a single-layer configuration, with plate cells that form closed units via the increased stiffness of the units (Fig. 6.28).

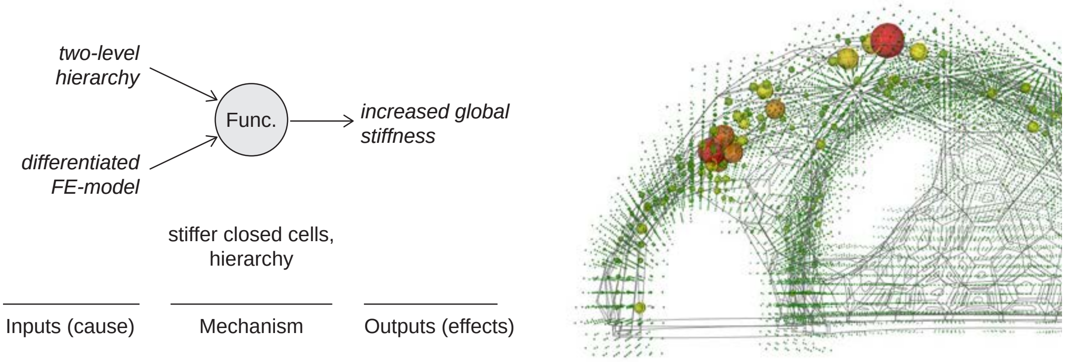


Figure 6.28: Principle of Increased Stiffness. Right image from [174] by Schwinn (2011).

Inputs (cause) – A plate structure based on the principle of Hierarchy (A.2.5) with a trivalent configuration on two levels of hierarchy; a differentiated FE-Model as defined by the principle of Automated FEM (A.4.3).

Mechanism (function) – Due to the tolerances in the finger joints that are necessary for assembly, the connections between plates remained relatively deformable until plates were joined into closed polyhedral units without open plate edges. In this case, the plate cells exhibited extraordinary stability [100, p. 528], which correspondingly increased stiffness globally. The effect is enabled by the principle of Hierarchy (A.2.5), where the shell, instead of being composed of many small plates, is composed of fewer and stiffer plate cells, which, in turn, are composed of smaller plates thus forming a hierarchical system.

Outputs (effect) – The double-layer configuration increased the structural depth of the plate cell which increased its bending stiffness. Added stiffness locally, on the level of the plate cell, thus corresponded to an increased structural capacity globally and a reduction of deflection in the plate system.

Significance – The effect of increased stiffness was confirmed using 3d-scanning, which showed that deflection in the single-layer zones was significantly larger than in the double-layer zones [100, p. 528] (Fig. 6.28 right).

This principe thus allows increasing the span of the plate structure without loosing geometrical flexibility and geometric resolution of the shell through increased plate sizes. It is also a component in the high-level principle of Free-form Segmented Shell (A.4.6).

A.4.5 Principle of Stress Reduction

The Principle of Stress Reduction associates reduced stress within plates with plate cells that are composed of convex plates. This was confirmed via a comparison of stress concentrations between convex and concave plates using structural analysis (Fig. 6.29).

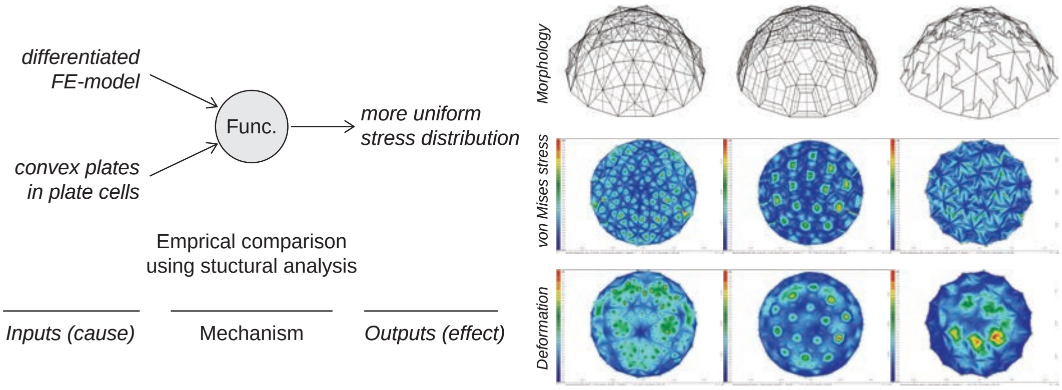


Figure 6.29: Principle of Stress Reduction. Right: Comparision of stress concentrations. From [109] by ICD/ITKE University of Stuttgart (2011).

Inputs (cause) – FE-Models of modular plate cell variations as defined by the principle of Automated FEM (A.4.3); those variations were composed of convex and concave plates generated using different subdivision strategies.

Mechanism (function) – FEM and structural analysis of the different plate cell models and the subsequent comparison of analysis results.

Outputs (effect) – The result of the analysis was that concave plates lead to high stress concentrations in the corners for overall loading and boundary conditions [109, p. 32], while plate cells with convex plates exhibit significantly less stress concentrations and more uniform stress distributions.

Significance – The conclusion was that concave plates had to be avoided and, consequently, a subdivision strategy was chosen that results in convex plates while also meeting the trivalence requirement as defined in the principle of Principle of Frustum (A.2.3). As such, the principle is also a component in the high-level principle of Free-form Segmented Shell (A.4.6).

From the point of view of reverse biomimetics, the stress concentrations in concave plates might be a reason why such plates do not exist in nature and, particularly in the case of sea urchins, the plate morphology is always convex [109, p. 33].

A.4.6 Principle of Free-form Segmented Shells

The high-level principle of Free-form Segmented Shells associates bending-free joints and global bending-bearing structural behavior with ‘pure’ plate structures and free-form shell geometry via geometric locking and shell action in the plates (Fig. 6.30).

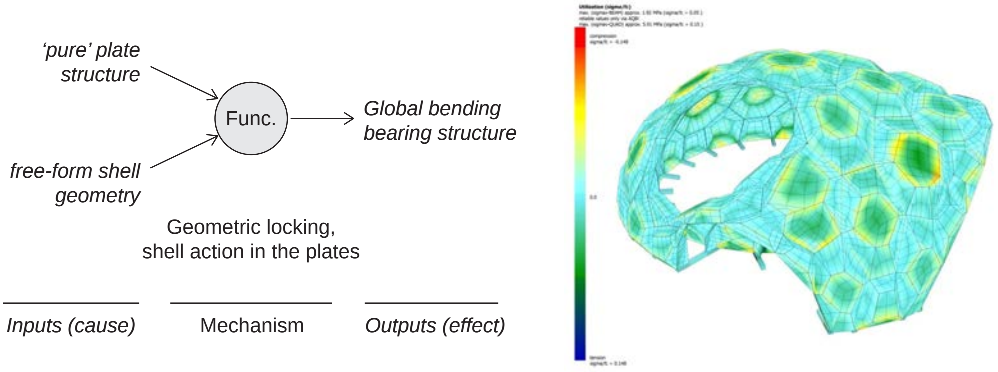


Figure 6.30: Principle of Free-form Segmented Shells. Right: FE-Model. From [108] by ICD/ITKE University of Stuttgart (2011).

Inputs (cause) – The principle aggregates the principles of Automated FEM (A.4.3), Increased Stiffness (A.4.4), and Stress Reduction (A.4.5); it is based on free-form plate shell generated according to the Principle of Global Cell Arrangement (A.3.4) exhibiting the organizational principles of ‘pure’ plate structures.

Mechanism (function) – The influence of different trivalent plate arrangements on the structural behavior of free-form shells and of the joints was evaluated through Automated FEM (A.4.3) and structural analysis.

Outputs (effect) – The simulations showed global shell action and membrane forces in the plates (plate action) in spite of the non-load-optimized shape. Furthermore, the FE results confirmed that for pure plate structures of free-form (meaning non-load-optimized) geometry and loading no bending moments occur in the joints. It also showed that the lateral forces in the edges and the global deformation of the structure were greater than the equivalent continuous model with bending moment transferring capacity [109, p. 32].

Significance – Unlike traditional lightweight construction, which can only be applied to load-optimized shapes, this new finding can be applied as a design principle to a range of freeform segmented shells while still resulting in shell action [109, p. 34]. This is one of the most significant findings with regards to designing segmented timber shells.

A.5 Fabrication Design Principles

A.5.1 Principle of Finger Joint Fabrication Modeling

The Principle of Finger Joint Fabrication Modeling associates a computational approach for finger joint fabrication modeling with a finger joint production sequence via the mathematical description of the relationship between work piece and milling tool (Fig. 6.31).

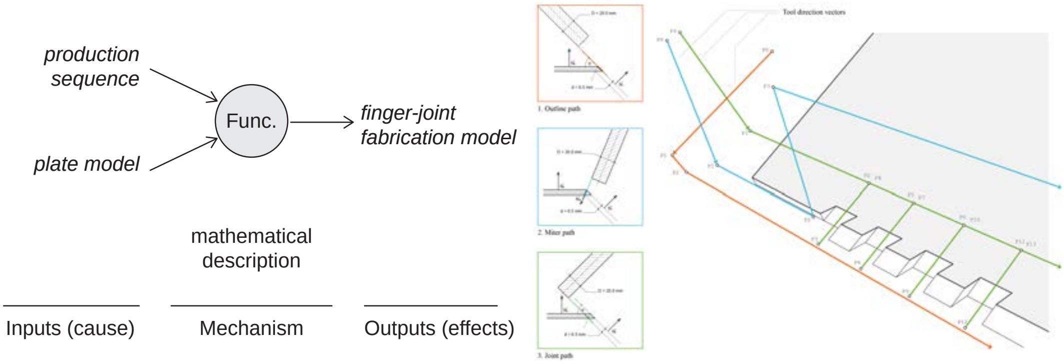


Figure 6.31: Principle of Finger Joint Fabrication Modeling. Right images from [171] by Schwinn (2011).

Inputs (cause) – The finger joint fabrication sequence as described in the Principle of Finger Jointing (A.1.1); digital plate model as produced by the Principle of Modular Plate Cell (A.2.4); additional parameters such as finger joint width and material thickness.

Mechanism (function) – The different tool paths for the fabrication of a plate are a function of the angles between the plate and its neighboring plates [109, p. 35]. The spatial relation between work piece and milling effector during fabrication sequence could consequently be described mathematically using trigonometry and linear algebra [174, p. 162]. The model consisted of ordered point clouds that represent tool positions and orientations and that were visually represented as polylines.

Outputs (effect) – A parametric rule-based description that underlies finger joint cutting enabled the procedural generation of tool paths. A robust and repeatable CAM digital process for finger jointing was developed resulting in a finger joint fabrication model.

Significance – This principle is a precondition for Automated Fabrication Modeling (A.5.3).

A.5.2 Principle of Scalability

The Principle of Scalability associates a computationally lightweight fabrication data model with geometrically complex finger joints via a custom data model (Fig. 6.32).

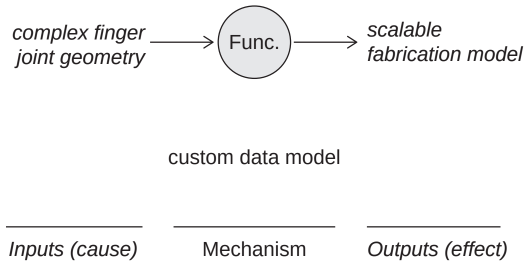


Figure 6.32: Principle of Scalability.

Inputs (cause) – Finger joint connections are geometrically complex (A.1.2) and therefore an accurate three-dimensional representation can put considerable load on the data model.

Mechanism (function) – The generation of fabrication data is implemented independently from the geometric representation and visualization of the finger joint model. The fabrication model consists of 'lightweight' proxy data objects such as ordered point clouds representing tool paths, which are generated in relation to polygonal Non-Uniform Rational Basis Spline (NURBS) patches defining the geometry model. Material type and thickness are stored in alphanumerical form [174, p. 163].

Outputs (effect) – A computationally lightweight fabrication data model from which both the fabrication data and the visualization model are derived.

Significance – The data structure of the digital information model favors scalability over representation: contrary to commercial CAM packages where the geometric representation of finger joints is the basis for fabrication modeling, in this approach no model of the final geometry is required. The scalable data structure enables the integral generative definition of the more than 100,000 individual finger joints without affecting user interaction through potentially limited responsiveness of the model.

This principle is a precondition for the CAM process described in the principle of Automated Fabrication Modeling (A.5.3).

A.5.3 Principle of Automated Fabrication Modeling

The Principle of Automated Fabrication Modeling associates a digital fabrication model with a geometry model and a finger joint modeling approach via a parametrization of the modeling approach (Fig. 6.33).

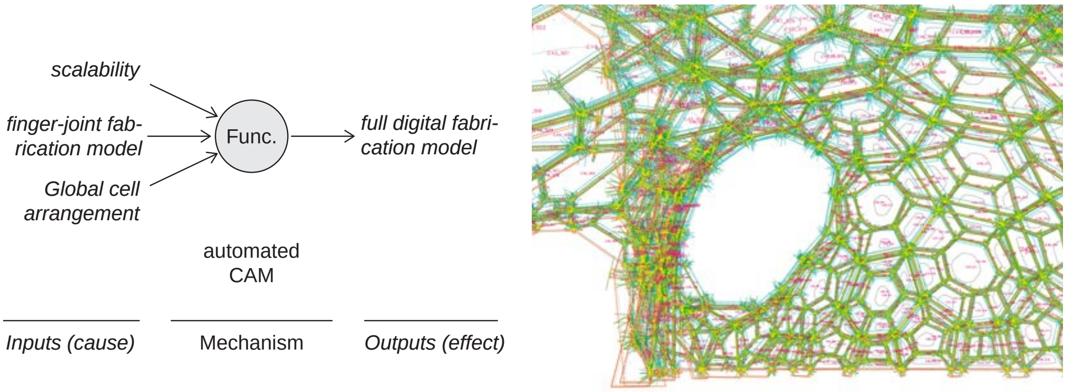


Figure 6.33: Principle of Automated Fabrication Modeling. Right image from [130] by Schwinn (2011).

Inputs (cause) – A finger-joint modeling approach as in the principle of Finger Joint Fabrication Modeling (A.5.1); plate geometry model such as the output of the principle of Global Cell Arrangement (A.3.4); a scalable data model for finger joint fabrication (A.5.2).

Mechanism (function) – Based on the analysis of topology within the plate cell, adjacency between plates is determined. This allows for calculating angles between plates that are adjacent to a given edge in the polyhedral plate cell, including computing concave and convex relationships between plates. The angle between the adjacent plates and material thickness are parameters for functions that return ordered point clouds for the different steps in the fabrication sequence described in the principle of Finger Joint Production (A.1.1) [174, p. 164].

Outputs (effect) – The full digital fabrication model consists of tool paths for contour cutting, mitering and spot facing generated in the sequence in which these are robotically actuated [174, p. 164]. The tool paths are associated with their respective parent plates via unique tool path IDs. The resulting digital fabrication model encapsulates machining information for each of the case's 855 plates, tool paths for its more than 100,000 finger joints, and bolt locations for the impermanent hinged connection between the cells [100, p. 528].

Significance – Tool path information for the production of all the unique plate elements can be automatically generated. The principle therefore overcomes the limitations of the standardized features and serial production constraints and is a precondition for the principle of NC-Code Generation (A.5.4).

A.5.4 Principle of NC-Code Generation

The Principle of NC-Code Generation associates the automatic generation of machine code files with a digital fabrication model via the automated extraction of position and orientation information (Fig. 6.34).

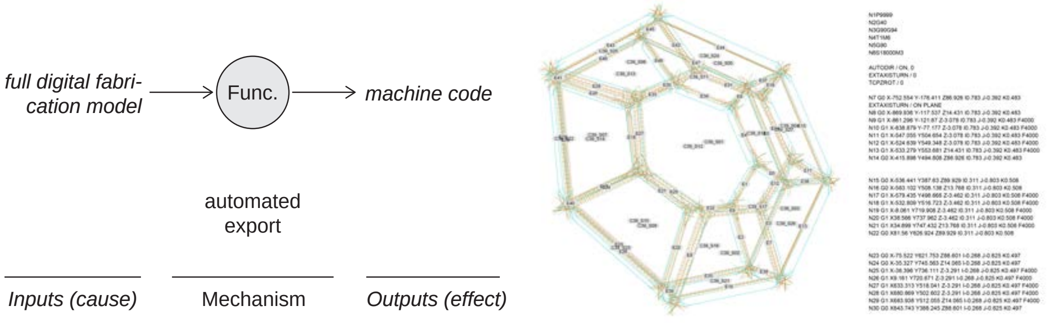


Figure 6.34: Principle of NC-Code Generation. Right image from [174] by Schwinn (2011).

Inputs (cause) – A digital fabrication model containing tool paths with uniquely identifiable tool paths (A.5.3).

Mechanism (function) – Tool path information is exported as structured data in Cartesian coordinates describing position and orientation of the tool center point (TCP) for linear continuous path movements in the sequence that they are robotically actuated. Global coordinates of the fabrication model in world space are transformed to a local coordinate system (LCS) positioned on the plate centroid. The local coordinates in the LCS then are equal to the coordinates in the base coordinate system (BCS) in which the raw plate is machined, in this case the turntable: the origin of the BCS is located in the center of the turntable; the plate's normal vector corresponds to the BCS z-axis, and the direction of the plate's longest edge corresponds to the BCS x-axis. The point data is augmented by fabricational information, such as machining speed (feed) and the spindle's rotational speed, as a function of material and cutting tool [174, p. 164].

Outputs (effect) – One fabrication file is generated per plate containing the NC-code for all three milling job cycles.

Significance – In this way, each plate could be unambiguously positioned in the BCS without having to reorient and reposition each individual plate in the 3D model. Automating NC-code generation is a precondition for mass-customized manufacturing and one-off and batch production paradigms [15].

This principle provides inputs to Open CNC (A.5.6) and Fabrication Validation (A.5.7).

A.5.5 Principle of 7-Axis Robotic Fabrication

The Principle of 7-Axis Robotic Fabrication associates a kinematic model of the 7-axis fabrication setup with specific machine hardware via the mathematical description of the robot kinematics (Fig. 6.35).

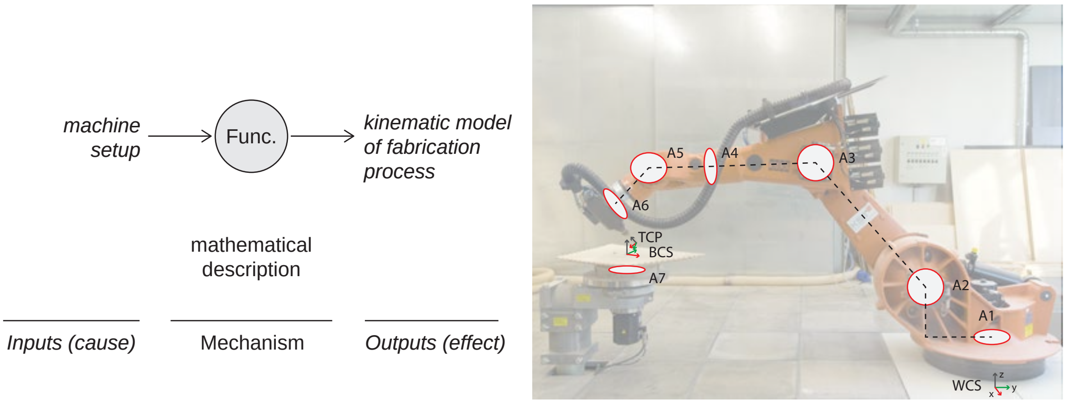


Figure 6.35: Principle of 7-Axis Robotic Fabrication.

Inputs (cause) – A 6-axis industrial robot arm with flange-mounted spindle and a kinematically coupled vertical positioner (turntable) as seventh axis.

Mechanism (function) – The additional DOF provided by the external axis in the 7-axis fabrication setup allows the 6-axis robot arm to synchronously reorient the work piece and therefore the trimming of a stock piece from every direction in one machining sequence.

Outputs (effect) – An efficient and time-saving fabrication process compared to 6-axis fabrication, which required repositioning of the work piece during fabrication (see the fabrication process described in the preliminary work 6.2.1).

Significance – The principle is the precondition for realizing the plate system's geometric differentiation. The fabrication setup extends the design space considerably further than that of traditional CNC machinery as it now allows the production of complex finger joint plate connections [174, p. 129].

The kinematics of the fabrication setup are a precondition for the principles Connecting Angles (A.1.3), Fabrication Validation (A.5.7), and the high-level principle of Machinic Morphospace (A.5.8).

A.5.6 Principle of Open CNC

The high-level Principle of Open-CNC associates brand independence with NC-Code that can be executed independent of a particular machine manufacturer via code generation according to ISO-standard (Fig. 6.36).

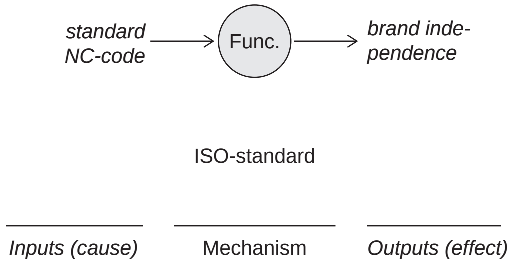


Figure 6.36: Principle of Open CNC.

Inputs (cause) – Machine code generated as described in NC-Code Generation (A.5.4) defining TCP positions and orientations without specifications other than the minimum number of axes needed for fabrication.

Mechanism (function) – Methods described in the Principle of NC-code Generation (A.5.4) with a focus on ISO-conformity (ISO 6983) [174, p. 164].

Outputs (effect) – Reusable code and open access.

Significance – Offers the possibility of sharing code with contractors that use either different robots or specific machining configuration. By utilizing NC-code as opposed to robot-specific code, the fabrication becomes independent of a specific robot brand [174, p. 165]. On the contractor side, in this scenario, the code needs to be validated with respect to a specific fabrication setup, which requires a post-processor that translates the generic code to machine specific code similar to the process described in the principle of Fabrication Validation (A.5.7).

A.5.7 Principle of Fabrication Validation / Post-Processing

The Principle of Fabrication Validation associates a validated fabrication process and machine-specific code with a 7-axis machine configuration and standard NC-code via kinematic simulation of the fabrication process (Fig. 6.37).

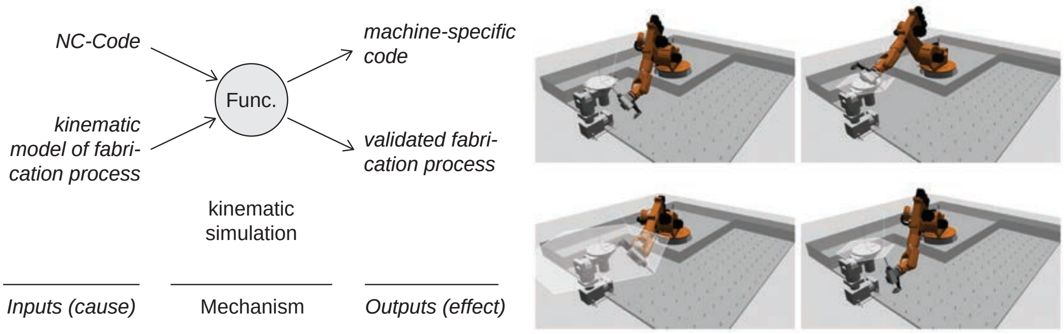


Figure 6.37: Principle of Fabrication Validation. Right images from [129] by Krieg (2012).

Inputs (cause) – A kinematic model of the 7-axis machine configuration Principle of 7-Axis Robotic Fabrication (A.5.5); standard NC-code in Cartesian coordinates Principle of NC-Code Generation (A.5.4).

Mechanism (function) – Reverse transformation of Cartesian coordinates into the angular joint space of the industrial robot was performed in a dedicated post-processor [171, p. 58]. Simulation of the tool path with respect to the corresponding robot poses enables the designer / fabricator to anticipate singularities, out-of-reach-positions, and collisions with the work piece and the robot itself. Modification of effector or workpiece orientation, or of the plate model itself then ensured a feasible and collision-free fabrication process. Once the fabrication process was validated, machine-specific control code was generated by the post-processor, in this case QDesign Robomove.

Outputs (effect) – A validated fabrication process corresponding to a validation of the design geometry with respect to its producibility; generation of machine-specific control code.

Significance – The ability to simulate the fabrication process is the precondition for tying the simulation results back into the geometric design as part of the Steering of Form (A.3.8), as well as for describing the space of producible form as part of the high-level principle of Machinic Morphospace (A.5.8).

Finally, the validation of the fabrication process for each individual element in the construction through simulation is the precondition for the fabrication of the elements (A.6.1).

A.5.8 Principle of Machinic Morphospace

The Principle of Machinic Morphospace associates the delineation of the space of producible form with a given machine setup and fabrication process via kinematic simulation and validation of the fabrication process (Fig. 6.38).

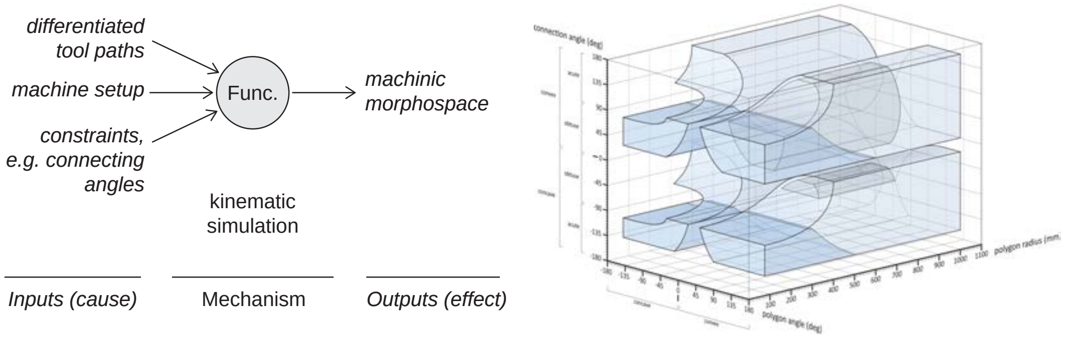


Figure 6.38: Principle of Machinic Morphospace. Right image from [129] by Krieg (2012).

Inputs (cause) – A specific machine setup for fabrication (A.5.5) and a definition of fabrication limitations as described in the principle of Connecting Angles (A.1.3). A mechanism for fabrication validation as described in the principle of Fabrication Validation (A.5.7).

Mechanism (function) – Kinematic simulation of the fabrication process as described in the Principle of Fabrication Validation (A.5.7) allows identifying the limits of the geometric solution space with respect to minimum and maximum plate size (radius), connecting angles between the plates (Principle of Connecting Angles), and interior polygon angles.

Outputs (effect) – The delineation of the geometric solution space associated with a specific machine setup.

Significance – The definition of the machinic morphospace is a precondition for fabrication-oriented design integration.

In this case, it provides the permissible domains for the geometric attributes of the plates, such as size, connecting angles, and interior polygon angles, and thus provides input to the principle of Frustum (A.2.3).

A.6 Production Principles

A.6.1 Principle of Robotic Pre-Fabrication

The Principle of Robotic Pre-Fabrication associates pre-fabrication of finger-jointed plate structures with the use of a 7-axis robotic fabrication process via off-site fabrication in the context of a workshop, which provides the necessary infrastructure to safely operate an industrial robot (Fig. 6.39).

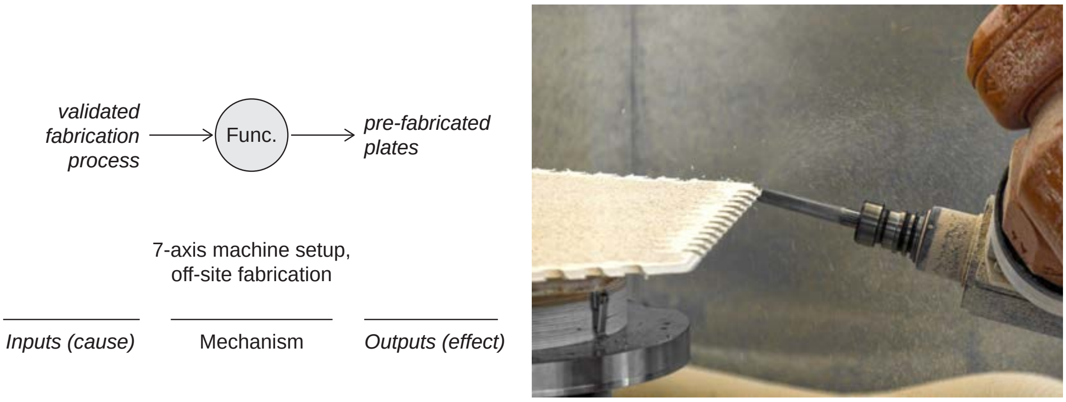


Figure 6.39: Principle of Robotic Pre-Fabrication. Right image from [174] by Krieg (2011).

Inputs (cause) – The principles's input is a validated robotic fabrication process as defined in the principle of Fabrication Validation (A.5.7) for a 7-axis robotic fabrication setup consisting of an industrial robot arm and turntable as additional axis.

Mechanism (function) – All elements of the plate structure were robotically fabricated in-house in the university's Robolab. In the first step, in order to minimize cut-off and material waste, plates of different plate cells were nested together on sheets of 2.5 m x 1.25 m and cut out with oversize by the robot. In the following step, the individual plates were affixed to the turntable where the fabrication sequence of outlining, mitering, and finger joint cutting took place.

Outputs (effect) – Robotically pre-fabricated finger-jointed plates.

Significance – The robotic process is limited to the machining of the plates. Pre-fabrication of plates thus is the precondition for pre-assembly and weather-proofing of plate cells (A.6.3).

On the one hand, robotic pre-fabrication requires segmentation of a shell structure and, in turn, places limits to the size and weight of segments due to transportability and handling constraints. On the other, robotic pre-fabrication makes the benefits of pre-fabrication, such as controlled fabrication environment, quality control, reduction of on-site labor, applicable to shell structures.

A.6.2 Principle of Assemble-ability

The Principle of Assemble-ability associates (1) the ability to pre-assemble plate cells off-site from individual plates with finger joints and plate arrangement; and (2) the ability to assemble the structure on-site from plate cells with the global arrangement of plate cells via connection details and assembly sequencing (Fig. 6.40).

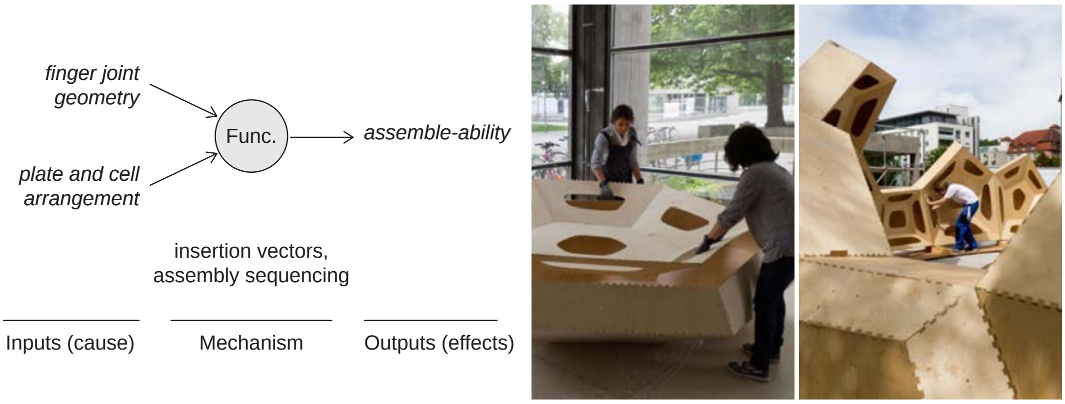


Figure 6.40: Principle of Assemble-ability. Right images by Busch (2011).

Inputs (cause) – (1) On the level of the plate cell, assemble-ability is dictated by joint geometry (A.1.2) and plate arrangement (A.2.3); (2) on the global level by the arrangement of the plate cells (A.2.1).

Mechanism (function) – (1) Assembly of a plate cell requires joining plates along their finger-jointed edges. The assembly logic takes into account the geometrical constraints during the insertion of new elements into the plate cell. Elasticity of the plates practically allows joining plates to more than one edge at a time even though, theoretically, finger joints on different edges require different insertion vectors (Fig. 6.40, center). (2) Assembly of the structure is based on a sequence of construction that minimizes form work and support structures, and on allowing for manual accessibility to important connection details during assembly [174, p. 163] (Fig. 6.40, right).

Outputs (effect) – Assemble-ability: the ability to assemble cells from plates and the shell from plate cells by incorporating assembly constraints. In the case of (2) the assembly sequence.

Significance – The principle is a precondition for Cell Pre-assembly (A.6.3) and On-site Assembly (A.6.4) principles.

A.6.3 Principle of Pre-Assembly

The Principle of Pre-Assembly associates pre-assembled and finished plate cells with robotically fabricated finger-jointed plates via a manual assembly and finishing process in the context of the workshop (Fig. 6.41).

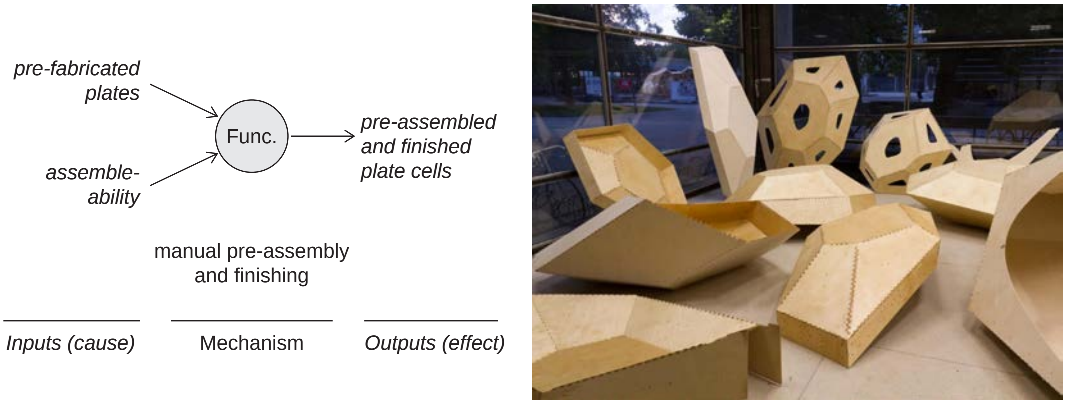


Figure 6.41: Principle of Pre-Assembly. Right image by Busch (2011).

Inputs (cause) – Robotically pre-fabricated 6.5 mm plywood plates as described in Robotic Pre-Fabrication (A.6.1); assemble-ability of plates as described in Assemble-ability (A.6.2).

Mechanism (function) – In a first step, flexible PU adhesive is applied at the joints to prevent disassembly of plates. In a second step, plate are sequentially joined along their edges through interlocking of the fingers. In the third step, plate cells are sanded along their finger-jointed edges for aesthetic reasons. In the last step, plate cells are finished by coating and varnishing to prevent fungal growth and for weather-proofing.

Outputs (effect) – Pre-assembled and finished plate cells.

Significance – The principle is a precondition for the principle of On-site Assembly (A.6.4). Pre-assembly of the cells allowed sequencing of pavilion assembly such that the cells in the lower ring would be assembled first in order to allow continuous construction and to minimize storage requirements.

On the one hand, the complexity of the finger-joint connection required a manual pre-assembly process. On the other had, assembly in the workshop could occur under controlled conditions and on-site labor was minimized.

A.6.4 Principle of On-site Assembly

The Principle of On-site Assembly associates the fully assembled segmented shell with pre-assembled plate cells via a defined assembly sequence and bolted connections between the cells (Fig. 6.42).

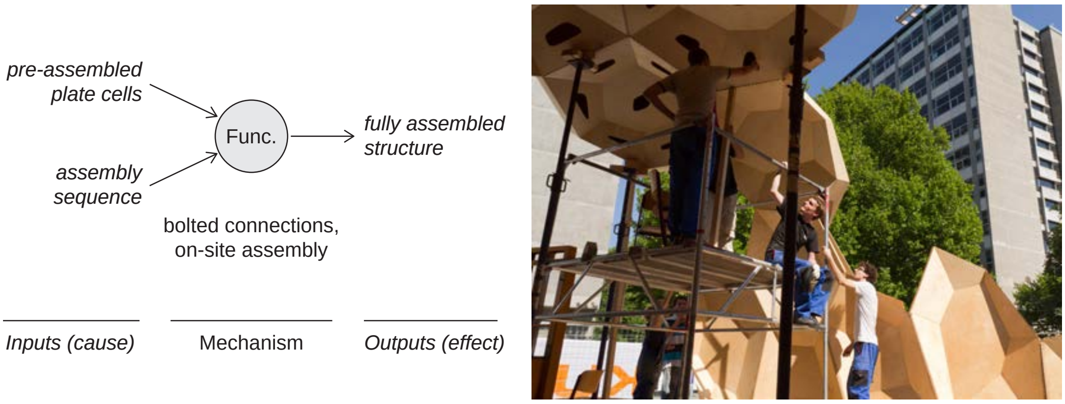


Figure 6.42: Principle of On-site Assembly. Right image by Busch (2011).

Inputs (cause) – Pre-assembled cells as the output of the Principle of Pre-assembly (A.6.3); a given assembly sequence minimizing support structures as defined by Assemble-ability (A.6.2).

Mechanism (function) – Plate cells were manually assembled on-site using a mobile platform. Due to the self-stabilizing effect of segments that are joined to at least two existing segment edges, only minimal temporary support was required to support the partially completed shell during assembly (Fig. 6.42, right). Openings in the inner layer of the double-layer plate cell allowed accessing the screw connections between plate cells and facilitated assembly and disassembly of the whole structure [171, p. 60].

Outputs (effect) – The fully assembled segmented shell: on-site work took exactly four weeks [5] including the preparation of the base. Total pre-fabrication and construction time was about six weeks.

Significance – Effectively, only minimal temporary support was necessary during construction. The structure could be easily assembled and disassembled as demonstrated on November 30, 2011, when the whole structure was disassembled by Holzbau Ochs GmbH within one working day and trucked away for storage.

---

[5] According to email records, on-site assembly started on July 28, 2011 and lasted until August 25.

6.4 Second-level Principle Chains of Case A

On the second level, the individual low-level principles are connected to form principle chains, also called inference chains, where the output of one principle forms the input of the following principle. On this level, principle chains illustrate the causal dependency between input factors and results of the development domains. The notation of principle chains thereby uses a graph-based notation for describing parametric relationships as demonstrated by the TensorFlow machine-learning library [199, p. 44]. In literature, a similar notation is used in block diagrams in Cybernetics, in Neural Network diagrams, and in process modeling, such as Business Process Modeling and Notation (BPMN). An early example for the use of this type of notation in the architectural context is Chris Abel's notation of system behavior in "Urban Chaos and Self-Organization" [1, p. 501].

6.4.1 Joint Development

Joint development associates bending-free edges and aesthetic qualities of the connection with geometrical (finger joint logic, edge-wise connection, trivalent arrangement) and fabricational inputs (subtractive fabrication, machine setup) via the specific sequence of low-level principles (Fig. 6.43).

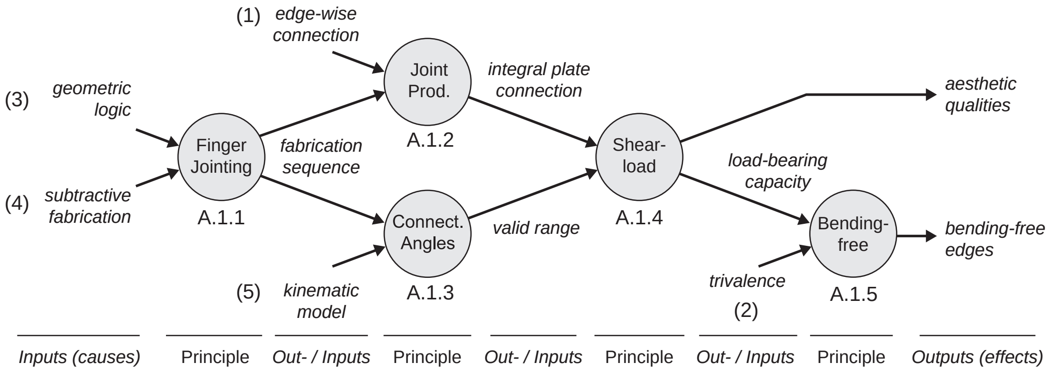


Figure 6.43: Chain of Joint Development Principles.

6.4.1.1 Inputs

Joint development was driven by objective [i] to extend the range of application of finger-jointing, while maintaining its inherent advantages (see also 5.5.4). Inputs for this domain thus were geometric and fabricational: in terms of geometry, plates had to be connected (1) in an edge-wise way and (2) in a trivalent plate arrangement. An edge-wise connection through integrated joints also implicated (3) a specific geometric relation between joint geometry and plate connection angle. Finally, fabrication of finger joints was predicated on (4) the inherent logic of a subtractive fabrication process and (5) a specific robotic fabrication setup.

6.4.1.2 Principle chain

Discrete development steps, which have been identified and defined as low-level functional principles in A.1, are used to associate the results of joint development with the geometric and fabricational inputs mentioned above by connecting their inputs and outputs:

- (A.1.1) the Principle of Finger Jointing associates a fabrication sequence for finger joints with the geometric logic of integrated plate joints and the fabrication logic of subtractive fabrication;
- (A.1.2) in the Principle of Finger Joint Production, the production of integral plate connections is associated with the requirement for an edge-wise connection and the given fabrication sequence via CNC-fabrication; in parallel,
- (A.1.3) the Principle of Connecting Angles defines the valid range of connection angles between plates as a function of fabrication sequence and specific robotic fabrication setup; consecutively,
- (A.1.4) the Principle of Shear-load Transfer defines the load-bearing capacity of the connection and its aesthetic qualities as a function of the integral geometric features of the joint and the connection angles; and finally
- (A.1.5) the Principle of Bending-free Edges states that bending-free edges in a bending-bearing plate structure depend on a shear load-bearing plate connection and a trivalent plate layout.

Figure 6.44: Close-up of joint detail in the finished pavilion. From [100] by Menges (2011).

6.4.1.3 Result

Output of the domain is a hinged, shear-resisting, and edge-wise connection for plates that exhibits intrinsic aesthetic qualities. In a trivalent arrangement of plates it can achieve a bending bearing structure while being composed of bending-free edges (Fig. 6.44).

6.4.1.4 Significance

Joint development meets the objective related to the expanding solution space in that it extends the range of application of finger-jointing and at the same time maintains its structural and aesthetic advantages. While being informed by fabrication process and setup, the results of Joint Development subsequently provide necessary inputs for the ensuing development process.

6.4.2 Cell Development

Cell development associates an adaptive model of polyhedral plate cells and bending-free edges on two levels of hierarchy with geometrical, structural, fabricational, and logistical inputs via the specific sequence of low-level functional principles (Fig. 6.45).

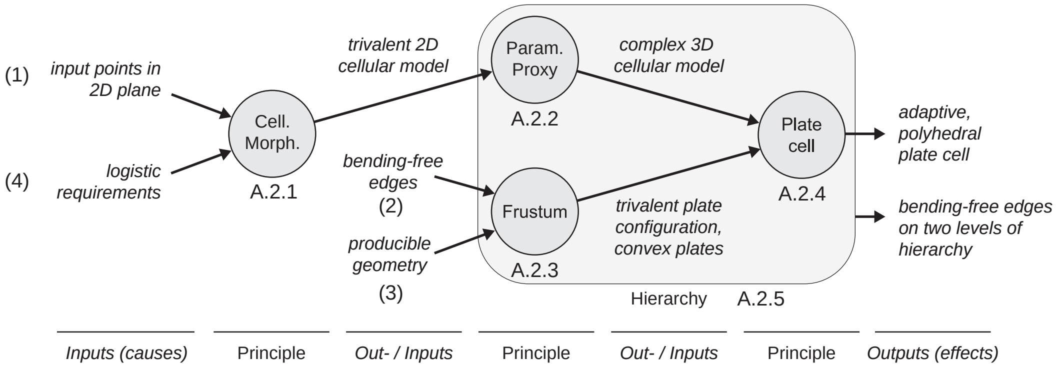


Figure 6.45: Chain of Cell Development Principles.

6.4.2.1 Inputs

The development of plate cells was driven by the objectives [vi] to develop a lightweight material system on the basis of the introduced robotic fabrication process for integrated timber joints; and [viii] to develop a modular system of shell segments that allows for a high degree of adaptability and performance (enabled by the geometric differentiation and configuration of its plate elements).

Inputs for this domain thus relate to geometry, structure, fabrication, and handling: (1) in terms of geometry, a set of user-defined input points defines number, location, size, and shape of plate cells; (2) structural inputs are the bending-free edges based on shear-load capacity and trivalence; (3) producibility as defined by the range of producible geometry; and (4) logistic requirements are defined by transportability and manageable element sizes for assembly;

6.4.2.2 Principle chain

Discrete development steps, which have been identified and defined as low-level functional principles in A.2, are used to relate the results of cell development to the multi-domain inputs mentioned above by connecting the inputs and outputs of individual principles. In the first step of cell development,

- (A.2.1) the Principle of Cellular Morphology associates a 2-dimensional subdivision of the plane and a corresponding triangular mesh with user defined input points and logistic requirements;
- (A.2.2) the Principle of Parametric Proxy associates a trivalent 3-dimensional cellular model with the previously mentioned 2-dimensional map; in parallel,
- (A.2.3) the Principle of Frustum relates a trivalent plate configuration within the plate cell to (2) the bending-free edges principle and (3) the range of producible geometry; consecutively,
- (A.2.4) the Principle of Modular Plate Cell associates an adaptive, geometrically differentiated, polyhedral plate cell with a trivalent 3-dimensional cellular model and the trivalent plate configuration; and finally
- (A.2.5) the Principle of Hierarchy defines a two-level hierarchy of bending-free edges as a function of the trivalent configuration of the plates within the cell as well as of the arrangement of cells within the segmented shell.

6.4.2.3 Result

The results of cell development were a geometrically adaptive model of a polyhedral plate cell providing a high degree of geometric differentiation and a hierarchical material system for segmented shells characterized by bending-free edges both on the level of the plates as well as on the level of the shell segments (Fig. 6.46).

6.4.2.4 Significance

Cell development meets the objectives related to the exploration of the solution space in that it results in a lightweight material system (based on the introduced robotic fabrication process for integrated timber joints) that allows for a high degree of adaptability while maintaining structural performance and producibility. Furthermore, the development of the modular plate cell and its arrangement in the global structure is an example for successfully combining biomimetic principles (e.g., three plates meeting in one point) with fabricational constraints (e.g., connecting angles that lie in the producible range).

Figure 6.46: Geometric differentiation of segment shapes (Busch, 2011).

Regarding the overall development process, the capacity of the cell to adapt to varying boundary conditions is the precondition for accommodating geometric differentiation as defined by the Form-Defining Principles in Design Development (6.4.3).

6.4.3 Design Development

Design Development relates morphological adaptation of the segmented shell to programmatic requirements, structural analysis, and control parameters via the specific sequence of low-level functional principles (Fig. 6.47).

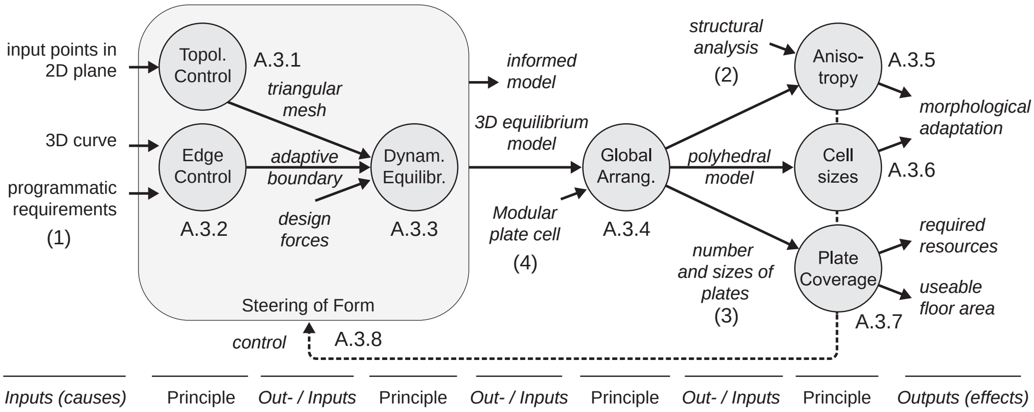


Figure 6.47: Chain of Form-defining principles.

6.4.3.1 Inputs

Starting from the objective [ii] to develop computational design methods that activate the solution space provided by 7-axis robotic fabrication, inputs for this category are control parameters that take architectural, programmatic, structural, and logistical requirements into account.

Types of parameters are geometrical in case of (1) architectural and programmatic inputs, and numerical in case of (2) structural analysis and (3) logistic aspects, such as the number and sizes of plate cells in relation to the available resources. Additionally, (4) the adaptive shell segment as the outcome of Cell Development forms an input for Design Development.

6.4.3.2 Principle Chain

Eight discrete development steps have been identified in the Design Development domain and defined as low-level functional principles in A.3. Their inputs and outputs can now be connected to form a principle chain that relates morphological adaptation of the segmented shell to the inputs mentioned above via the following sequence of functional principles:

- (A.3.1) the Principle of Topological Control defining a triangular mesh as a function of user-defined inputs points in the 2D plane;
- (A.3.2) the Principle of Interactive Edge Control, associating an adaptive boundary of the reference mesh with a 3-dimensional curve, and architectural and programmatic requirements;
- (A.3.3) the Principle of Dynamic Equilibrium associating a 3-dimensional triangular mesh in dynamic equilibrium with a planar, 2-dimensional triangular mesh and an adaptive system boundary;
- (A.3.4) the Principle of Global Cell Arrangement relating a full polyhedral model of a segmented shell to a geometrically differentiated 3-dimensional model in dynamic equilibrium and an adaptive modular plate cell;
- (A.3.5) the Principle of Anisotropy defining morphological adaptation to the main load-bearing direction and reduced stress concentrations in the structure as a function of structural analysis of the polyhedral model;
- (A.3.6) the Principle of Variable Cell Sizes defining morphological adaptation of the plate structure as a function of geometric relations within the plate model, such as the distance of a plate cell to the boundary of the system;
- (A.3.7) the Principle of Cell Coverage associating required resources, such as time and material, and programmatic ‘fitness’, such as usable floor area, with the number and sizes of building elements; and finally
- (A.3.8) the higher-level Principle of Steering of Form defining an ‘informed’, geometrically differentiated polyhedral model as the result of programmatic,

structural, fabricational (via modular plate cell) and logistical steering parameters via feedback loops.

6.4.3.3 Result

The result of design development was a global cell arrangement that exhibits morphological adaptation of plate cells in terms of cells sizes and proportions (anisotropy) and that meets architectural, programmatic, and logistical requirements via the high-level principle of Steering of Form (A.3.8) (Fig. 6.48).

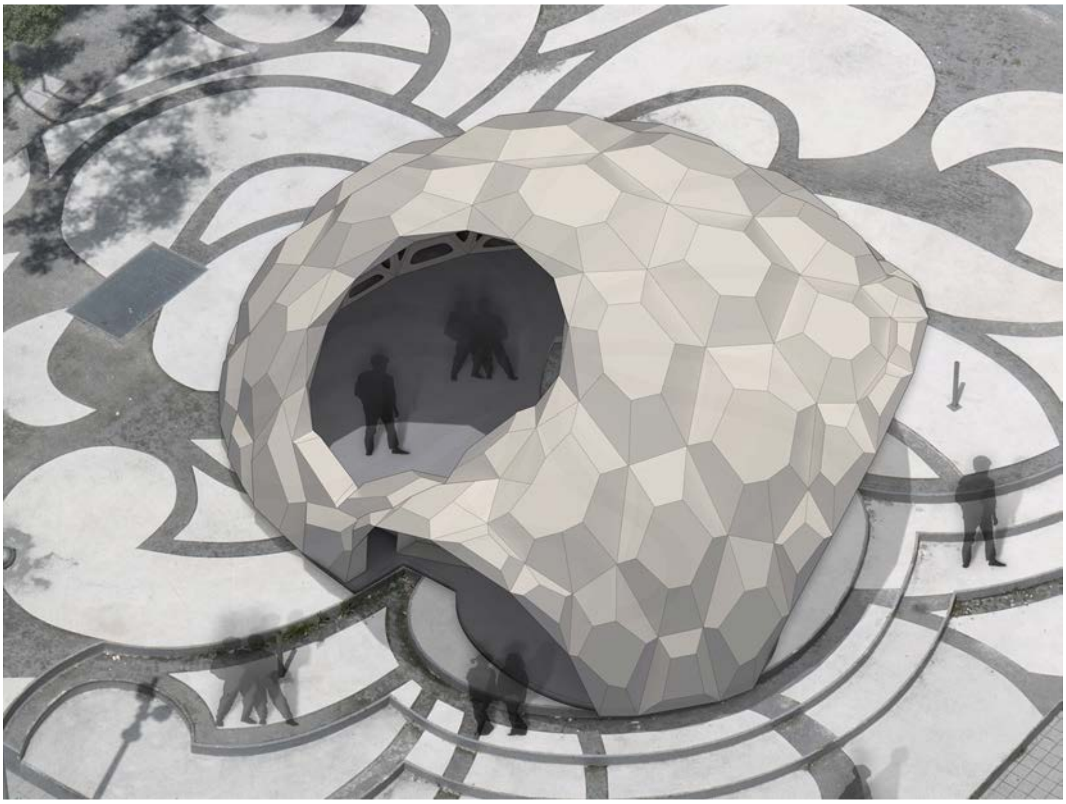


Figure 6.48: Design model showing morphological variation of plate cells in global segment arrangement (Krieg, 2011).

6.4.3.4 Significance

Design development thus meets the objective related to design computation in that it results in fabrication- and structure-informed morphological adaptation expressed by the principles A.3.5 and A.3.6, thereby demonstrating the geometric flexibility of the material system; and by A.3.7 as validated resource requirements and programmatic 'fitness' of the resultant structure.

The most significant aspect in design development are the feedback loops integrating the results of the principle chain into the formation process thereby breaking the linearity of the information flow. Scores can be assigned to individual aspects, such as stress in the structure, number of plate elements and resultant fabrication time, angles between plate, etc., which are fed back as steering parameters that steer the formation process of the pavilion. The Design Development domain thus constitutes the core of the development of the segmented shell where objectives and constraints are integrated through feedback loops. This applies to 'upstream' loops, such as in the case of joint and cell development, and to 'downstream' loops, as in the case of structural design, fabrication and construction development. During the development of the project, this feedback has been implemented in an interactive user-centered process.

6.4.4 Structural Development

Structural Development relates a globally bending bearing plate structure composed of hinged segments, increased stiffness, and a uniform stress distribution to the observation that segmented shells in biology exhibit a trivalent plate configuration, physical finger-joint connection samples, and a full polygonal plate model composed of convex plates via a specific sequence of functional principles (Fig. 6.49).

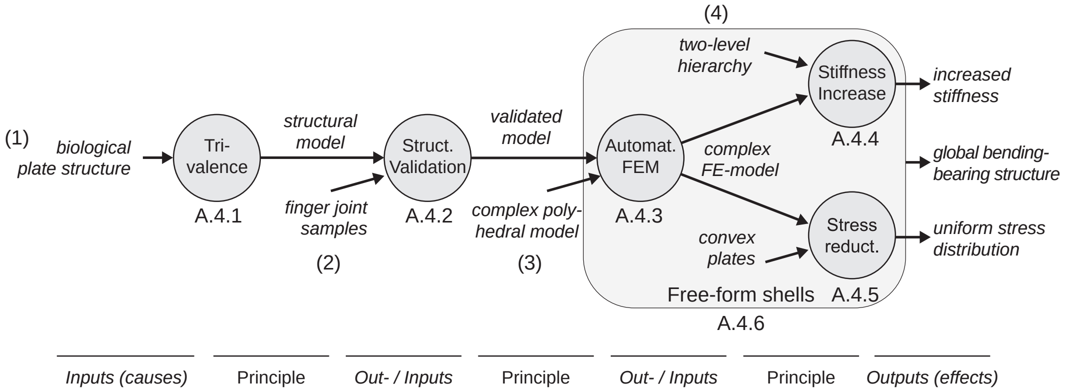


Figure 6.49: Chain of Structural Design principles.

6.4.4.1 Inputs

Structural Development was driven by the objectives [xii] to test the hypothesis of achieving bending-free structures by following the 3-plate-rule, [xiii] to develop a structural model of finger-joint connections, and [xiv] to build plate structures, which transfer only normal and shear forces between the edges. Starting points for structural development thus were (1) the observation of segmented shells in biology, (2) physical samples of finger joint connections, (3) a full polygonal plate model of a segmented shell consisting of polyhedral cells that are composed of (4) convex plates.

6.4.4.2 Principle Chain

Six discrete development steps have been identified in the Structural Development domain and defined as functional principles in A.4.

The inputs and outputs of the low-level functional principles can now be connected to form a principle chain that relates a globally bending bearing plate structure composed of hinged segments, increased stiffness, and a uniform stress distribution to the inputs mentioned above via the following sequence of functional principles:

- (A.4.1) the Principle of Trivalence, in which an abstract structural model of the finger joint connection relates to the observation that segmented shells in biology exhibit a trivalent plate configuration (input 1);
- (A.4.2) the Principle of Structural Validation, in which a validated structural model depends on joint samples (input 2) and the abstract structural model;
- (A.4.3) the Principle of Automated FEM, in which a differentiated FE-model is a function of a validated structural model and a geometrically differentiated polygonal plate model (input 3);
- (A.4.4) the Principle of Increased Stiffness defining increased stiffness as a function of computational FE-modeling and closed polyhedral cells that meet the trivalence criterion on two levels of hierarchy (input 4);
- (A.4.5) the Principle of Stress Reduction associating a more uniform stress distribution in the structure with convex plates and computational FE-modeling; and finally
- (A.4.6) the higher-level Principle of Free-form Segmented Shells associating a globally bending-bearing structure with a trivalent arrangement of convex plates in a hierarchical plate system, computational FE-modeling, and a non-load-optimized global shell form.

6.4.4.3 Result

The results of Structural Development were increased stiffness of the shell due to the two-level hierarchy, a more uniform stress distribution through convex plates, and a globally bending-bearing structure via the high-level principle of Free-form shells (A.4.6) (Fig. 6.50).

6.4.4.4 Significance

Structural development thus meets the objectives related to structural design in that this sequence of principles results in increased stiffness of the plate cells and correspondingly in a reduction of deflection of the segmented shell as expressed by principle A.4.4; in reduced stress concentrations in the structure due to convex plates expressed by A.4.5; and in a globally bending bearing structure where only membrane stresses occur in the plates (plate action) expressed by A.4.6.

This last result constitutes the most significant aspect in structural development where, unlike conventional lightweight construction, this high-level principle can be applied to freeform, i.e., non-load-optimized segmented shells while still produ-

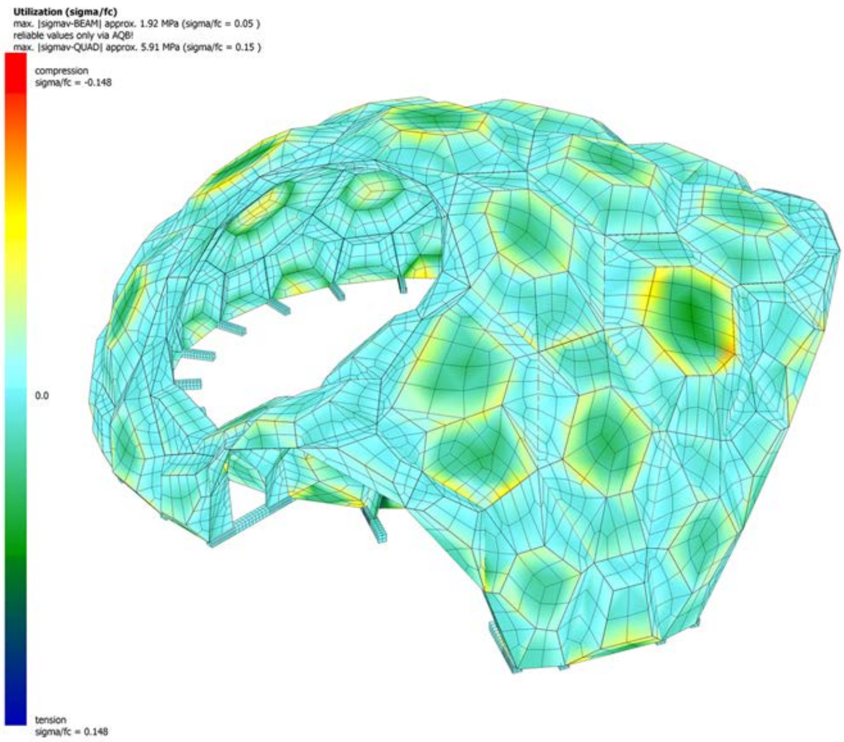


Figure 6.50: Structural model showing utilization of the plates in the plate structure. From [108] by ICD/ITKE University of Stuttgart (2011).

cing membrane stresses in the structure. In addition to new possibilities in design computation and digital fabrication, this finding significantly expands the possible solution space for segmented plate structures through new structural possibilities.

6.4.5 Fabrication Development

Fabrication Development relates a validated fabrication process, including machine-specific control code, and the delineation of the machinic morphospace to a finger-joint fabrication strategy, a given machine setup and a polyhedral plate model via a specific sequence of low-level functional principles (Fig. 6.51).

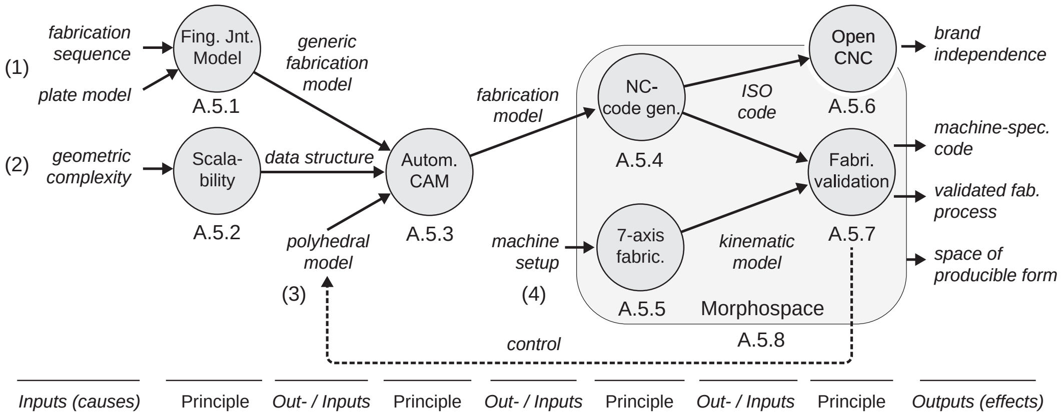


Figure 6.51: Chain of Fabrication Design principles.

6.4.5.1 Inputs

Given objectives [iii] to develop integrative computational design and fabrication models, and [v] to integrate tool path and machine code generation into the design model, inputs for this category are (1) the sequence of fabrication steps as defined in the principle of Finger Jointing (A.1.1); (2) the given geometric complexity of an integrated timber joint such as finger joints; (3) a geometrically differentiated, polyhedral plate model as the result of the principle of Global Cell Arrangement (A.3.4); and (4) a specific machine setup.

6.4.5.2 Principle Chain

Eight discrete development steps have been identified in the Fabrication Development domain and defined as functional principles in A.5. The inputs and outputs of the low-level functional principles can be connected to form a principle chain that relates a validated fabrication process, including machine-specific control code, and the delineation of the machinic morphospace to the inputs mentioned above via the following sequence of low-level functional principles:

- (A.5.1) the Principle of Finger Joint Fabrication Modeling, in which a mathematical model of finger-joint fabrication represents a given fabrication sequence;
- (A.5.2) the Principle of Scalability defining a scalable data structure for fabrication modeling based on the geometric complexity of finger joints;
- (A.5.3) the Principle of Automated Fabrication Modeling defining a complete digital fabrication model containing all tool paths and fabrication parameters as a function of the abstract model of finger joint fabrication and a scalable data structure for fabrication modeling;
- (A.5.4) the Principle of NC-Code Generation, in which a complete digital fabrication model is the prerequisite for ISO-standard (ISO 6983) machine control code;
- (A.5.5) the Principle of 7-axis Robotic Fabrication defining a kinematic model of robotic fabrication as a function of a specific machine setup;
- (A.5.6) the Principle of Open-CNC associating brand-independent machine control code with ISO-compliance in machine code generation;
- (A.5.7) the Principle of Fabrication Validation, in which a validated fabrication process and machine-specific control code depend on a kinematic model of the robotic fabrication process and ISO-standard control code; and finally,
- (A.5.8) the higher-level Principle of Machinic Morphospace defining the delineation of the fabricational solution space of plates in a segmented shell as the result of a given machine setup, fabrication constraints, such as connecting angles, and a validation mechanism for fabrication.

6.4.5.3 Result

The results of Fabrication Development were a validated fabrication process through kinematic simulation including machine code, independence from a specific equipment manufacturer due to ISO-compliance of machine code, and a conceptualization of the space of producible form via the high-level principle of Machinic Morphospace (A.5.8) (Fig. 6.52).

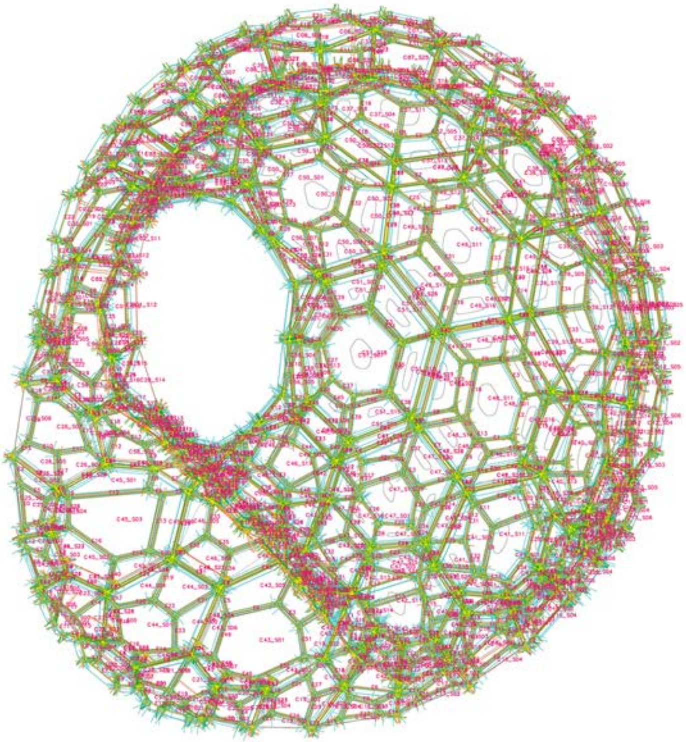


Figure 6.52: Fabrication data model showing tool paths and plate IDs necessary for fabrication (Schwinn, 2011).

6.4.5.4 Significance

Fabrication development thus meets the objectives related to design integration and custom fabrication routines in that this sequence of principles results in machine-specific control code and a validated fabrication process expressed by the principle A.5.7; and in the delineation of the geometric solution space for the fabrication of plates expressed by A.5.8.

Two aspects in this sequence are particularly significant. First, the combination of the principles of Automated Fabrication Modeling (A.5.3) and the automation of machine code programming as described in the principle of NC-Code Generation (A.5.4) are a prerequisite for the efficient fabrication of highly differentiated structures (see [15, p. 121]). Second, the principle of Machinic Morphospace (A.5.8) is a general principle that can be applied in other contexts as it is the prerequisite of fabrication-oriented design integration.

6.4.6 Production Domain

The Production Domain relates a fully assembled structure, which requires only minimal falsework during assembly, to a validated 7-axis robotic fabrication process, finger joint geometry, and specific plate and cell arrangement principles via a specific sequence of low-level functional principles (Fig. 6.53).

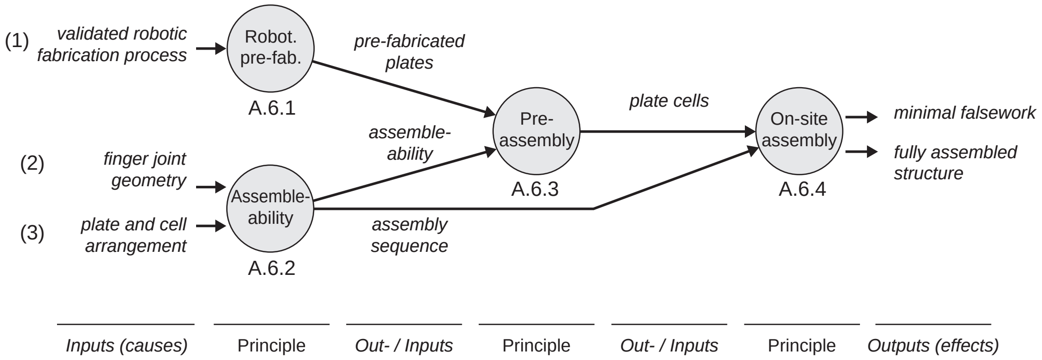


Figure 6.53: Chain of Production principles.

6.4.6.1 Inputs

Given objective [ix] to transfer the developed material system into a pavilion with specific architectural requirements, inputs for this domain are (1) the validated robotic fabrication process as defined in the principle of Fabrication Validation (A.5.7) including machine-specific code; (2) the specific plate arrangement constraints, which are based on joint geometry, and (3) cell arrangement constraints, which are based on cell geometry, that together determine assemble-ability.

6.4.6.2 Principle Chain

Four discrete development steps have been identified in the Production domain and defined as functional principles in A.6. The inputs and outputs of functional principles can be connected to form a principle chain that relates a fully assembled structure to the inputs mentioned above via the following sequence:

- (A.6.1) the Principle of Robotic Pre-fabrication associating pre-fabricated finger-jointed plates with a validated robotic fabrication process via a specific 7-axis robot setup;
- (A.6.2) the Principle of Assemble-ability defining assemble-ability and assembly sequence of the segmented shell as a function of the specific joint and plate geometry, and plate and cell arrangement;
- (A.6.3) the Principle of Pre-Assembly relating pre-assembled and finished plate cells to robotically pre-fabricated plates and assemble-ability; and finally

- (A.6.4) the Principle of On-site Assembly where the fully assembled structure is contingent upon pre-assembled plate cells and a given assembly sequence.

6.4.6.3 Result

The output of the Production domain was the fully assembled structure requiring only minimal falsework in the assembly process (Fig. 6.54).

Figure 6.54: Aerial view of the finished pavilion. From [100] by Busch (2011).

6.4.6.4 Significance

Production meets the objectives related to the transfer of the developed material system into a pavilion in that this principle chain results in the finished, fully assembled structure expressed by the principle A.6.4.

Two aspects of the result are particularly important: first, only minimal temporary support was necessary during construction which allowed minimizing on-site labor and construction time; second, due to the modular nature of the segmented shell, the temporary bolted connections between the plate cells, and easily accessible connection details, the segmented shell could not only be easily assembled and disassembled, as demonstrated in November 2011, but also reassembled: in the summer of 2018, the shell was reassembled by the timber construction firm Ochs GmbH, one of the project supporters at the time.

6.5 Third-level Abstraction

On the third level, relationships and interfaces between the second-level principle chains (that are composed of first-level principles) come into focus. Using this approach of increasing abstraction, the *vertical* relationships of low-level principles to high-level domains and the *horizontal* relationships / interfaces between domains in the design, development, and realization process can be described.

The analysis of functional principles and their associations in the design and development of the ICD/ITKE Research Pavilion 2011 reveals a complex network of interactions with multiple interfaces between different development domains and cross-domain feedback loops. This section thus first summarizes the results of the analysis and then provides a further description of the interrelations across domains.

6.5.1 Summary of Principles

Table (Tab. 6.1) provides a summary of the principles that have been identified and categorized according to development domain as described above and gives an overview on the interfaces between principles within and across the respective domains. System-external global inputs and outputs, such as machine setup or the effect of on-site assembly, are marked as ‘ex’ unless otherwise noted.

Table 6.1: Summary table of functional principles in Case Study 1.

No.	Principle	Inputs from	Outputs to
A.1	Joint development principles		
A.1.1	Finger Jointing	ex	A.1.2/3, A.5.1
A.1.2	Finger Joint Production	A.1.1	A.1.4, A.4.2, A.5.2, A.6.2
A.1.3	Connecting Angles	A.1.1, A.5.5	A.1.4, A.5.8
A.1.4	Shear-load Transfer	A.1.2/3	A.1.5
A.1.5	Bending-free Edges	A.1.4, A.4.1	A.2.1, A.2.3
A.2	Cell development principles		
A.2.1	Cellular Morphology	A.1.5	A.2.2, A.6.2
A.2.2	Parametric Proxy	A.2.1, A.3.3	A.2.4+5
A.2.3	Frustum	A.1.5, A.4.5, A.5.8	A.2.4+5, A.6.2
A.2.4	Modular Plate Cell	A.2.2, A.2.3	A.2.5, A.3.4, A.5.1
A.2.5	Hierarchy	A.2.2-4	A.4.4
A.3	Design development principles		
A.3.1	Topological Control	A.3.8	A.3.3
A.3.2	Interactive Edge Control	A.3.8	A.3.3

Continued on next page

No.	Principle	Inputs from	Outputs to
A.3.3	Dynamic Equilibrium	A.3.1+2, A.3.8	A.2.2, A.3.4
A.3.4	Global Cell Arrangement	A.2.4, A.3.3	A.3.5-7, A.4.3, A.5.3
A.3.5	Anisotropy	A.3.4, A.4.3	A.3.8
A.3.6	Variable Cell Size	A.3.4	A.3.8
A.3.7	Plate Coverage	A.3.4	A.3.8
A.3.8	Steering of Form	A.3.5-7, A.5.7	A.3.1-3
A.4	Structural development principles		
A.4.1	Trivalence	ex	A.1.5, A.4.2
A.4.2	Structural Validation	A.1.2, A.4.1	A.4.3
A.4.3	Automated FEM	A.3.4, A.4.2	A.3.5, A.4.4-6
A.4.4	Increased Stiffness	A.2.5, A.4.3	A.4.6
A.4.5	Stress Reduction	A.4.3	A.2.3, A.4.6
A.4.6	Free-form Segmented Shells	A.3.4, A.4.3-5	ex
A.5	Fabrication development principles		
A.5.1	Finger Joint Modeling	A.1.1 , A.2.4	A.5.3
A.5.2	Scalability	A.1.2	A.5.3
A.5.3	Autom. Fabrication Modeling	A.3.4, A.5.1+2	A.5.4
A.5.4	NC-Code Generation	A.5.3	A.5.6+7
A.5.5	7-axis Robotic Fabrication	machine setup	A.1.3, A.5.7+8
A.5.6	Open CNC	A.5.4	ex
A.5.7	Fabrication Validation	A.5.4+5	A.3.8, A.5.8, A.6.1
A.5.8	Machinic Morphospace	A.1.3, A.5.5+7	A.2.3, ex
A.6	Production principles		
A.6.1	Robotic Pre-Fabrication	A.5.7	A.6.3
A.6.2	Assemble-ability	A.1.2, A.2.1/3	A.6.3+4
A.6.3	Pre-Assembly	A.6.1+2	A.6.4
A.6.4	On-site Assembly	A.6.2+3	ex

The corresponding flow of information, including global inputs and outputs, is shown as a directed graph in Fig. 6.55.

The visualization reveals that some of the domains that in conventional architectural planning are considered ‘down-stream’ processes, such as Fabrication development (A.5), have a significant influence on the ‘upstream’ design process. In the following, this influence is termed *integration* such that domains are said to be ‘integrating’ domain-external aspects, if they are influenced by them.

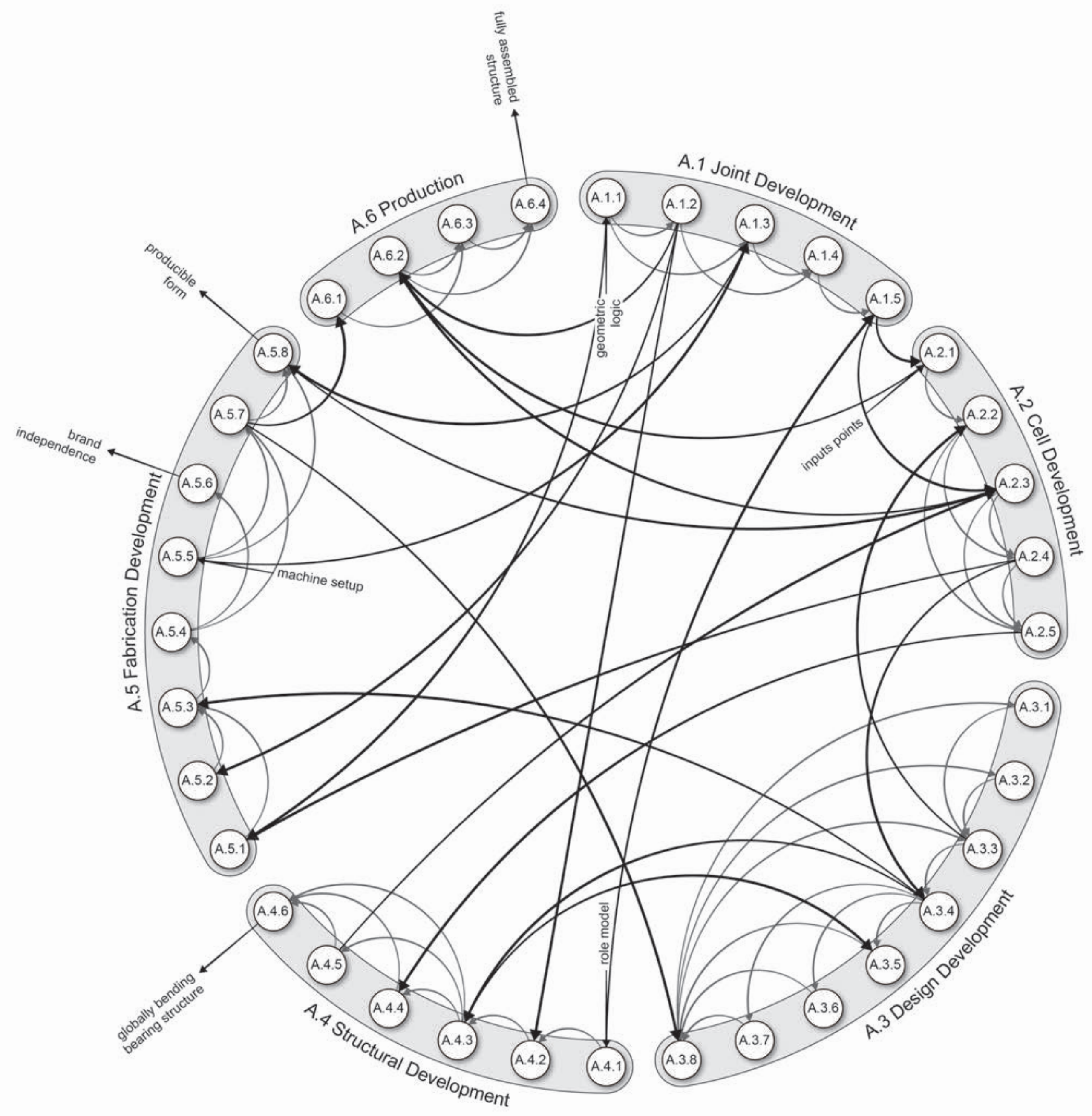


Figure 6.55: Interfaces and cross-domain feedback loops in Case A.

6.5.2 Cross-domain integration

For instance, the **Joint Development** domain (A.1) integrates aspects of structural development and fabrication (A.4 and A.5) and itself influences cell development, structure and fabrication development, and production. Joint development thus is in a bi-directional, cross-domain relationship with structural and fabricational development thereby confirming empirical knowledge that was acquired during the realization of the research pavilion. The domain has a total of nine cross-domain connections.

Specifically, the principle of Connecting Angles (A.1.3) integrates 7-Axis Robotic Fabrication (A.5.5) and its kinematic model; additionally, the principle of Bending-free Edges (A.1.5) integrates the principle of Trivalence (A.4.1), which is a precondition for the development of shell segments (A.2).

Similarly, the **Cell Development** domain (A.2) integrates aspects of joint development, design, structural, and fabrication development, while also influencing design, structure, and fabrication development, as well as production, thereby forming three bi-directional, cross-domain relationships with design, structure, and fabrication development. The domain has the highest total number of ten cross-domain connections.

Specifically, the principle of Cellular Morphology (A.2.1) integrates the three-plate rule of the principle of Bending-free Edges (A.1.5). The principle of Parametric Proxy (A.2.2) integrates the principle of Dynamic Equilibrium (A.3.3). The principle of Frustum (A.2.3) integrates fabricational aspects through the principle of Machinic Morphospace (A.5.8), which defines the limits of the space of producible form, in this case plate sizes, polygon angles, and connection angles, as well as the preference for convex plates from the principle of Stress Reduction (A.4.5).

The **Design Development** (A.3) domain is at the centre of three bi-directional, cross-domain relationships integrating and influencing aspects of cell development, structural development, and fabrication development. The domain has a total of six cross-domain connections.

Specifically, the principle of Global Cell Arrangement (A.3.4) integrates the principle of Modular Plate Cell (A.2.4) from the cell development domain (A.2). The principle of Anisotropy (A.3.5) integrates structural aspects of FE-analysis (A.4.3). Finally, the principle of Steering of Form (A.3.8) integrates Fabrication Validation (A.5.7).

Given the proposed categorization of principles and causality (principle order) within the development domains, all principles with a higher order number than the Principle of Global Cell Arrangement (A.3.4) affect Design Development as part of a feedback loop. For example, (A.3.4) is controlled by a range of parameters ranging from logistic (A.3.5 Plate Coverage) and geometric (A.3.6 Anisotropy, A.3.7 Variable Cell Size), to structural (A.4.5 Stress Reduction) and fabricational parameters (A.5.7 Fabrication Validation) through A.5.8 Steering of Form resulting in morphological adaptation of the shell structure.

The domain of **Structural Development** (A.4) is in a bi-directional, cross-domain relationship with joint, cell, and design development (A.1-3). The domain has a total of six cross-domain connections.

Specifically, the principle of Structural Validation (A.4.2) integrates Finger Joint Production (A.1.2) from the joint development domain (A.1): only when there is a process for producing finger jointed plate samples, physical load tests can be performed. The principle of Automated FEM (A.4.3) integrates Global Cell Arrangement (A.3.4) from the design development domain (A.3). Finally, the principle of Increased Stiffness (A.4.4) integrates the principle of Hierarchy (A.2.5) from the cell development domain (A.2).

The **Fabrication development** domain (A.5) integrates aspects of joint, cell, and design development (A.1-3) with which it is engaged in bi-directional, cross-domain relationships. Additionally, it predicates the construction of the pavilion (A.6). The domain has a total of nine cross-domain connections.

Specifically, Finger Joint Modeling (A.5.1) integrates the principle of Finger Jointing (A.1.1), which defines a sequence of fabrication steps for finger joints, and the model of a plate cell (A.2.4). Scalability (A.5.2) integrates the geometric complexity of Finger Joint Production (A.1.2). Automated Fabrication Modeling (A.5.3) integrates Global Cell Arrangement (A.3.8). Finally, Machinic Morphospace (A.5.8) integrates the range of valid connection angles (A.1.3) from the joint development domain (A.1).

Lastly, the **Production domain** (A.6) integrates aspects of Joint and Cell Development (A.1 and A.2) and Fabrication Development (A.5), but does not actively influence any other domain. The production domain has a total number of four cross-domain connections. With the fully assembled structure being the outcome of the production domain, the domain generates the principal system-external output.

Specifically, Robotic Fabrication (A.6.1) integrates a validated fabrication process and data model as is the outcome of the principle of Fabrication Validation (A.5.7). The principle of Assemble-ability (A.6.2) integrates joint geometry and the arrangement of cells and plates as is the outcome of the principles of Finger Joint Production (A.1.2), Cellular Morphology (A.2.1), and Frustum (A.2.3). Assemble-ability, in this case, is the outcome of the previous principles, not an explicit design driver. As a design principle, assemble-ability would typically influence both the development of the connection details (A.1) and building elements (A.2) in the design process. As stated in A.6.2, while the specific finger joint geometry theoretically makes it impossible to assemble three plates along two edges, it is through the elasticity of the material (6.5 mm Birch plywood) that pre-assembly of plate cells was practically possible.

6.5.3 Cross-domain Feedback Loops and Concurrent Developments

The analysis of integration between domains shows that not all bi-directional relationships between domains affect their own input as would be the definition of 'feedback' (see 2.1). Therefore, a further distinction needs to be made regarding the quality of those relationships.

In this context a 'feedback loop' is defined as a bi-directional relationship between two domains A and B, where an output O_A of domain A affects an input of domain B at time t; the corresponding output O_B then affects an input of domain A at time $t + 1$ such that it changes the original output O_A in what effectively forms a loop in the flow of information.

Alternatively, developments in different domains can occur concurrently where, rather than the outputs of domain A effectively changing its own inputs, the outputs O_B of domain B feed back into domain A at a point where they do not affect the corresponding inputs of domain B. This constellation is termed 'concurrent relationship'. In the following, the seven bi-directional cross-domain relationships that have been identified above are described in detail and illustrated.

The **first concurrent relationship** exists between the domains of joint development (A.1) and structural development (A.4): the principle of Structural validation (A.4.2) integrates the principle of Finger-joint production (A.1.2), while the principle of Bending-free edges (A.1.5) integrates the principle of Trivalence (A.4.1) without affecting A.1.2 (Fig. 6.56).

A **second concurrent relationship** exists between the domains of joint development (A.1) and fabrication development (A.5): the principles of Finger Joint Modeling (A.5.1) and Scalability (A.5.2) integrate Finger Jointing (A.1.1) and Finger Joint Production (A.1.2), respectively; while at the same time the principle of Connecting Angles (A.1.3) integrates 7-axis robotic fabrication (A.5.5) without affecting A.1.1 or A.1.2 (Fig. 6.57).

A **first feedback loop** exists between the domains of cell development (A.2) and design development (A.3): the principle of Global Cell Arrangement (A.3.4) integrates the principle of Modular Plate Cell (A.2.4), while the principle of Parametric Proxy (A.2.2) integrates Dynamic Equilibrium (A.3.3) via Steering of Form (A.3.8) thereby affecting A.2.4 (Fig. 6.58).

A **third concurrent relationship** exists between the domains of cell development (A.2) and structural development (A.4): the principle of Frustum (A.2.3) integrates

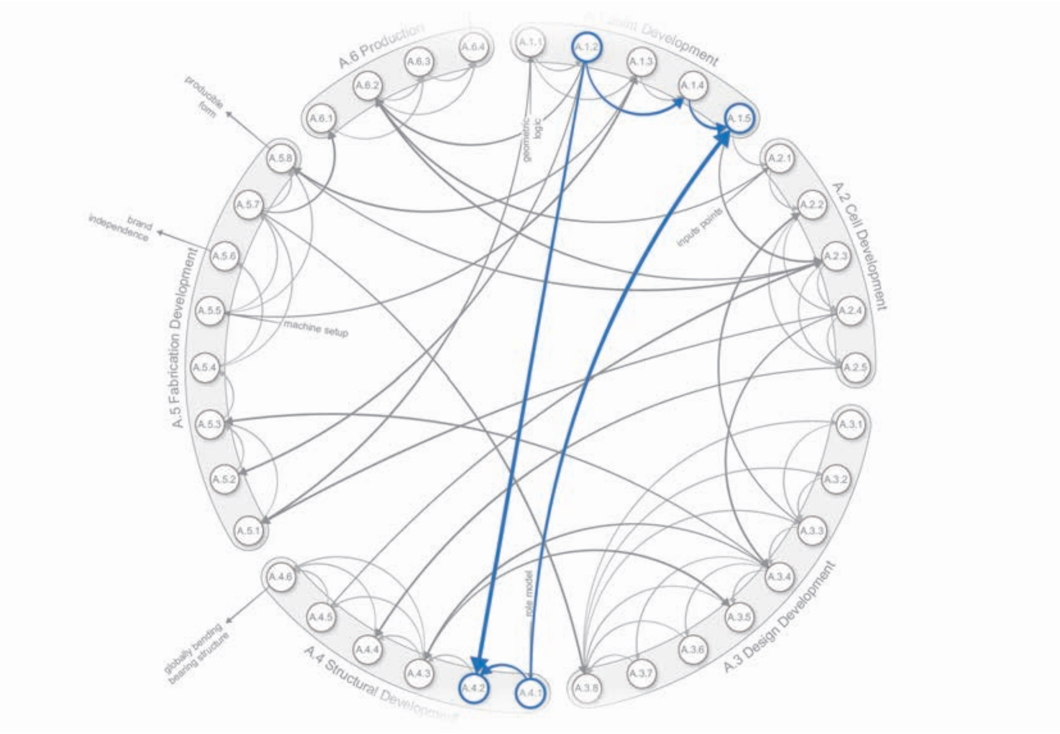


Figure 6.56: Concurrent relationship between joint development and structural joint development domains.

Figure 6.57: Concurrent relationship between joint development and fabrication development domains.

the principle of Stress Reduction (A.4.5), while the principle of Increased Stiffness (A.4.4) integrates the principle of Hierarchy (A.2.5) without affecting A.4.5 (Fig. 6.59).

The **second feedback loop** exists between the domains of cell development (A.2) and fabrication development (A.5): the principle of Finger Joint Modeling (A.5.1)

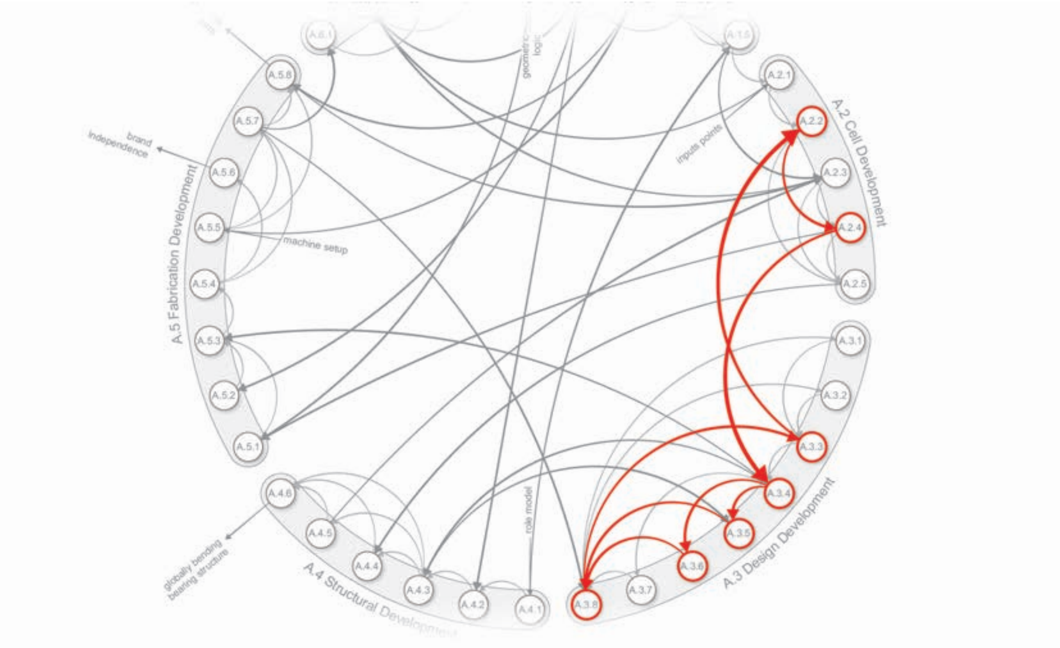


Figure 6.58: Concurrent relationship between cell- and design development domains.

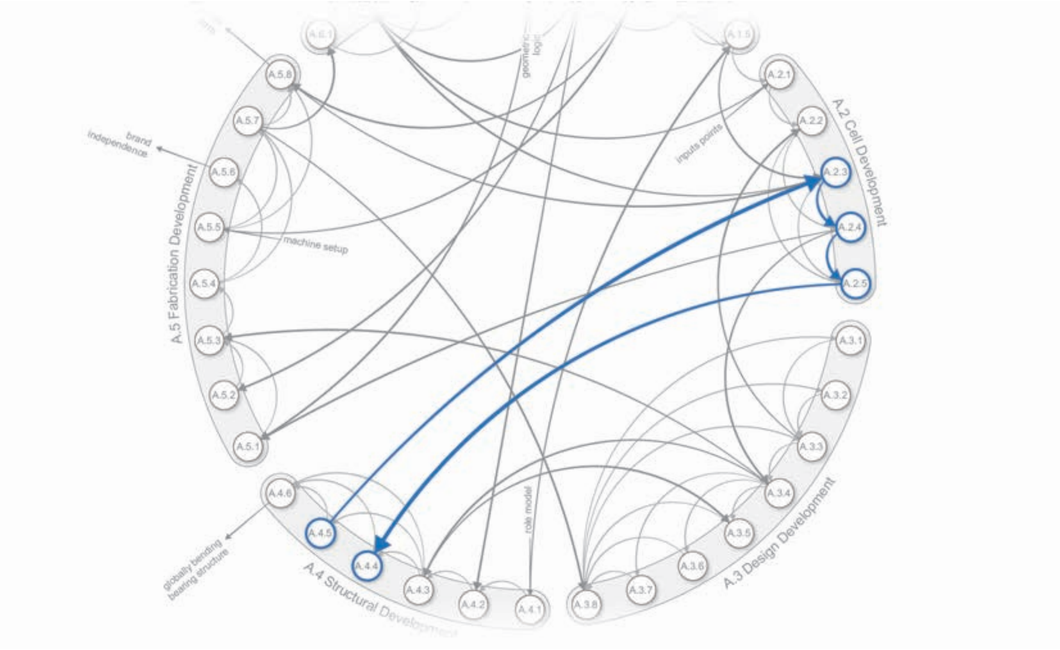


Figure 6.59: Concurrent relationship between cell- and structural development domains.

integrates the Modular Plate Cell (A.2.4), while the principle of Frustum (A.2.3) integrates the Machinic Morphospace (A.5.8) thereby affecting A.2.4 (Fig. 6.60).

The **third feedback loop** exists between the domains of design development (A.3) and structural development (A.4): the principle of Automated FEM (A.4.3) integrates Global Cell Arrangement (A.3.4), while the principle of Anisotropy (A.3.5) integrates structural evaluation of Automated FEM (A.4.3) thereby affecting A.3.4 via the principle of Steering of Form (A.3.8) (Fig. 6.61).

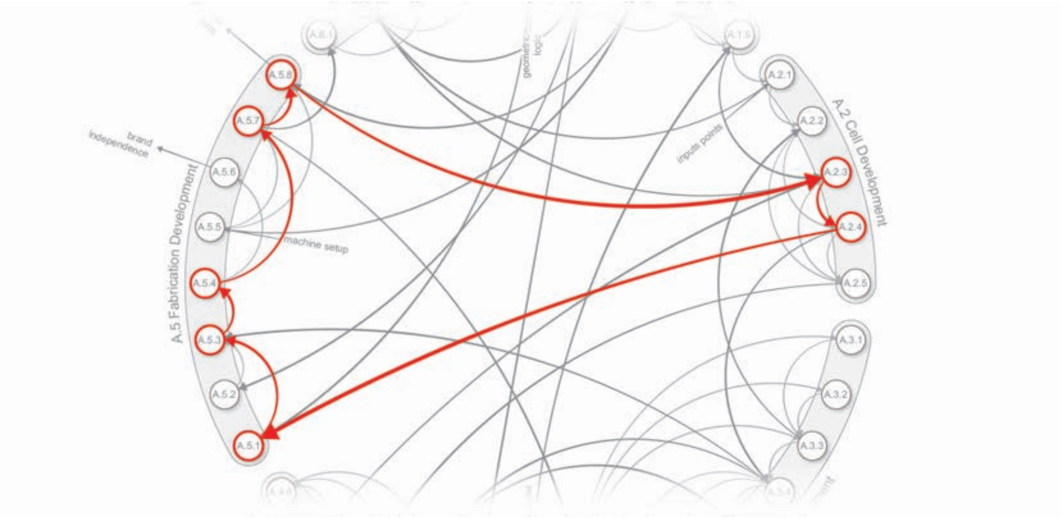


Figure 6.60: Feedback loop between cell- and fabrication development domains.

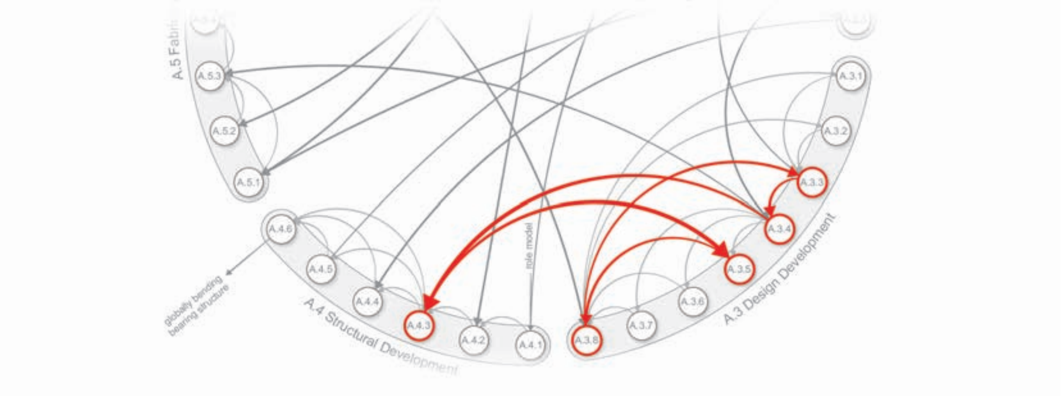


Figure 6.61: Feedback loop between design- and structural development domains.

The **fourth feedback loop** exists between the domains of design development (A.3) and fabrication development (A.5): the principle of Automated Fabrication Modeling (A.5.3) integrates Global Cell Arrangement (A.3.4), while the principle of Steering of Form (A.3.8) integrates Fabrication Validation (A.5.7) thereby forming a closed loop via the principle of Dynamic Equilibrium (A.3.3) (Fig. 6.62).

Furthermore, the feedback loops that have been identified above also suggest that there are **multi-lateral feedback loops** in the development process of Case A. Using the same method of tracing the flow of information, this can be confirmed between the domains of cell-, design-, and structural development and between the domains of cell-, design-, and fabrication development (Fig. 6.63).

The above representations, which are based on the inputs and outputs of low-level functional principles, are an important tool for tracing the flow of information through the network and thereby for explicitly identifying multi-lateral relationships between domains. Although most of the relationships might have been expected from experience, they would otherwise be difficult to pinpoint.

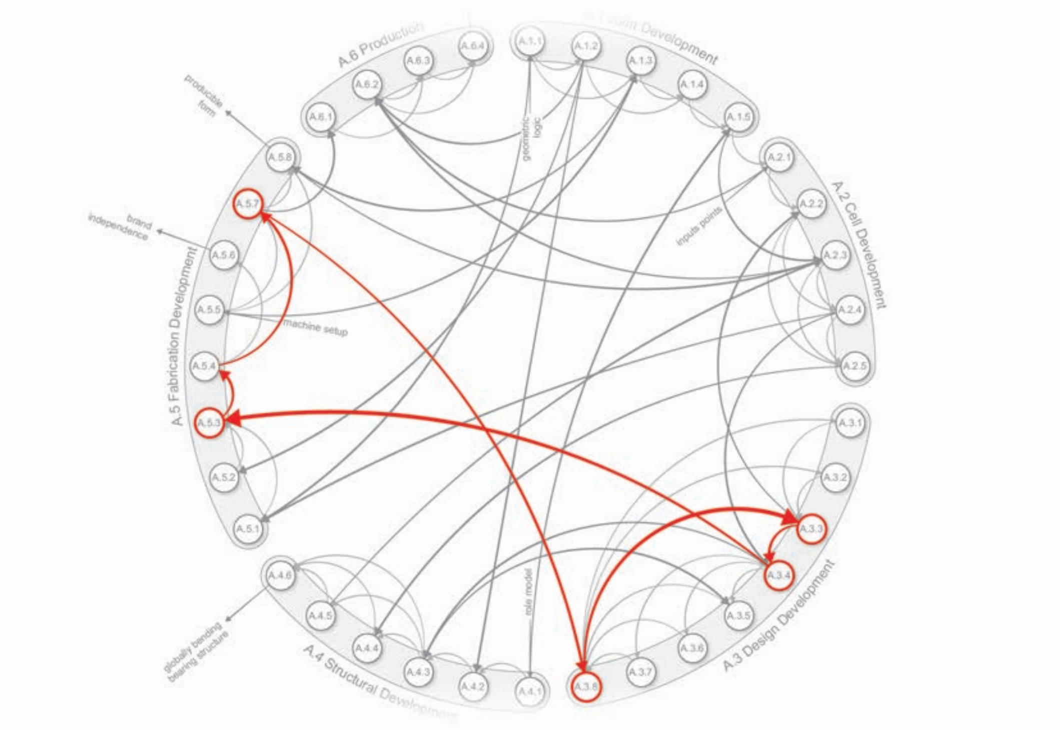


Figure 6.62: Feedback loop between design development and fabrication development domains.

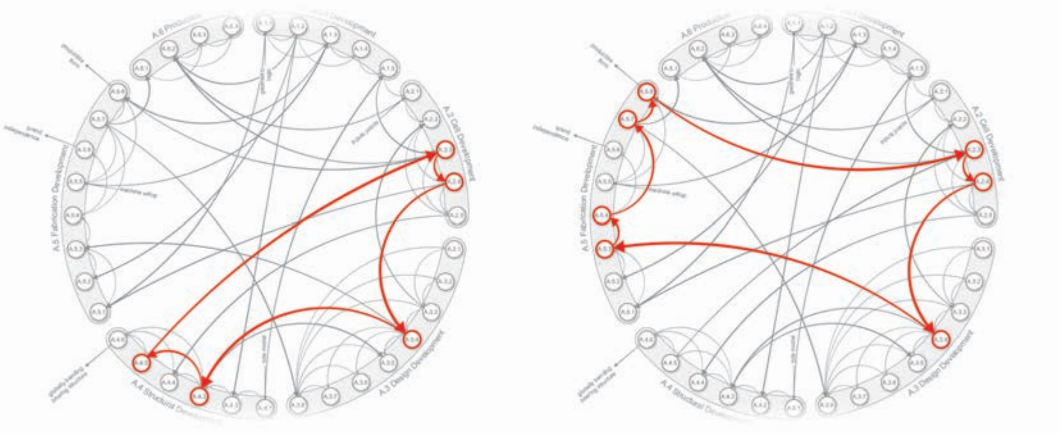


Figure 6.63: Multi-lateral feedback loops. Left: between cell-, design-, and structural development domains. Right: between cell-, design-, and fabrication development domains.

A second important aspect is that this approach also allows identifying relationships that were expected to be feedback loops, but turned out to be concurrent relationships. One example is the relationship between joint development and fabrication development, where fabrication sequence (A.1.1), Finger Joint Production (A.1.2), and Connecting Angles (A.1.3) turn out not to have been embedded in a feedback loop with fabrication development. In retrospect, this becomes clearer considering that joint development and fabrication development occurred in succession rather than in parallel.

As a reminder, these are not observations that apply to the design of segmented shells in general, but are specific to Case A. Rather, they are the specifics, from which a generalization will be attempted as part of the definition of design principles in Chapter 8.

6.6 Results and Discussion

At the centre of this case study was the analysis of a full-scale built prototype, the ICD/ITKE Research Pavilion 2011 (Case A), with regard to functional principles in the development of segmented shells—specifically in the areas of joint, segment, global design, structure, fabrication, and production development. Through analysis and abstraction of principles the case study not only highlighted the chain of causality in the development of the project but also revealed a previously implicit network of interactions and information flow through this network. Specifically, the case study led to the following two main results:

1. A total of 36 functional principles have been identified, defined and categorized, and their interactions illustrated and discussed.
2. Based on the analysis of the network, seven bi-directional, cross-domain relationships could be explicitly identified that were at the core of design development, of which four are feedback loops.

In addition to the low-level functional principles, their relationships, and the feedback loops between development domains, high-level principles could be identified that are relevant beyond the specific case for the design of segmented shells in general. Such principles are present in the areas of design development (A.3), structural and fabricational development (A.4 and 5), as well as production (A.6).

Specifically, in the principle of Steering of form (A.3.8), numerical parameters, such as stress in the structure, number of plate elements, resultant fabrication time, angles between plate, etc., but also geometric parameters, can be used as steering parameters for dynamic equilibrium methods that steer the formation process of the shell (see 4.3.4). In the domain of structural development, the high-level principle of Free-form segmented shells (A.4.6) can be applied to freeform, i.e., non-load-optimized shell shapes while still producing membrane stresses in the structure, unlike conventional lightweight construction. In the domain of fabrication development, the principle of Automated Fabrication Modeling (A.5.3) and the automation of machine code generation as described in the principle of NC-Code Generation (A.5.4) are prerequisites for the efficient fabrication of highly differentiated structures. Similarly, the principle of Machinic Morphospace (A.5.8) is a high-level principle that can be applied in other contexts as it is the prerequisite for fabrication-oriented design integration. Finally, in the production domain, the principles of Robotic pre-fabrication (A.6.1) and Assembly-ability (A.6.2) can be identified as general principles that apply beyond the specific case under investigation.

One of the most important findings of this case study, however, are the seven bi-lateral relationships in the development of the ICD/ITKE Research Pavilion 2011 that have been made explicit. This is important for two reasons: first, this allows identifying the missing links in the flow of information responsible for the lack of feedback in established design processes, which results in the challenges mentioned above that the building industry is facing; second, while top-down relationships can conveniently be represented through hierarchical modeling approaches such as parametric modeling, bottom-up relationships are still hard to model. This consequently raises the question, which methods and approaches are particularly suited to feed down-stream results back into the formation process.

In the case of the ICD/ITKE Research Pavilion 2011 these loops have been implemented intuitively as design iterations, through empirical and repeated parametric modeling. Behavioral computation, however, now provides an opportunity to facilitate and implement functional feedback loops beyond conventional design iterations.

In the case of the Landesgartenschau Exhibition Hall, for the first time in the context of the design of segmented shells, an agent-based approach has been utilized to integrate multi-disciplinary requirements and constraints into the design process based on the definition of low-level rules, which are termed behaviors. This case, which confirmed the viability of this approach, is the subject of the following case study.

6.7 Outlook

While Case A demonstrated the potential that integrated computational design and robotic fabrication of finger-jointed segmented shells offer for lightweight construction, more than constituting a one-off, its physical manifestation can now be viewed as the outcome of the flow of information through this specific network of functional principles.

Furthermore, the explicitly stated network of principles and correlations is the pre-condition for the further development of design principles for segmented shells. By reframing the pavilion as the physical expression of the flow of information in this network, a new hypothesis can be stated with regard to the transferability of performative aspects of the case, which will be investigated in further research. The hypothesis being that, if the outcome of the specific network of principles is performative, then the network itself encapsulates the performativity and makes it transferable.

Further development will thus focus on transferring functional principles to design principles (see 8) and on implementing selected design principles within an ABMS software framework for the design of segmented shells, which will be presented in the development chapter of this dissertation (see 9).

The prospect of this inductive research approach therefore is to be able to abstract and generalize parts of the network as high-level design principles for segmented shells, to implement them computationally as part of the ABMS approach, and, finally, to be able to reuse and evaluate them in different contexts resulting in different design outcomes.

7

Case Study II: Landesgartenschau Exhibition Hall

7.1 Introduction

Only a few segmented timber shells have been built so far, mostly in academia, without consideration of the requirements of building practice. While the introduction of robotic fabrication has enabled the production of differentiated building elements within economic constraints, the development of segmentation, joints, and assembly still poses challenges to the applicability of segmented timber shells in building practice.

This study consequently investigates the case of a segmented timber shell that was developed at the intersection of research and practice—the Landesgartenschau Exhibition Hall ("Case B"). In this project, the consideration of the requirements of building codes and regulations, economic considerations of fabrication and assembly, as well as quality control formed an integral part of the research and development. The findings of the previous case ICD/ITKE Research Pavilion 2011 (see Chapter 6) thereby formed the starting point for its development.

In the following sections, Case B is first contextualized in view of its specific background, in addition to the general background to the case studies given in Section 5.5. The background chapter is then followed by the main body of the case study, which focuses on the identification of functional principles. These are established by associating the project's results with the input factors that drove the development of the approaches and methods—the 'mechanisms' that relate input and output (see

Section 5.3). Finally, the identified functional principles are summarized and their role and impact on the different stages of the design and realization of the segmented shell discussed.

7.1.1 Main Characteristics and Relevance of Case B

According to the previously established taxonomy (see 4.2), the Landesgartenschau Exhibition Hall can be categorized as a free-form, single-layer segmented timber shell consisting of planar polygonal segments. Made from Beech plywood with a material thickness of 50 mm, a structural span of 10.78 m (material thickness/span ratio of 1/215), and a structural weight/floor area ratio of 66.56 $\mathrm{kg\,m^{-2}}$, the shell is a lightweight thin shell.[1]

Located in the city of Schwäbisch Gmünd, the exhibition hall is a fully enclosed, insulated and waterproof building that hosted an exhibition during the biennial landscaping exhibition “Landesgartenschau” in 2014 and has since been serving as an educational and event space (Fig. 7.1).

Figure 7.1: Interior view of Landesgartenschau Exhibition Hall. From [103] by Krieg (2014).

The project was developed from spring 2013 to the end of the same year, followed by roughly two months of fabrication, one month of assembly on site, and one month of interior fit-out and finishing, in 2014. The hall was opened at the end of April 2014 coinciding with the opening of the Landesgartenschau.

The Landesgartenschau Exhibition Hall is a freeform segmented shell consisting of two dome-shaped, synclastic surface areas and one saddle-shaped, anticlastic area in between. The dome-shaped spaces each feature a large opening, which is

---

[1] Given the shell surface area of 243.14 $\mathrm{m^2}$ and the specific weight of the Beech plywood of 730 $\mathrm{kg\,m^{-3}}$ [80, p. 9], the weight of the structural layer is 8.94 t; the gross floor area is 134.31 $\mathrm{m^2}$; for comparison, the weight/shell surface area is 36.77 $\mathrm{kg\,m^{-2}}$.

supported by columns that hold a glass facade.

A characteristic feature of the interior space and thus part of the architectural experience are the 243 individual Beech plywood plates with their visible finger-joint connections forming the interior finished surface of the shell. The polygonal outlines of this visible structural layer reflect the underlying surface curvature: convex in synclastic areas and concave, bowtie-shaped, in the anticlastic areas. While passing from the first, lower one of the two dome-shaped spaces to the taller main space with its apex at 6 m, the gradual transformation from convex to concave and back to convex plate outlines becomes apparent. On the outside of the shell, the cladding layer made from Larch 3-ply mirrors this transformation (Fig. 7.2).

Figure 7.2: Changing plate outlines of the exterior cladding layer reflect the changing curvature of the shell (Halbe, 2014).

Case B demonstrates the structural, geometrical and fabricational developments in the design of segmented timber shells: 243 geometrically unique Beech plywood plates with a combined surface area of 243.14 m^2 were robotically prefabricated and joined along their edges by 7356 individual finger joints. Approximately 12 m^3 of plywood were used to enclose 605 m^3 of volume while covering a usable floor area of 125 m^2.

The main contributions of Case B and hence its relevance to the field of segmented timber shell design are:

1. It is one of the first buildings to use Beech plywood as the main structural building material [103, p. 124].
2. It is the first shell whose structural layer is entirely robotically prefabricated [102, p. 139].
3. Its custom plate connection consisting of finger joints and crossing screws is

robotically fabricated and easily installed also in anticlastic shell surface areas [115, p. 138].

4. It is the first segmented timber shell whose segmentation was developed using an agent-based design approach focussing on the integration of material and fabrication constraints in the design process [172, p. 187].

The research pursued a bottom-up approach, where constraints of material and fabrication were identified as 'low-level' design drivers and integrated into the design process in order to achieve producible load-bearing configurations of shell segments on the 'high level' of the entire shell. Using ABMS, the requirements were implemented on the level of the building element as agent behaviors leading to goal-oriented system effects on the macro-level. As of today, the only other known segmented shell to have been designed using a similar approach is the BUGA Wood Pavilion at the Bundesgartenschau in Heilbronn in 2019.

As such, Case B also forms the starting point for the further development of the ABM method for segmented shells as part of this research (see Chapters 8 and 9).

7.1.2 Scope of the Case Study

This case study investigates design integration of architectural, structural, and fabricational requirements in the context of joint, segment, global design, and fabrication development in the Landesgartenschau Exhibition Hall.

The focus of the investigation specifically is on identifying discrete steps in the development, fabrication and construction processes, on the methods and approaches (mechanisms) used in the steps, on the relationships between discrete steps, and on their role in the design process, e.g., as design drivers or constraints.

This analysis will not focus on the technical details of the ABM implementation used at the time. Rather the implementation of design principles derived from this case study and the previous one will be the focus of Chapter 9. Furthermore, methods and approaches that were not immediately affecting the development process, for example in structural simulation and analysis or in the evaluation of the built structure through 3D laser scanning, are outside the scope of this case study.

7.1.3 Research Team and Contributions of the Author

The Exhibition Hall is a research demonstrator for the interdisciplinary research and development project "Robotics in Timber Construction." Involving architects, structural engineers, geodetic engineers, and timber construction engineers, research partners included the Institute for Computational Design and Construction (Prof. Achim Menges, Oliver D. Krieg, Tobias Schwinn), the Institute for Building Structures and Structural Design (Prof. Jan Knippers, Jian-Min Li), and the Institute of Engineering Geodesy (Prof. Volker Schwieger, Annette Schmitt) at the University of Stuttgart, and the timber construction firm Müllerblaustein Holzbau GmbH (Reinhold Müller, Benjamin Eisele).

The research project was funded by the European Union EFRE fund and the federal state of Baden-Württemberg through the “Cluster Forst und Holz Initiative.” Additional funds for the construction of the demonstrator were provided by the Landesgartenschau Schwäbisch Gmünd GmbH and ForstBW, by the project participants and by additional supporters. For the full credit list see Appendix A.2.

Contributions of the author included (1) the initial review of applicable modeling approaches for segmented timber shells consisting of planar segments; (2) the development of the integrative agent-based modeling approach and its implementation within Rhino®and Grasshopper®; (3) the collaborative design of the demonstrator building with Prof. Achim Menges and Oliver D. Krieg; (4) the development of process simulation including the kinematic model of the industrial robot and the development of the robot code generation; and (5) the off-site robotic pre-fabrication of segments at Müllerblaustein’s facilities in Blaustein together with Oliver D. Krieg and the team at Müllerblaustein.

The results of this research project have been peer-reviewed and published in conference proceedings and journals. In particular, the case study references approaches and findings described in Krieg, Schwinn and Menges [102], Krieg et al. [103], Li and Knippers [114], Li and Knippers [115], Schmitt and Schwieger [169], Schwinn, Krieg and Menges [172], and Schwinn and Menges [175].

7.2 Specific Background and Objectives of Case B

Complementing the general research context and background described in Chapter 5, this section summarizes the specific context and background of Case B, including the objectives that were pursued in the research.

While the overarching research aimed at the development and construction of a robotically fabricated, lightweight timber plate system through a biologically informed, integrative computational design method [103, p. 109], the individual objectives primarily fall into six research domains: (1) trivalent plates structures and their joints, (2) planar approximation of free-form surfaces, (3) structural design, (4) the development of an integrative design approach, (5) quality control of the fabrication process, and (6) the development of a novel light-weight, plate-based building system for freeform surface shells and demonstration in a prototype building.

7.2.1 Trivalent Plate Structures and Finger Joints

Previous research into segmented shells showed that “the individual plates in a plate structure can be arranged in such a way that they are the primary load bearing elements” [172, p. 183]. Specifically, La Magna, Waimer and Knippers [108] confirmed previous statements in literature, for example by Wester [208], that in plate structures with a trivalent plate layout, where “three plates meet at one point and are hinged at the three intersection lines, each plate is constrained by the other two. Thus bending stiffness is generated even though the joints themselves are only hinged connections” [114, p. 3]. As stated by Bagger [9], in this case the structural stability is based on in-plane axial forces and in-plane shear forces distributed along the plate edges (see 4.4.2).

Teethed and interlocking connections, such as finger joints, were subsequently identified as particularly suitable for the connections between the segments in segmented timber shells as they are hinged connections with limited bending-bearing but very high shear load-bearing capacity. The particular arrangement of the segments combined with a shear resistant joint type can result in a globally bending-bearing structural behavior while being composed of bending-weak, hinged connections (see [108, p. 2]).

Joint design thus is a crucial part of the development of lightweight timber plate structures not only with respect to ensuring load transfer between the plates but also with respect to meeting building code requirements and to affecting the ability to assemble a segmented shell structure. Krieg et al. [103, p. 115] even identified the “lack of applicable joint constructions satisfying conditions of sufficient structural load transfer and the restricted adaptability to various geometric and structural situations” as the main reason for the scarcity of segmented shell structures in building construction.

Different types of robotically fabricated finger joints have consequently been developed for specific structural and architectural tasks, including the ones described in the previous case study (see Chapter 6) or the micro-finger joints for curved edges described in Krieg and Menges [99]. Following this previous work, the objective in the development of Case B thus was

i. to further develop plate arrangement and timber joints in order to meet the requirements of the building industry in terms of building code and regulations, efficient and economic use of material, to keep production times short, and to facilitate straightforward assembly [103, p. 119].

7.2.2 Planar Approximation

Achieving load-bearing capacity in double-curved shell structures through the specific arrangement of hollow, polyhedral segments (“plate cells”) consisting of planar plates has been demonstrated in La Magna et al. [109], albeit at the cost of an increased number of resultant plates. One of the challenges involving plate structures in the context of an application in the building industry thus constituted the reduction of the number of segments necessary to cover a double-curved surface: in other words, the approximation of non-trivial geometry through constraint-based subdivision, of which the main constraints are planarity and trivalence.

Approximation of point clouds, for example generated by 3d-scanning, or of double-curved surfaces through planar subdivision is a topic that is actively being researched in the field of computer graphics (see [41; 219]). In the architecture field, structural performance criteria, building technology, fabrication, and building materials, which usually come in planar form, also have to be taken into account. Therefore planar approximation of free-form surfaces is a highly relevant and active field of research as demonstrated by numerous publications on polygonal [72; 116; 126; 192; 203; 204] and quad-based planar approximation [53; 83; 118; 148; 218].

Bagger [9, p. 10] reports that in 1984, Wester has proposed an approach for approximating synclastic surfaces with plane-based facetted geometry that is similar to “cutting off slices of an apple with plane random cuts until all the peel is gone.” Another recurring approach in literature is the so-called Tangent-Plane-Intersection (TPI) method [126; 192; 203] (Fig. 7.3).

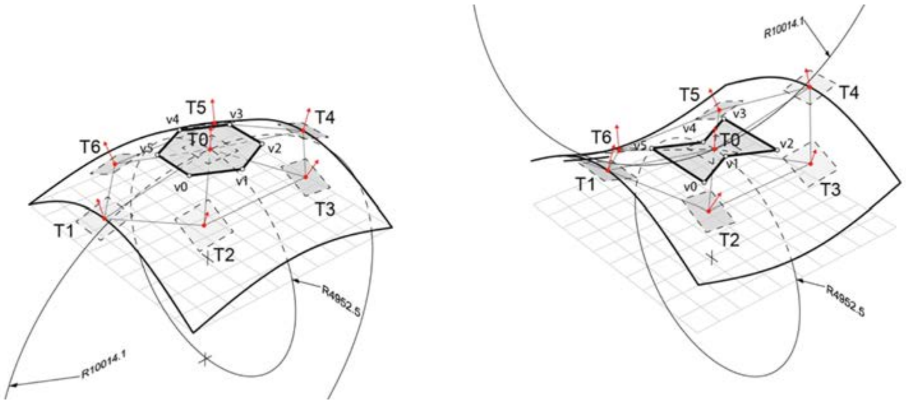


Figure 7.3: Tangent plane intersection. Images from [172] by Schwinn (2014).

TPI offers several useful characteristics that make it particularly relevant in the context of designing segmented timber shells:

1. All intersection points of a tangent plane T_0 with its neighboring planes lie in the plane of T_0 and can consequently represent the vertices of a planar polygon [103, p. 113].

2. In synclastic and anticlastic surface areas, TPI produces perfectly planar polygonal outlines without the need for a computationally expensive iterative approximation and therefore allows for designer interactivity.
3. The duality between the plane-based polygonal subdivision and the point-based triangular network, where edges describe adjacencies between polygons and vertices describe tangency points, states that for every triangulation of a double-curved surface exists a corresponding trivalent plate structure generated by TPI.
4. Such a trivalent configuration is beneficial for the structural behavior of plate structures, as shown by Wester [208], Bagger [9], and La Magna et al. [108].
5. A characteristic feature of the TPI method is that the outlines of the resulting segments not only reflect the topology of the triangular dual, but also the curvature of the underlying surface: convex in synclastic areas and concave, star-shaped in anticlastic areas.

The main effort in this approach then becomes the generation of a triangulation that yields a valid plate layout. The conjugation method proposed by Wang et al. [203] and the advancing front method proposed by Troche [192] for triangulating double-curved surfaces both respond to the local curvature of the underlying surface and produce meaningful results for some types of doubly curved surfaces. However, both are procedural, top-down triangulation approaches and only allow for global and indirect control of the triangles and, conversely, of plate sizes. This makes integrating material and fabrication constraints into the plate generation difficult.

Manahl et al. [126], while focussing on ornamental panelization, try to address this issue by giving the designer control over the individual plate geometry through the interactive manipulation of the vertices in the triangular dual. However, due to the ripple effects that changing one plate has on the neighboring plates, this process quickly becomes impractical with rising number of segments and manipulations. This leads to the conclusion that rules need to be defined, by which vertices in the mesh change location based on the resultant plates.

The objective regarding planar approximation consequently was

ii. to embed TPI as part of a rule-based computational approach, which changes the position of vertices in the mesh based on the morphology of their associated plates and on the local surface properties, while taking into account aspects of fabrication and logistics.

This would then allow to steer each plate toward meeting defined performance criteria, ultimately resulting in an architectural constraint-based segmentation. Such a segmentation would be generated from the bottom up based on local constraints as opposed to a top-down planar approximation based exclusively on geometric concerns.

7.2.3 Structural Joint Design

One of the challenges in the structural design of segmented shells is to join the segments in such a way that they become an integral structure that generates shell action and transfers external loads into membrane forces [115, p. 123]. As stated above, segmented plate shells with trivalent plate layout are kinematically stable, similar to triangulated lattice shells (see 4.4.2). Connections that provide bending stiffness are thus not required [103, p. 115]. However, trivalent polyhedrons will become kinematically unstable if the angle between two consecutive edges is close to 180° [115, p. 130]. The question consequently was, to what extent adding bending stiffness in connections could improve the structural performance of plate shells, especially in areas where interior polygon angles approach 180° [115, p. 130].

Furthermore, as the segments in a shell have to be produced with a certain gap between the plates in order to facilitate assembly, the precise offset from the joint axis would have to be determined. From a structural point of view, this led to the question of how the gaps between plates would influence the overall performance of the plate structure [115, p. 130].

Regarding structural design of the joint, objectives thus were

iii. to design a connection that is viable for an application in the building industry in that provides sufficient strength and stiffness between the panels, while meeting the requirements of building code;
iv. to study adding bending stiffness in the joints in those areas where the angle between consecutive plate edges approaches 180°; and
v. to study the influence of the joint gap on the global structural performance of the shell.

7.2.4 Integrative Design Approach

Previous work showed that an integrated, performance-driven computational design approach is paramount to the development of an architectural timber plate structure [109; 171], as the result of a conventional form-driven design approach can hardly be re-engineered in the later stages of the design process, for example, with regards to light weight or fabrication. Therefore, in order to achieve a synthesis between design and fabrication, the rules and constraints of a given fabrication approach and even of a specific fabrication setup were considered design drivers informing the design process. To this end, the machinic morphospace method was developed, which, by delineating the producible from the geometrically possible, allows the conceptualization of the solution space of a particular fabrication setup within the design domain [129, p. 38].

While this method allows the evaluation of a design with respect to its producibility, the open question with regards to design integration was how fabrication characteristics and material properties can now actively drive design processes through feedback, effectively forming a Digital Loop (as opposed to the established Digital Chain).

While such a loop has been implemented as a manual process in Case A, as demonstrated through the analysis of the flow of information in the case study, the research focus in Case B was on a coherent computational approach for synthesizing form-generation and materialization. Using agent-based modeling, the premise was that through the calculation of the local interactions between individual building elements (autonomous agents) with defined rule sets (behaviors), a global configuration of plates emerges that meets higher-level goals and constraints [103, p. 113].

The corresponding hypothesis was that developing computational methods for integrating material information, structural behavior, fabrication and assembly constraints into the design stage in the form of a Digital Loop would essentially provide two main advantages: (1) not only would the effectiveness of the planning process as a whole likely be increased; but (2) such a computationally integrated approach would also enable novel architectural and structural solutions that are outside the scope of conventional design approaches.

Regarding design integration, the objective thus was

vi. to computationally integrate various, potentially conflicting requirements of fabrication, geometry, structure, and architecture in a behavioral model in order to form a Digital Loop [103, p. 112].

7.2.5 Quality Control

The question regarding the role of joint gaps in the structural performance is closely related to structural and assembly tolerances that fabrication process and material behavior have to meet since both are the main determinants of the resultant gap size [115, p. 130]. Consequently, the accuracy of the robotically fabricated plates needed to be tracked in order to be able to determine if the fabrication process meets the tolerances required by structural joint design. This would not only ensure that the joint was structurally viable and that assembly was possible, but it would also allow to evaluate if the accuracy of robotic fabrication was sufficient to meet the accuracy requirements for the fabrication of segmented timber shells.

Regarding quality control, objectives thus were

vii. to determine the accuracy of the fabrication process using laser-tracking and to validate robotic fabrication for the pre-fabrication of segmented timber shells.

viii. to track the behavior of the building during its lifetime by repeatedly surveying the global shell geometry using 3D-laser scanning.

7.2.6 Building System and Demonstrator

Regarding the development of a building system for segmented timber shells, the focus was on a modular, lightweight plate system capable of adapting to free-form surfaces by making use of the extended design space, or Machinic Morphospace, of robotic fabrication [103, p. 112], while taking into account the requirements of the building industry. As described above, this included the further development of integrated timber joints and segmentation towards a finger-joint plate system. From a computational point of view, the TPI method would ensure planar building elements in the segmentation of the shell as part of a constraint-based approach. The objectives thus were

ix. to integrate the developments mentioned above into a modular, adaptive building system consisting of planar plates;

x. to demonstrate the applicability of segmented timber shells in building practice, by integrating the functional requirements of an exhibition space in the design process.

This required not only the ability to integrate (1) the demands of the design brief with regards to program, but also (2) code compliance; (3) producibility; (4) requirements for permanent buildings, such as wind and snow load requirements, full enclosure, and building physics; as well as (5) interfacing of the shell with other building systems, notably the facade.

7.3 First-level Functional Principles in Case B

At the core of both case studies is the question *how* the results in the cases under investigation were achieved. The purpose of this section therefore is to identify functional principles in the design and development process that have led to the research results. The principles are arranged in increasing hierarchical order from low-level to high-level principles.

Similar to Case Study 1, low-level functional principles are grouped by development domain in which they occur. Domains are the development of joints, segments, shell form-finding, structural design, fabrication, quality control, and assembly and construction principles. By describing the methods (mechanisms) that were used at the lowest level of the development, a causal relationship can be established between the results (outputs) that were achieved in each development area and their input factors and objectives (inputs).

The individual low-level principles are defined in the following sections. The sections B.1-B.9 thus constitute the basis for the investigation into causal dependencies and relationships between development domains, and finally for the definition of design principles.

B.1 Joint Development Principles

B.1.1 Principle of Finger Jointing

The Principle of Finger-jointing associates the teethed pattern of finger joints with their geometric logic and the logic of subtractive fabrication via spot-facing and flank milling (Fig. 7.4).

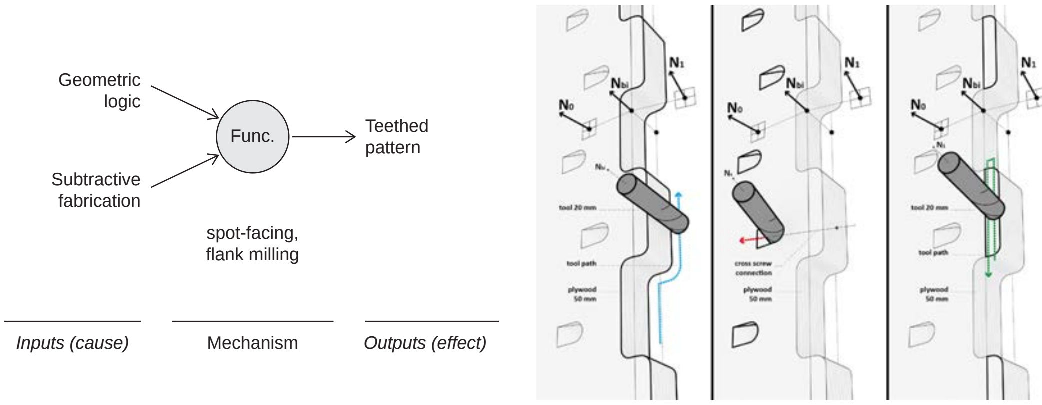


Figure 7.4: Principle of Finger Jointing. Right image from [172] by Krieg (2014).

Inputs (cause) – Geometrical: in traditional finger joints, the depth of the fingers and their structural capacity are a function of the angle between plates; assembly of more than two plates along one assembly vector is impossible. Fabricational: milling with the tool shaft reduces fabrication time; concave tool paths, however, when milled with the tool shaft, result in rounded corners.

Methods (mechanism) – The finger joint geometry has been adapted (1) to allow for the depth of the finger joints to be independent of the angles between plates; (2) to enable assembly of more than two plates along one insertion vector; and (3) to increase fabrication speed by milling the joints with the tool shaft (flank milling). The rounded corners in the indented part of the joint are matched by the rounded corners in the protruding counterpart of the joint.

Outputs (effects) – The result is a custom teethed joint type for edge-wise plate connections.

Significance – A precondition for the principle of Shear-load transfer.

B.1.2 Principle of Shear-load Transfer

The Principle of Shear-load transfer associates shear-load bearing capacity of the joint with a teethed pattern in an edge-wise connection via interlocking between the plates (Fig. 7.5).

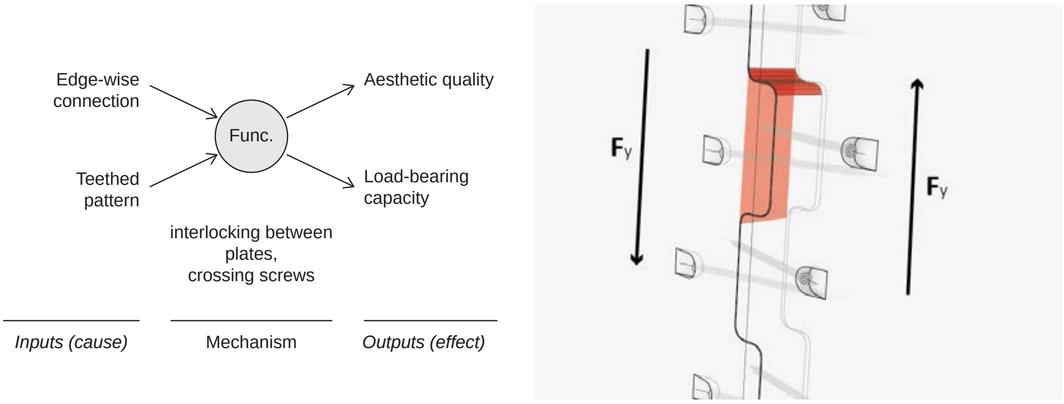


Figure 7.5: Principle of Shear-load Transfer. Right image from [103] by Li (2015).

Inputs (cause) – Edge-wise connected plates with integral connections around their edges.

Methods (mechanisms) – The connection functions similar to step joints in traditional truss works: “the thrust force [is] taken by the contact surface and transferred to the shear plane” [115, p. 126]. In order to accommodate axial forces and the out-of-plane shear forces, for example in the assembly direction, crossing screws were inserted, which lie in parallel planes, normal to the plate edge [103, p. 117]. The crossing screws also resist bending moments between the plates, which occur when the angle between two consecutive plate edges approximates 180 degrees (see principles B.4.2 and B.2.3).

Outputs (effects) – The resulting connection is characterized by the pattern of alternating ‘teeth’ along the plate edges, which is composed of individual interlocking joints of 94 mm width and 30 mm depth (excluding joint gap), high in-plane shear capacity, and to a lesser degree bending bearing capacity.

Significance – The principle is a precondition for the principle of Material Overlap and Geometric Compatibility.

B.1.3 Principle of Material Overlap

The Principle of Material Overlap associates a minimum shell thickness with the use of metal fasteners, such as screws, in edge-wise connected plate segments based on the specific building code requirements (Fig. 7.6).

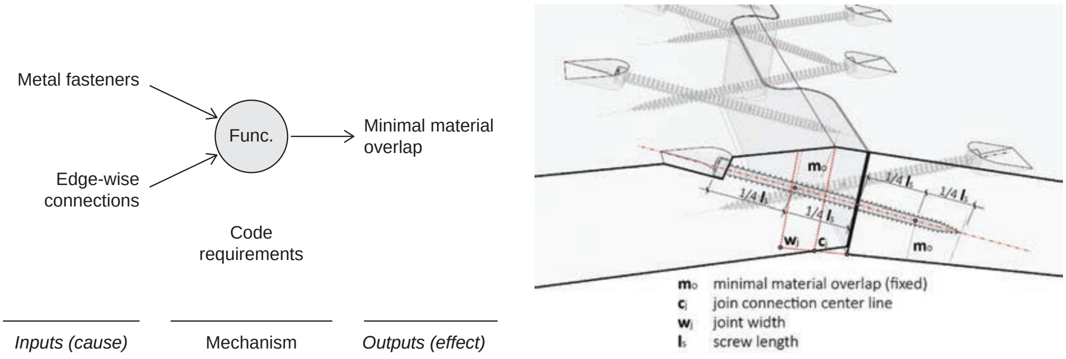


Figure 7.6: Principle of Material Overlap. Right image from [103] by Krieg (2014).

Input (cause) – In integral joints, the load transfer mostly occurs through the geometry of the joint, for example through interlocking. In this case, however, additional crossing screws were inserted in the edge faces of the plates in order to accommodate in-plane and out-of-plane shear and axial forces. When using screws certain minimal offsets from component edges and spacing of screws have to be observed in order to avoid splitting of the wood.

Methods (mechanism) – Following *Building code requirements* defined in the European Technical Approvals (ETA-12/0114 2012) resulted in a minimum offset of $5 * d$ from the centroid of the screw immersed in the material to the plate edge, which determined the length of the screw ($1/4 * l_s >= 5 * d$); and in a minimum of $2.5 * d$ from the centroid to the surface of the material [115, p. 130].

Output (effect) – Taking into account the results of physical load testing and structural analysis of the global structure, pairs of fully threaded screws of 120x6 mm (length x diameter) were used with 20 mm distance between each other. As the screws are installed from the outside of the structure, shorter screws of 80 mm length were used in anticlastic surface areas in the negative principal curvature direction to reduce the guiding slot length [115, p. 132].

Significance – The screw diameter of 6 mm, itself a function of structural requirements, thus determined the resultant timber plate thickness of 50 mm. Required material overlap thus is one of the most critical aspects in the development of a joint for edge-wise connected timber plate structures as it determines the minimal possible thickness of the shell.

B.1.4 Principle of Geometric Compatibility

The Principle of Geometric Compatibility associates geometric compatibility/ assemble-ability of finger-jointed plates with the surface curvature at the location of the plate via the orientation of the edge faces (Fig. 7.7).

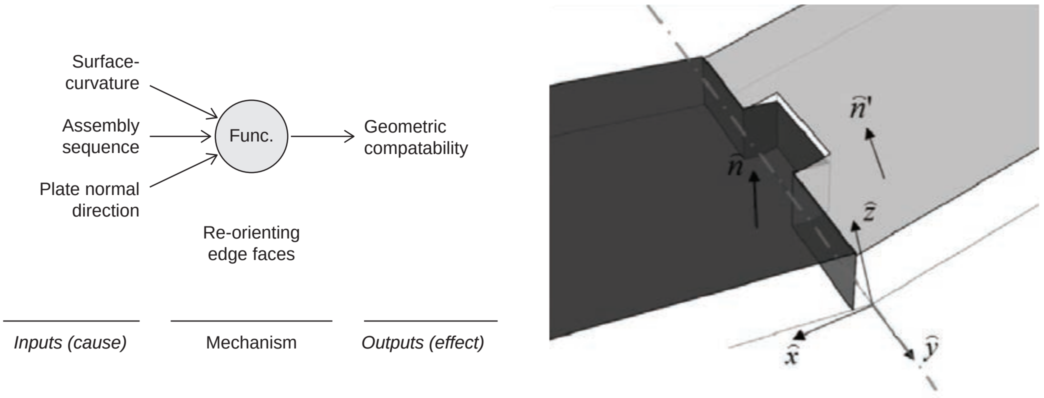


Figure 7.7: Principle of Geometric Compatibility. Right image from [115] by Li (2015).

Inputs (cause) – The Principle of Geometric Compatibility responds to *Assembly-related constraints* requiring a clear path in the direction of the plate normal for the insertion of plates from the outside. Depending on the Gaussian curvature of the surface, however, edge faces might be angled in opposite directions leading to blockage in the assembly direction.

Methods (mechanisms) – In both positive principle curvature directions in synclastic areas, the local joint coordinate system of an edge face is aligned with the sum of the face normals of the adjacent plates (z-axis) and the edge direction (y-axis). In this case no special assembly sequence is needed.

In anticlastic surface areas, this strategy leads to blockage due to the opposite orientation of the edges in the principal curvature directions. Therefore, the local joint coordinate systems of edge faces that are situated in the negative, concave curvature direction need to be re-aligned: parallel with its own plate normal (z-axis) and the edge direction (y-axis) at edges that facing the previously installed plate; and parallel to the face normal of the adjacent plate (z-axis) and the edge direction (y-axis) at edges that are facing the subsequently assembled plate (see [115, p. 132]). This requires the definition of a plate assembly sequence.

Outputs (effect) – In anticlastic surface areas, edge orientation and geometric compatibility are a function of assembly sequence and of surface curvature.

Significance – This principle ensures assemble-ability of the segmented plate shell.

B.1.5 Principle of Joint Adaptation

The Principle of Joint Adaptation associates an adaptive joint model with varying plate angles, principle curvature directions, and varying structural requirements via an integrative parametric model of the joint (Fig. 7.8).

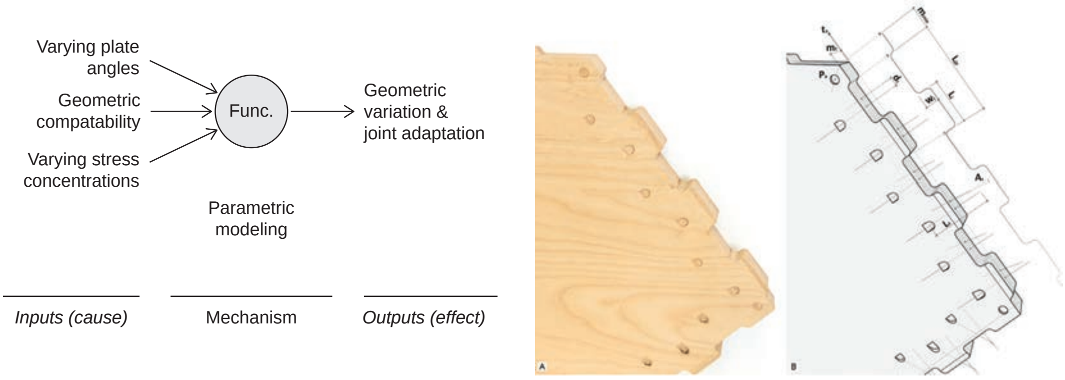


Figure 7.8: Principle of Joint Adaptation. Right images from [172] by Krieg (2014).

Inputs (cause) – The Principle of Joint Adaptation responds to *geometric and structural requirements*. Similar to the case of unstrained grid-shells, joints need to adapt to varying angles between building elements in segmented plate shells (see Section 4.2.4); additionally, local adjustments of edge orientations to ensure geometric compatibility.

From a structural point of view, joints need to be engineered to accommodate varying stress concentrations due to different load cases and based on the location of the joint in the shell.

Methods (mechanisms) – These functionalities were implemented using a generative, parametric modeling approach, thereby making every detail adaptive to locally varying requirements and to local and global parameters [172, p. 183]. Examples of geometric parameters are the connection angle between two adjacent plates and their shared edge's length as well as material thickness and joint size; example for structural parameters are the number and location of crossing screws per joint [172, p. 184].

Outputs (effect) – The result is an integral timber joint for edge-wise connecting timber plates that is not only adaptive to geometric and structural requirements in a plate structure, but also includes assembly-related features, such as screw pockets that indicate the location and direction of the crossing-screws, thereby facilitating assembly.

Significance – The integration of multiple functional requirements into the connection design through an adaptive joint model is an example of the principle of Geometric Differentiation (B.1.6). Joint adaptation is the pre-condition for meeting structure- and assembly-related requirements.

B.1.6 Principle of Geometric Differentiation

The principle of Geometric Differentiation is a general principle that associates a geometrically differentiated outcome with a diverse set of input factors via functional integration mechanisms (Fig. 7.9).

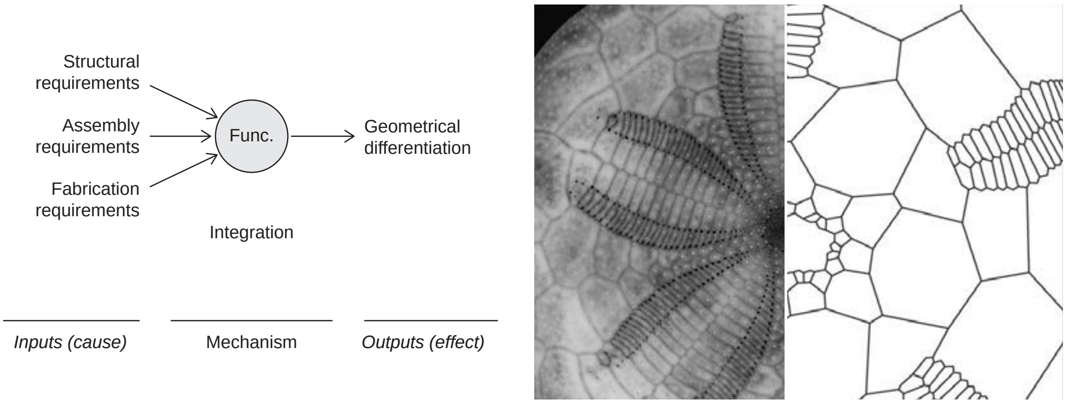


Figure 7.9: Principle of Geometric Differentiation. Center-right: Top view of sea urchin skeleton. From [130] by van der Steld (2012). Right: Schematic view of plate arrangement. From [100] by Krieg (2012).

Inputs (cause) – Structure-, assembly-, and fabrication-related input factors.

Methods (mechanisms) – Geometric differentiation can be observed in biological structures where is not only closely related to performance, e.g. their capacity for survival, but also negotiates between different function-based requirements [102, p. 125 and 134].

In this case, an adaptive joint model that allows the concurrent integration of the mechanisms described in the principles of geometric compatibility, material overlap, and in structural joint development principles.

Outputs (effects) – The result is a geometrically differentiated, locally adaptive, large-scale finger joint connection for timber plates with embedded functionalities.

Significance – Geometric differentiation of building elements is a fundamental principle for the design of lightweight structures, which can also be observed in nature. The project demonstrates the corresponding design concept “less material and more form” [102, p. 140].

B.2 Structural Joint Principles

B.2.1 Principle of Cross-laminated Veneer

The Principle of Cross-laminated Veneer associates the use of multi-layer Beech plywood (multiplex) with the use of screws in timber plates that are edge-wise connected at the perimeter via material characteristics and building code requirements (Fig. 7.10).

Figure 7.10: Principle of Cross-laminated Veneer. Right image by ICD/ITKE/IIGS University of Stuttgart (2014).

Inputs (cause) – Crossing screw connections between plates occur at all edges of a plate and membrane forces might come from various directions. A material is thus desirable that has similar structural properties in all directions, such as plywood and cross-laminated timber [115, p. 125].

Methods (mechanisms) – In plywood, alternating layers of veneer are laminated in 90° angles to each other. The stiffness and strength of plywood made from Beech wood are about twice as high as plywood made from softwood, and therefore less material and fewer screws are needed [115, p. 125].

Furthermore, grain directions parallel to the screws need to be avoided. The thickness of individual layers of veneer in Beech plywood is much lower than that in cross-laminated timber, which ensures that the screws penetrate multiple layers of different grain directions. For the exact make-up of the plywood, see [115, p. 126] and [80].

Outputs (effect) – Beech plywood/ Laminated-Veneer-Lumber (LVL) was chosen due to its material characteristics and code requirements related to the use of screws in wood connections.

Significance – The use of Beech plywood is the pre-condition for realizing a thin shell of just 50 mm thickness. Furthermore, promoting the use of Beech wood was a relevant aspect of the project as Beech wood is a locally available material with large reserves in Baden-Württemberg (see [102]).

B.2.2 Principle of Finger Joint

The Principle of Finger Joints associates in-plane shear resistance with geometric and material parameters of finger joints via a step joint model.

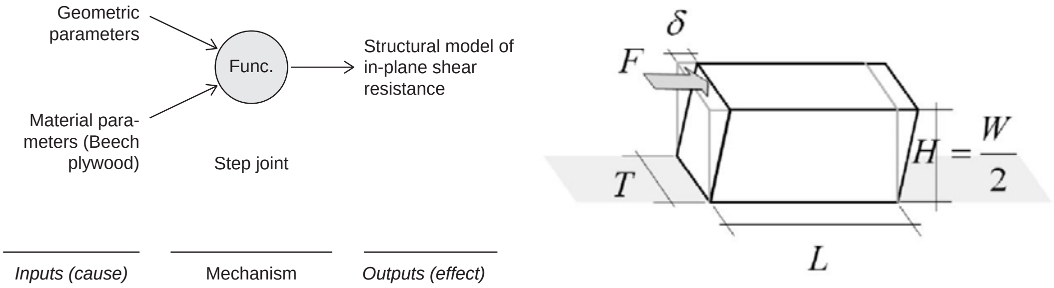


Figure 7.11: Principle of Finger Joint. Right image from [115] by Li (2015).

.

Inputs (cause) – Geometric parameters such as height and length of the joint and plate thickness. Material parameters of Beech plywood, such as shear modulus and characteristic shear strength of the plywood.

Methods (mechanisms) – Finger joints function similarly to step joints in traditional truss works. The thrust force is taken by the contact surface and transferred to the shear plane [103, p. 117].

The stiffness generated from a single finger (wooden block) is a function of shear modulus, displacement and height, length and thickness. The capacity of the finger joint in the local y-direction (the in-plane shear direction) is a function of the duration of the load and the moisture content, characteristic shear strength of the plywood, and the length and thickness of the block [115, p. 127].

Li and Knippers also showed that “the maximum membrane forces and the shear displacement in the finger joint are not strongly related to the grain direction of the surface ply” [115, p. 127].

Furthermore, “if there is a gap along the local x-direction of a connection [i.e. between two fingers], the wooden finger joint can only generate resistances after the contact takes place” (ibid.).

Outputs (effects) – An analytical structural model of the in-plane shear resistance of a finger joint; shear capacity of a finger joint.

Significance – This principle is the pre-condition for examining whether any joint exceeds its design capacity in the FE-model.

B.2.3 Principle of Crossing Screws

The Principle of Crossing Screws associates an analytical model of stiffness and capacity of a crossing screw connection with the application of crossing screws in edge-wise connected timber plates via the formation of a small truss between the plates (Fig. 7.12).

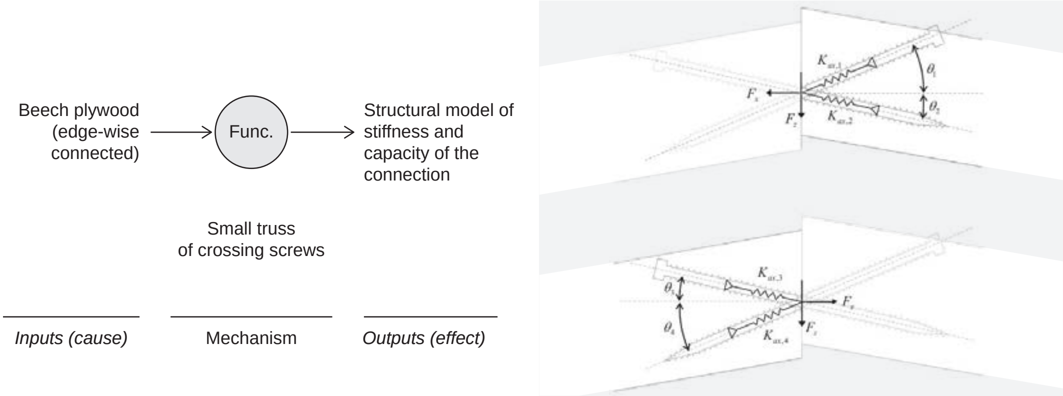


Figure 7.12: Principle of Crossing Screws. Right image from [115] by Li (2015).

Inputs (cause) – Edge-wise connected timber plates made of Beech plywood.

Methods (mechanisms) – Axial forces and out-of-plane shear are taken by crossing screws. Screws are oriented “in parallel planes, normal to the edge, with distance $d_s = 20$ mm to each other” [115, p. 126] on axis with the finger joint. In side view, the screws are inclined relative to the plate normal and intersect with a specific crossing angle that is parametrically controlled (see Principle of Joint Adaptation).

To examine the relation between stiffness and screw angles, a simplified structural model was used, where the crossing screws are treated as a two-dimensional truss structure subjected only to the axial force and the out-of-plane shear [115, p. 127]. The structural model is explained in detail in Li and Knippers [115].

Outputs (effects) – A model of the stiffness and capacity of the crossing screw joint as a function of the number and the stiffness of the screws and their withdrawing capacity.

Significance – This principle is a pre-condition for determining the withdrawing capacity through physical load tests.

B.2.4 Principle of Physical Load Testing

The Principle of Physical Load Testing associates characteristic values of the crossing screw connection with a mathematical joint model via determining the withdrawing capacity through physical load testing (Fig. 7.13).

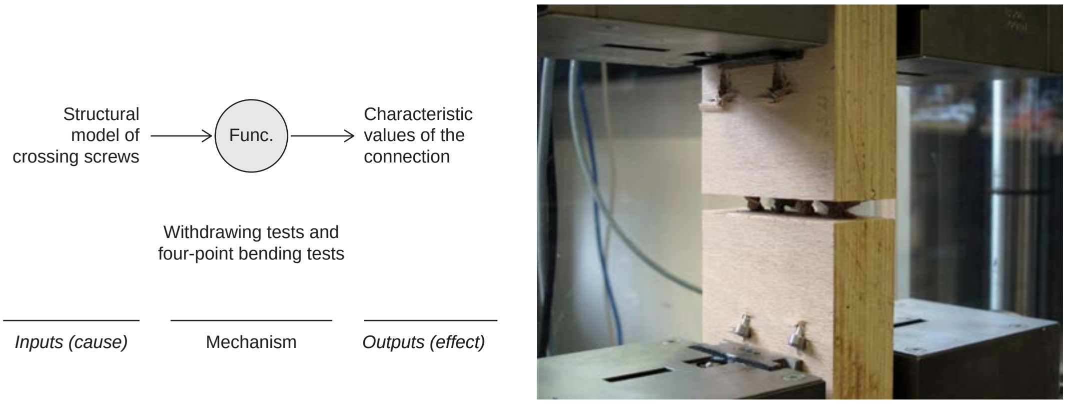


Figure 7.13: Principle of Physical Load Testing. Right image by Li (2014).

Inputs (cause) – Analytical structural model of crossing screws.

Methods (mechanisms) – A series of withdrawing tests, with various screw orientations, was conducted: Five combinations of inclining angle and sway angle of the plywood were tested using a screw with an outer diameter of 6 mm, a total length of 120 mm, and a penetration length of 60 mm. These parameters were the same as in the developed details of crossing screw joints. For each combination, five specimens were tested [115, p. 129].

Four-point-bending tests in order to determine the spring coefficient for bending resistance of a joint [103, p. 118].

Outputs (effects) – Influence of the screw orientation on the withdrawing stiffness and capacity. Li and Knippers [115, p. 129] also showed that “the characteristic capacity is not sensitive to the screw orientation in the plywood, whereas the stiffness changes with the screw orientation.”

With these characteristic values the spring coefficients for the axial resistance and the out-of-plane shear resistance can be calculated using the simplified truss model [103, p. 118] (see B.2.3). Spring coefficient for bending resistance.

Significance – The spring coefficients can then be used in the FE model to check if any connection exceeds its capacity under different load cases.

B.2.5 Principle of Joint Gaps

The Principle of Joint Gaps associates characteristic structural properties of the connection in both in-plane shear and axial load directions with gaps between the fingers and spring coefficients via Finite-Element (FE)-analysis of joint mechanism (Fig. 7.14).

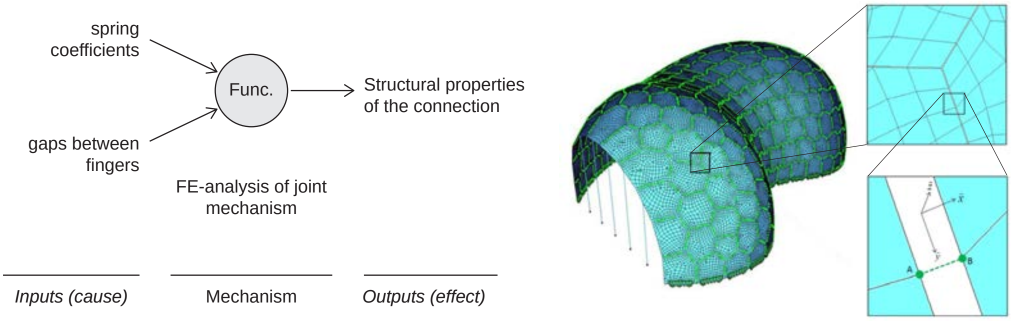


Figure 7.14: Principle of Joint Gaps. Right image adapted from [115] by Li (2015).

Inputs (cause) – 1 mm gaps between the joints in the local x and y directions based on fabrication- and assembly-related requirements; characteristic stiffness and capacity of the joint.

Methods (mechanism) – In a detailed FE-simulation, Li and Knippers [115] examined the effect of finger joints on the transfer of axial and shear forces between the plates and the role that the gaps play for the overall load-bearing capacity in a finger joint that is reinforced by crossing screws.

“In the developed finite element (FE) model, the stiffness of the connections is simulated by spring elements: every connection, which consists of a finger joint and a pair of crossing screws, is simulated by four spring elements that act on the two nodes situated at the opposite sides of two adjoining plates. Three springs are used for simulating axial resistance, in-plane shear resistance, and out-of-plane shear resistance, [...] a fourth spring is used for simulating bending resistance. [...] The internal forces of these spring elements are used for examining whether any joint exceeds the design capacity, The contour of each plate is a simple polygon; the detailed geometries of finger joints are not implied” [115, p. 133].

“The in-plane shear in a connection will be completely taken up by the crossing screws if the in-plane displacement of the connection is smaller than the gap width, but once the displacement reaches the gap width, contact occurs, and the finger joint starts to take the force.” This means that the spring coefficient in the in-plane shear direction is not constant, but rather a step function [115, p. 133].

For a detailed description of the FE-model including simulation of the plywood, failure criteria of joint and plywood, load cases and comparative analyses of different gap widths (0 mm, 1 mm, infinite) see Li and Knippers [115].

Outputs (effect) – Using FE-simulation and -analysis, the following two positive effects of finger joints were identified [115, p. 136]:

1. The high shear stiffness of finger joints limits the in-plane displacement after contact, and the maximum in-plane shear taken by a screw is thus bounded and it will not be over-loaded.
2. The high in-plane resistance of finger joints changes the force transfer in the structure so that more forces are transferred between the plates as in-plane shear forces, instead of axial forces; as a result, the axial forces, which are completely taken up by screw joints, are reduced.

The benefit of the finger joints becomes apparent only after the structure is fully loaded and the contacts of the finger joints take place [115, p. 137].

Even though the existence of gaps between panels generally reduces bending stiffness in the joints and weakens the plate shell structure, the gap between the plates prevents a lever effect in the axial force direction that might lead to the failure of the screw when contact between the plate happens due to bending. The chosen gap width of 1 mm therefore has proved to be acceptable [115, p. 138].

Significance – The gaps in the joints consequently fulfill three functions: (1) they cover the dimensional deviation caused by fabrication and moisture; (2) they facilitate assembly by creating buffer zones that allow for the positioning of the plates; and (3) they enable shear-load bearing capacity after the shell is fully loaded.

B.2.6 Principle of Higher Stiffness

The Principle of Higher Stiffness associates increased in-plane stiffness with the high stiffness of finger joints once the segmented shell is fully loaded via contact and interlocking between the joints (Fig. 7.15).

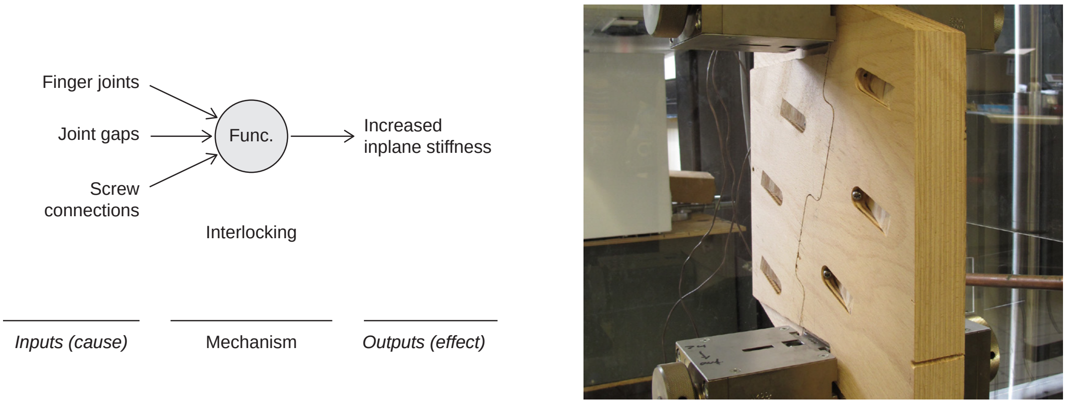


Figure 7.15: Principle of Higher Stiffness. Right image from [103] by Li (2014).

Inputs (cause) – High stiffness of finger joints compared to the stiffness of screws; screwed connection with gaps between joints.

Methods (mechanism) – Li and Knippers [115] showed that forces in trivalent plate structures are mainly transferred though plate edges in the form of in-plane shear forces. After in-plane contact in the shear direction, the high in-plane resistance of finger joints changes the force transfer in the structure so that more forces are transferred between the plates as in-plane shear forces, instead of axial forces. As a result, the axial forces, which are completely taken up by screw joints, are reduced.

Outputs (effect) – In comparison with plate structures that are joined only with screw connections, the application of finger joints can effectively increase the in-plane stiffness and thus make plate shell structures stiffer and stronger [115, p. 136].

Significance – The application of interlocking joints such as finger joints in segmented plate shells increases the performance of the structure.

B.3 Segmentation Principles

B.3.1 Principle of Triangulation

The Principle of Triangulation associates the generation of a triangulated mesh as the representation of a topological neighborhood with a series of input points on a double-curved surface via a triangulation mechanism (Fig. 7.16).

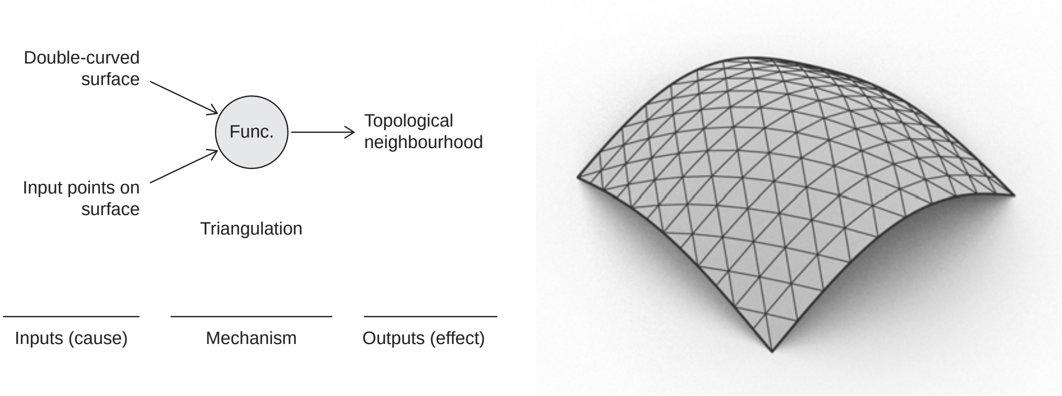


Figure 7.16: Principle of Triangulation. Right image by Schwinn (2014)

Inputs (cause) – Input points representing the location of tangent planes on a double-curved surface. In order to generate valid intersection points for each tangent plane, it is important to determine which tangent plane has to intersect with which other planes nearby.

Methods (mechanisms) – A topological neighborhood was defined by triangulating between the input points distributed across the surface. Due to the duality in TPI that was mentioned before, each vertex in the triangulation will correspond to one plate in the segmented shell and each edge in the triangulation will correspond to one sheared edge between adjacent plates. The triangulation was performed as a two-step process where an initial triangulation of the input points was computed using a Delaunay triangulation of the 2-dimensional (u, v) coordinates of the points in the parameter space of the surface [172, p. 181]. In a second step similar to Troche [192], the resultant edges of the triangulation were flipped, where necessary, based on the Cartesian distance between the input points on the surface, such that the edges of the resultant triangulation represent the shortest distances between the input points [172, p. 181].

Outputs (effect) – The result is a mesh M representing the distance-based topological relation between the input points on the double-curved surface.

Significance – The principle is the pre-condition for planar segmentation using the TPI method.

B.3.2 Principle of Planar Approximation

The Principle of Planar Approximation associates a mostly planar trivalent configuration of segments with a topological network of input points and the planarity requirement via a customization of the TPI method (Fig. 7.17).

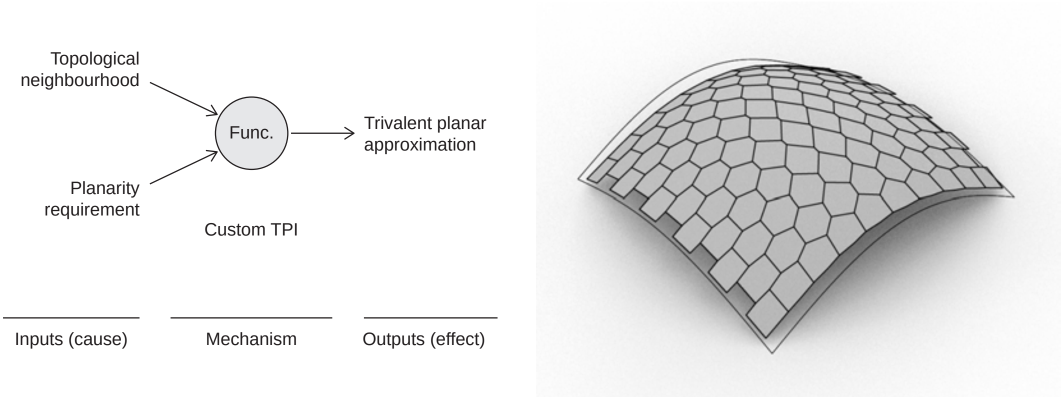


Figure 7.17: Principle of Planar Approximation. Right image by Schwinn (2014).

Inputs (cause) – The topological neighborhood of input points according to the Principle of Triangulation; planarity requirement.

Methods (mechanism) – The segmentation is generated using the TPI method, where each vertex v_i of mesh M corresponds to a planar polygon P_i in the tangent plane T_i at vertex v_i; and each mesh face F_i of M corresponds to a polygon vertex I_i calculated as the intersection of the three tangent planes at the vertices of the mesh face F_i. The resultant polygon is defined by the intersection points associated with the mesh faces adjacent to v_i.

This mechanism is equally robust on synclastic and anticlastic surfaces. In the transition zones between synclastic and anticlastic areas, however, where the Gaussian Curvature $K \approx 0$, the intersection points of nearly parallel planes might end up far from their input points, resulting in 'degenerate' or invalid polygons [172, p. 181].

An invalid polygon was subsequently defined as follows: if the projection I' of the intersection point I_i onto plane T_i of the mesh face falls outside its circumcircle C, then the intersection I_i is said to be invalid, meaning the intersection point is considered too far away in order to produce a valid polygon. In this case, the intersection point I_i was projected back onto the perimeter of C. The resultant polygonal outline was non-planar but the adjacent polygons still shared the same vertex [172, p. 182].

Outputs (effects) – Applied across the entire input mesh, this principle results in a mostly planar approximation of a double-curved surface in a structurally beneficial trivalent configuration.

Significance – This principle solves the intersection problem of TPI in areas where Gaussian curvature K approaches zero by temporarily allowing for non-planar polygons in the segmentation similar to Wang et al.[203].

B.3.3 Principle of Planarity

The Principle of Planarity associates increased planarity of individual non-planar plates with a given segmentation and the planarity requirement via a planarity check and feedback on the location of input points (Fig. 7.18).

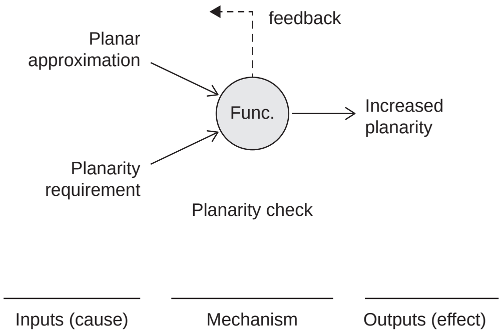


Figure 7.18: Principle of Planarity.

Inputs (cause) – Planar approximation of the input surface; planarity requirement.

Mechanism (function) – Planarity is measured by fitting a plane through the vertices the polygon and measuring the maximum distance of the points to the plane. If the distance exceed a certain tolerance value, then the plate is called non-planar.

In order to reduce non-planarity, the measure for non-planarity, i.e. the out-of-plane distance, at that point, is numerically differentiated. Moving the input point in the direction of the steepest slope (planarity change) thus moves the polygon in a gradient descent towards planarity [103, p. 113]. If non-planarity cannot be further reduced the process stops. Through iterative control of input point locations, the planarity of non-planar plates can therefore be increased.

Outputs (effect) – A segmentation with increased planarity of non-planar plates.

Significance – Planarity of plates is one of the criteria for segmentation validity.

B.3.4 Principle of Valid Plate Size

The Principle of Valid Plate Size associates valid plate sizes in the shell with a material-informed segmentation via a validity check regarding available stock material and feedback on the location of input points (Fig. 7.19).

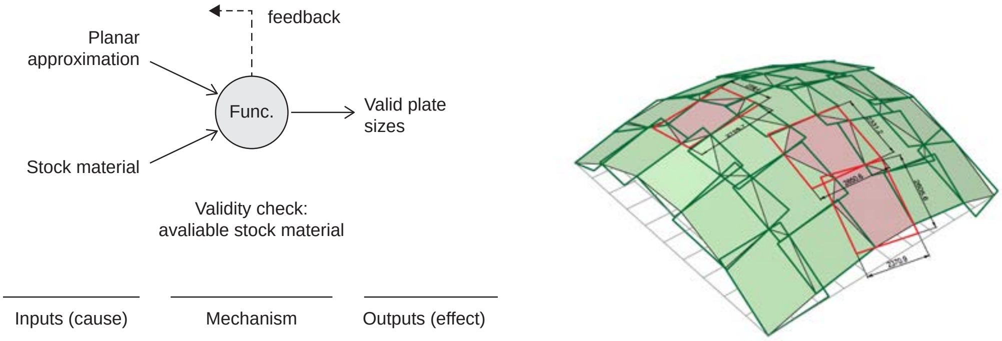


Figure 7.19: Principle of Valid Plate Size. Right image from [103] by Schwinn (2014).

Inputs (cause) – Planar approximation of the input surface; available stock material sizes.

Mechanism (function) – Plate size is measured by calculating the minimum bounding rectangle of the polygon. The edge lengths of the resulting rectangle have to be less than that of the available stock material.

The length of an edge is a function of the distance of the input points in the direction of that edge: if an edge exceeds the permissible range, decreasing the distance between input points will decrease the edge length; conversely, increasing the distance will increase the edge length of the bounding rectangle. Through iterative control of input point locations, the angles between plates can therefore be adjusted until they meet the permissible ranges that are based on the fabrication strategy [103, p. 114].

Outputs (effect) – A segmentation considered valid based on the available stock material.

Significance – Valid plate size is one of the criteria for segmentation validity.

B.3.5 Principle of Valid Plate Diameter

The Principle of Valid Plate Diameter associates valid plate diameters in the shell with a fabrication-informed segmentation via a validity check regarding robot working envelope and feedback on the location of input points (Fig. 7.20).

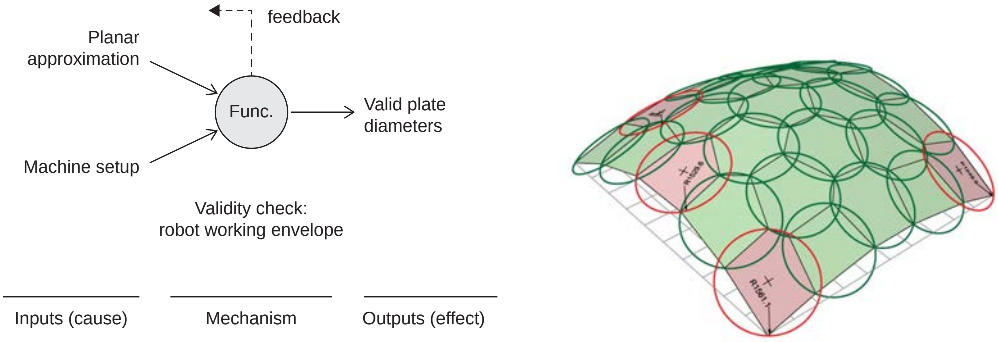


Figure 7.20: Principle of Valid Plate Diameter. Right image from [103] by Schwinn (2014).

Inputs (cause) – Planar approximation of the input surface; given machine setup for fabrication.

Mechanism (function) – Plate diameter is measured by calculating the minimum circumscribing circle of the polygon. The diameter of the resulting reference circle has to be less than the clear space available inside the robot cell; at the same, time every plate edge has to be accessible by the working point of the machining tool, which led to the choice of a vertical positioner (turntable) for reorienting the work piece.

The diameter the reference circle is a function of the distance of the input points: if the diameter exceeds the permissible range, decreasing the distance between input points that are furthest apart will decrease the diameter; conversely, if the distance between the circle centre and the closest plate edge falls below a certain minimum threshold determined by the reach of the robot, increasing the distance between input points that are closest together will result in a more evenly proportioned (less oblong) plate. Through iterative control of input point locations, the reference circle can thus be adjusted until it meets the permissible maximum and minimum distances that are based on the fabrication setup and working envelope [103, p. 114].

Outputs (effect) – A segmentation considered valid based on a given machine setup.

Significance – Valid plate diameter is one of the criteria for segmentation validity.

B.3.6 Principle of Valid Plate Angle

The Principle of Valid Plate Angle associates valid connection angles between plates in the shell with a fabrication-informed segmentation via a validity check regarding permissible connection angles and and feedback on the location of input points (Fig. 7.21).

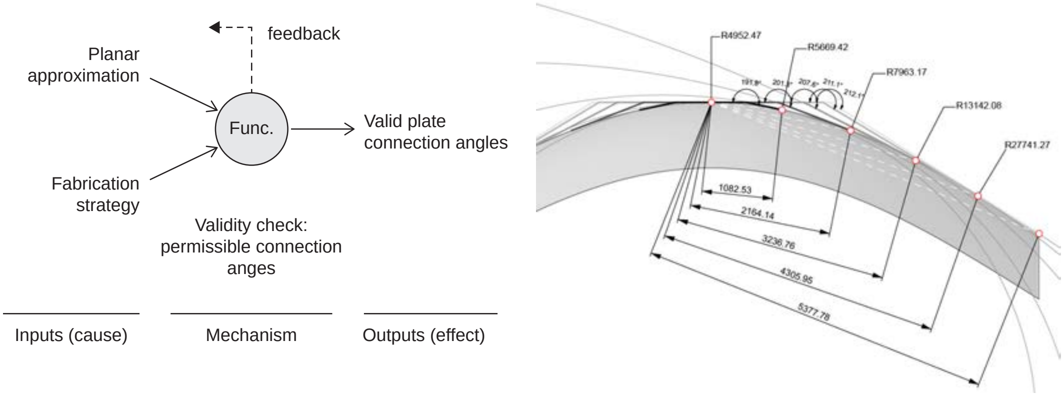


Figure 7.21: Principle of Valid Plate Angle. Right image from [172] by Schwinn (2014).

Inputs (cause) – Planar approximation of the input surface; given fabrication strategy of the connection.

Mechanism (function) – The angle between adjacent plates is a function of the distance between input points and of the amount of curvature at the input locations: decreasing the distance between input points will decrease the angle between plates; conversely, increasing the distance will increase the connection angle [172, p. 181]).

Through iterative control of input point locations, the angles between plates can therefore be adjusted until they meet the permissible ranges that are based on the fabrication strategy.

Outputs (effect) – A segmentation considered valid with respect to the fabrication strategy of the connection.

Significance – Valid plate angles are one of the criteria for segmentation validity.

B.3.7 Principle of Valid Edge Length

The Principle of Valid Edge Length associates valid edge lengths of plates in the shell with a structure-informed segmentation via a validity check regarding required minimum edge lengths and feedback on the location of input points (Fig. 7.22).

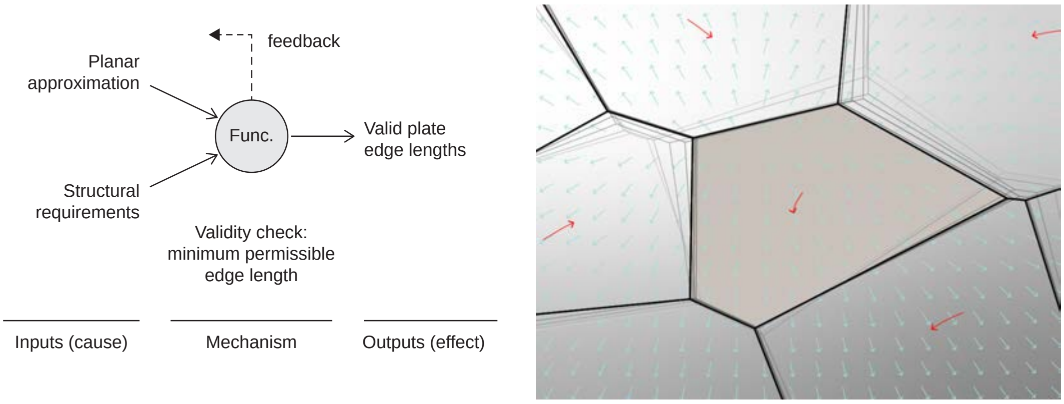


Figure 7.22: Principle of Valid Edge Length. Right image from [172] by Schwinn (2014).

Inputs (cause) – Planar approximation of the input surface; structural requirements of the connection such as the width of a finger joint.

Mechanism (function) – The length of shared edges between adjacent plates is inversely proportional to the distance between input points: decreasing the distance between input points will increase the edge length between associated plates; conversely, increasing the distance will decrease the edge length [172, p. 181].

Through iterative control of input point locations, the edge lengths between plates can be adjusted until they meet the required minimum length that is based on structural requirements.

Outputs (effect) – A segmentation considered valid based on structural requirements of the connection.

Significance – Valid edge lengths are one of the criteria for segmentation validity.

B.3.8 Principle of Segmentation Validity

The Principle of Segmentation Validity is a higher-level principle that associates a valid segmentation with a multi-criteria-informed segmentation by subsuming validity checks regarding geometry-, material-, fabrication-, and structure-related requirements (Fig. 7.23).

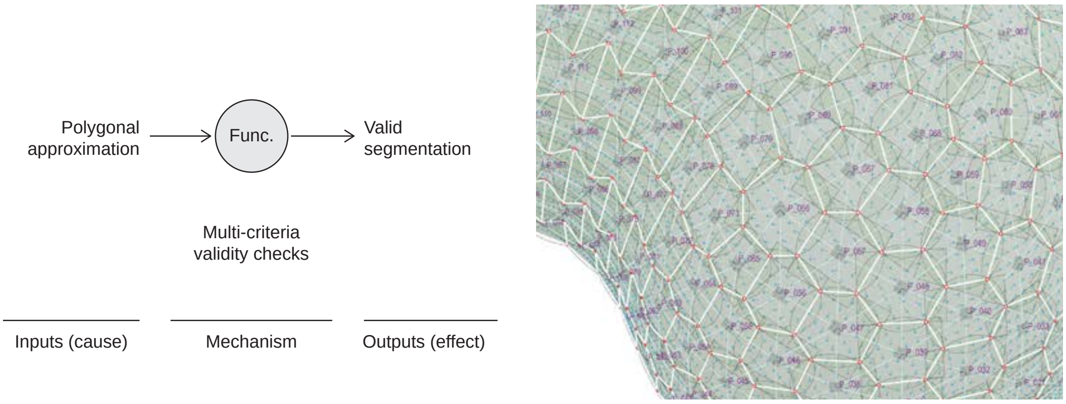


Figure 7.23: Principle of Segmentation Validity. Right image from [175] by Schwinn (2015).

Inputs (cause) – Planar approximation of the input surface; multiple potentially conflicting performance criteria.

Mechanism (function) – In order to achieve a segmentation that meets multi-criteria performance objectives, a negotiation of the different strategies pursued in the low-level principles is required.

Averaging translation vectors that are based on the low-level strategies is not an option as some strategies might cancel each other out (see [158]. Instead, a prioritized approach has been chosen that favors increasing planarity first and then pursues individual plate strategies by weighting translation vectors [172, p. 182].

Outputs (effect) – A segmentation considered valid with respect to multiple performance criteria.

Significance – This principle is a prerequisite for achieving a performance-oriented segmentation of a shell surface.

B.3.9 Principle of Planarization

The Principle of Planarization associates a planar segmentation with a valid planar approximation of a double-curved surface via a minimization mechanism (Fig. 7.24).

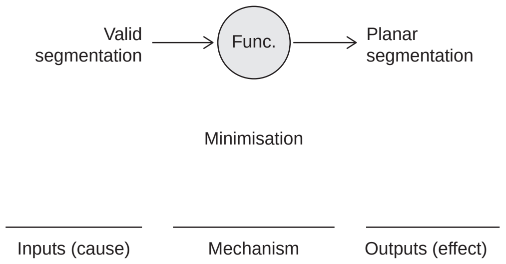


Figure 7.24: Principle of Planarization.

Inputs (cause) – Valid planar approximation of a double-curved surface following the Principle of Segmentation Validity.

Methods (mechanisms) – Due to the many system constraints, most notably the plate tangency constraint, the model might fail to converge to a stable solution in the parabolic areas. Therefore, in order to expedite the generation of a valid solution, a post-process was implemented that achieves planarity through an optimization method, minimizing non-planarity, similar to Wang et al. [203]. As a trade-off the resultant plate slightly deviates from the tangent plane at the corresponding vertex in the mesh dual [172, p. 183].

Outputs (effect) – The result is a valid planar segmentation also in areas in which the TPI method typically fails.

Significance – The combination of the Principles of Triangulation, Planar Approximation, Segmentation Validity, and Planarization provides an approach for the interactive manipulation of the segmentation pattern by changing the positions of the input points on the surface or by changing the input surface.

B.4 Structural Segmentation Principles

B.4.1 Principle of Trivalence

The Principle of Trivalence associates the kinematic stability of a segmental plate shell with hinged connections and a trivalent segment layout via geometric constraints between the plates that lock plate edges (Fig. 7.25).

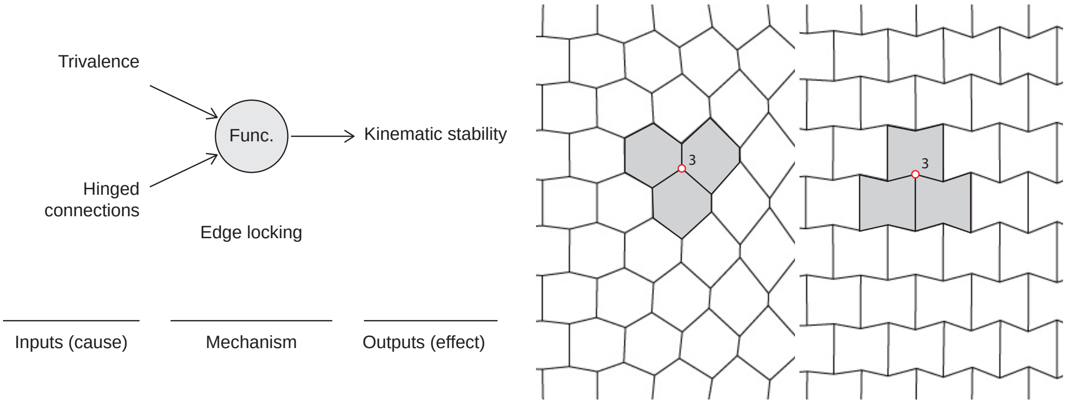


Figure 7.25: Principle of Trivalence.

Inputs – Trivalent plate layout; hinged connections.

Methods (mechanisms) – "For a plate structure with a trivalent geometric pattern, [where] three plates meet at one point andare hinged at the three intersection lines, each plate is constrained by the other two. Thus bending stiffness is generated even though the joints themselves are only hinged connections" [114, p. 3] (see also [208; 109]. In order to estimate the bending stiffness a simplified mathematical model is introduced. The model is described in detail in Li and Knippers [114].

Results (outputs) – It was shown how the bending stiffness in the shell depends on the angle between consecutive edges. The model also showed that hexagonal patterns have similar bending stiffness in both principle pattern directions regardless if the shape of the polygon is convex or concave as they occur in synclastic ($K > 0$) and anticlastic ($K < 0$) surface areas. The results of the model have been confirmed using FE-analysis [114, p. 7].

Significance – The type and direction of segment pattern has no effect on the bending resistance in the structure.

B.4.2 Principle of Co-Linearity

The Principle of Co-linearity associates zero local bending stiffness of hinged joints in a trivalent plate layout with co-linear consecutive edges and bending in the structure via the hinge mechanism (Fig. 7.26).

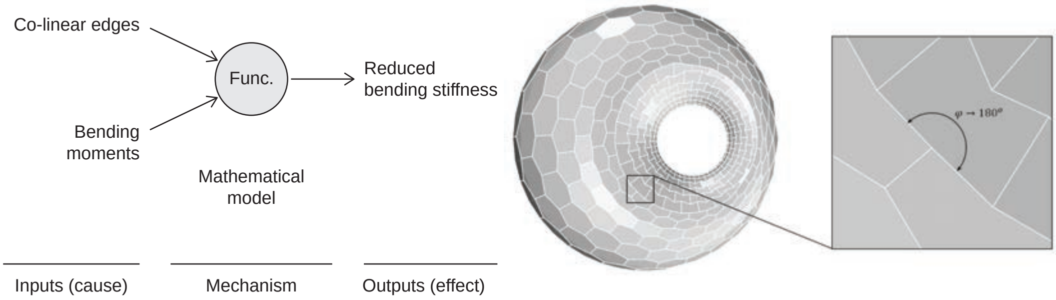


Figure 7.26: Principle of Co-Linearity. Right: Trivalent pattern at the transition zone, where the included angle approaches 180 degrees. From [114] by Li (2015).

Inputs (cause) – Bending moments acting on bending-weak joints; co-linear edges. In segmentations that are generated using TPI, this condition occurs in areas of zero Gaussian curvature – the transition zone between synclastic and anticlastic surface areas.

Methods (mechanism) – Co-linear edges are a special case where two consecutive edges in a polygon become nearly parallel, meaning that the vertices that define those edges almost lie on one line. Using an analytical approach, Li and Knippers [114, p. 10] showed that in this case the bending stiffness generated through geometric locking approaches zero and the rotation around the joint axis becomes too large to be acceptable. Maximum allowable rotation between two plates is set as 25 mrad.

Outputs (effect) – Reduced bending resistance of a plate shell in cases where the angle between consecutive edges in the segment pattern approaches zero.

Significance – As this condition should be avoided, two options are implied: either avoiding co-linearity between edges, although this might not always be possible, or using additional means of reinforcement, such as crossing screws.

B.4.3 Principle of Generative FE-Modeling

The Principle of Generative FE-Modeling associates a detailed FE-model for structural analysis with specific load cases, a validated joint model, and a segmented shell model via a generative structural modeling approach (Fig. 7.27).

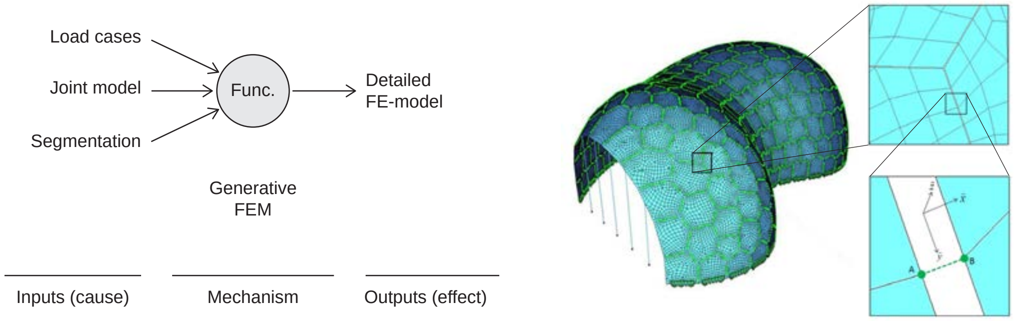


Figure 7.27: Principle of Generative FE-modeling. Right: Global FE-model of the pavilion. Adapted from [115] by Li (2015).

Inputs (cause) – Load cases defined by building code based on the requirements for permanent buildings; a validated joint model as developed in the Principle of Joint Gaps; and a geometrically differentiated B-rep model of a segmented shell, such as the outcome of the segmentation principles.

Mechanism (function) – A custom digital workflow was implemented that automatically generated structural models from trivalent, polygonal B-rep surface models [103, p. 117]. This allowed to (1) directly link structural analysis to the design model, which was iteratively developed using feedback processes and therefore subject to many changes; (2) to automatically vary the configuration of connections in order to run parameter studies; and (3) to find locally optimal solutions (Krieg et al. 2015, 117). The workflow included the discretization of the plate model using specific meshing algorithms, the generation of shell elements and springs connecting opposing edges between the plate segments, the definition of support conditions and, finally, load assumptions. With timber being a relatively light-weight material and given the requirements for permanent buildings, the critical load cases were combinations of unsymmetrical snow loads and the wind loads [115, p. 125].

Outputs (effect) – A detailed FE-model used for structural optimization and subsequent comparative analyses.

Significance – Generative FE-modeling enables automatic variation of model parameters as a precondition for subsequent evaluation (and selection). In this case it is the pre-condition for evaluating the effect of adding a small bending stiffness in the joints and for comparing stress distribution as a function of the principal orientation of the segment pattern.

B.4.4 Principle of Added Bending Stiffness

The Principle of Added Bending Stiffness associates increased bending resistance of a structure with locally bending-weak joints and crossing screws via increased bending stiffness in the joints (Fig. 7.28).

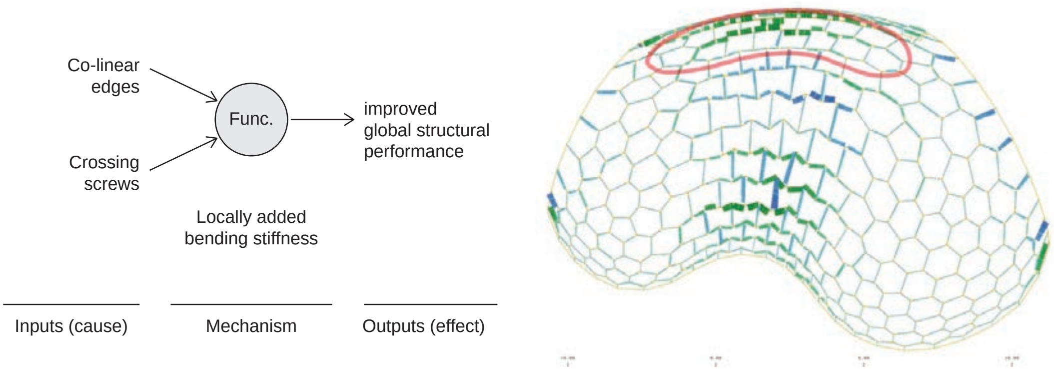


Figure 7.28: Principle of Added Bending Stiffness. Right: Rotations between plates considering the bending stiffness of screwed joints. From [114] by Li (2015).

Inputs (cause) – Locally bending-weak joints; crossing screws.

Methods (mechanisms) – With regards to the relationship between kinematic stability and segment layout, Li and Knippers [114] showed that the trivalent segment pattern does provide sufficient bending stiffness – with the exception of the transition zone between areas of positive and negative Gaussian curvature where consecutive plate angles are close to 180°. As shown in the Principle of Co-linearity, the bending stiffness generated from the trivalent geometry disappears in the transition zone, and the local rotation between plates become too large to be acceptable. By introducing a small bending stiffness provided by crossing screw joints the rotations between the plates in the transition zone become acceptable [114, p. 11].

A subsequent comparison of rotation values with and without bending stiffness showed that, with the exception of the transition area, the rotations between plates due to bending are very similar. “This fact indicates that the bending stiffness provided by the trivalent geometry is much larger than that provided by the screwed joints” [114, p. 11].

Outputs (effect) – Even though the bending stiffness provided by the trivalent geometry is much larger than the screwed connection, adding a relatively small bending stiffness significantly increases the stability of the structure.

Significance – This principle is particularly relevant in areas of zero Gaussian curvature.

B.4.5 Principle of Force Alignment

The Principle of Force Alignment associates a more uniform stress distribution in a trivalent pattern with a given loading via the alignment of the outer edges of plate polygons parallel to the principal load directions (Fig. 7.29).

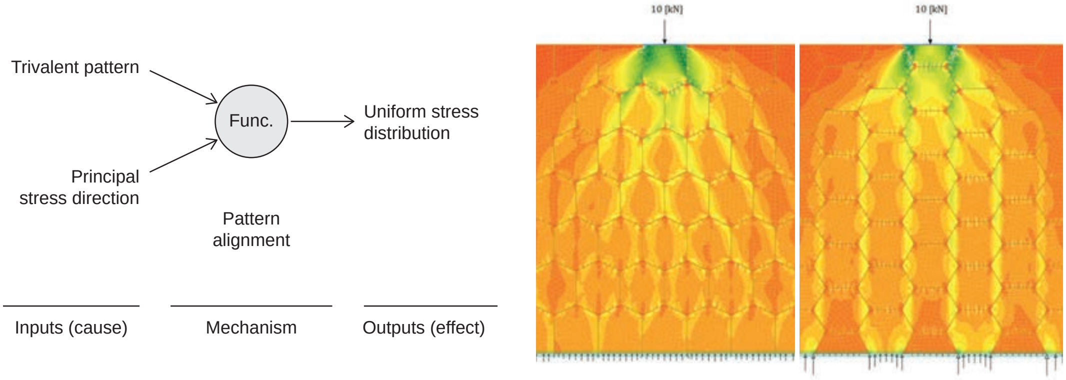


Figure 7.29: Principle of Force Alignment. Right: Stresses and reaction forces in aligned pattern orientation. From [114] by Li (2015).

Inputs – Load case; trivalent segment pattern.

Methods (mechanisms) – According to Li and Knippers [114, p. 7] "geometric pattern and joint stiffness [...] determine where and how the internal forces are transferred in segmental plate shells." The authors built four FE-models to investigate the force transfer in both convex and concave patterns in both major pattern directions. "Each model is applied with a downward force on the top to observe how the force is transferred in the segmental elements" (ibid.).

Regarding the relevance of the pattern orientation, FE-simulations showed that a plate pattern that is aligned with the main load direction (where two of the six hexagon edges are parallel to the main load direction) better distributes the downward load than a pattern where the polygon edges are oriented perpendicular to the main load direction. Even though the bending stiffness of the four patterns is the same, the stresses and reaction forces at the supports are more uniform if the pattern is aligned. Similar behaviors can be observed in the concave, bowtie-shaped patterns. [114, p. 7].

Results (outputs) – Segment patterns that are aligned with the principle load directions better distribute the load leading to a more uniform stress distribution and reaction forces at the supports.

Significance – Edges should be oriented parallel to the principal load directions.

B.5 Shell Form-finding Principles

B.5.1 Principle of Edge Control

The Principle of Edge Control associates form-adapted design surface boundaries with architectural design inputs and programmatic requirements via interactive manipulation of a 3-dimensional reference curves by the designer (Fig. 7.30).

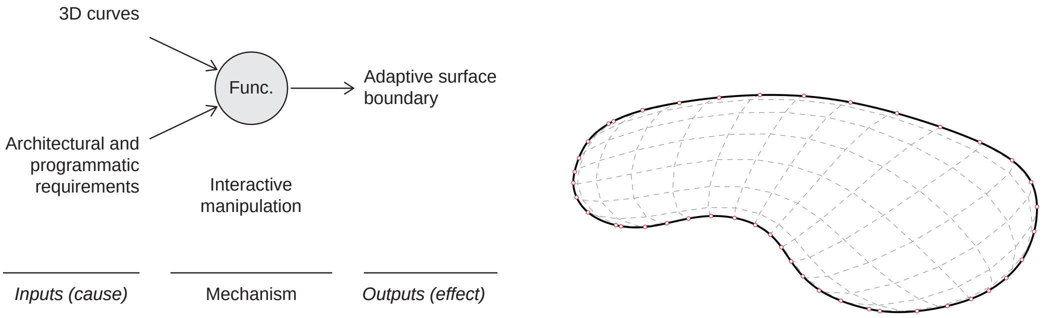


Figure 7.30: Principle of Edge Control.

Inputs (cause) – Architectural and programmatic requirements such as desired views or usable floor area to be covered by the shell.

Mechanism (function) – By interactively adjusting the control points of a boundary curve, which acts as a guide for the boundary of a reference mesh during form-finding, the designer can control the spatial outcome of the form-finding process in order to meet architectural and programmatic requirements. Similar to the corresponding principle in CS-I, particles that are located at the perimeter of a 2-dimensional reference mesh are to be mapped to the 3D boundary curve via PS-based mapping. The points that were mapped to the boundary curve are then allowed to slide along the curves in order to equalize strain in the springs along the mesh boundary.

Outputs (effect) – An adaptive boundary curve as a controller that allows adapting the reference mesh, and subsequently the design surface, to architectural and programmatic requirements.

Significance – The principle provides a controller for the interactive manipulation of a physics-based form-finding process and is a pre-condition for the principles of Dynamic Equilibrium and Steering of Form.

B.5.2 Principle of Local Controllers

The Principle of Local Controllers associates force vectors representing attraction or repulsion forces as part of a physics-based simulation with architectural and programmatic inputs via interactive manipulation of position, orientation, and weighting of reference geometry (Fig. 7.31).

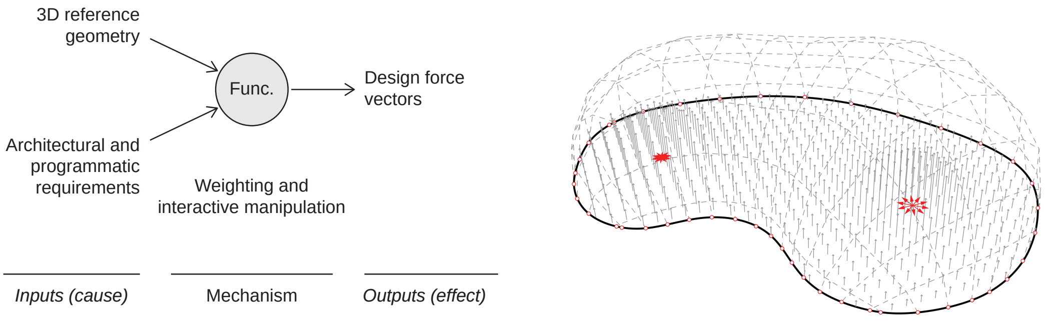


Figure 7.31: Principle of Local Controllers.

Inputs (cause) – Architectural and programmatic requirements that can be represented as design forces in simulation of physical form-finding; reference geometry in 3D space, such as points and lines.

Mechanism (function) – 3-dimensional reference geometry provides local controllers to integrate architectural and programmatic design inputs into the form-finding process. By interactively manipulating position, orientation, and weighting of the controllers, architectural context, such as site and views, and program can be represented as design forces (vectors) that act on particles in the mesh as part of a dynamic equilibrium method [172, p. 185].

Outputs (effect) – Vectors, representing origin, direction and magnitude of forces (design drivers) that act on particles in the PS-method.

Significance – The principle provides controllers for interacting with a physics-based form-finding process and is a pre-condition for the principle of Dynamic Equilibrium and Steering of Form.

B.5.3 Principle of Dynamic Equilibrium

The Principle of Dynamic Equilibrium associates a form-found, double-curved shell surface in dynamic equilibrium with an adaptive user-defined boundary curve, user-defined attraction and repulsion forces representing architectural and programmatic design drivers, and the structural requirement for double curvature via a dynamic equilibrium approach, such as particle-spring or dynamic relaxation 7.32).

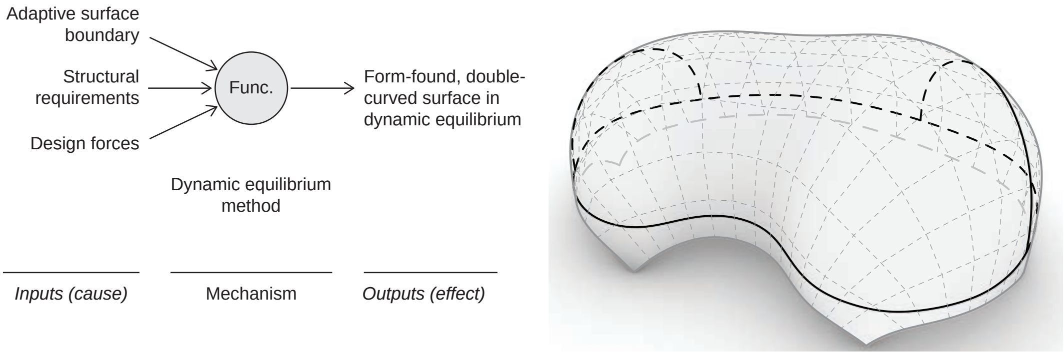


Figure 7.32: Principle of Dynamic Equilibrium.

Inputs (cause) – User-defined boundary curve; user-defined attraction and repulsion force vectors; structural requirement for double-curvature.

Mechanism (function) – Physics-based force simulation is used for the development of a custom digital design approach, integrating PS-based form finding, as well as additional external, force-driven design inputs that can integrate boundary conditions such as an architectural context and program [172, p. 185]. Similar to the corresponding principle in CS-I (A.3.3), every edge in a reference mesh is converted into a spring, assigned a stiffness value and a minimum rest length. With the application of a nominal mass of the nodes (particles), of a unary force representing negative gravity, and of the additional design forces, the relaxation process transforms the 2D mesh into a form-found 3D shape where all mesh edges relax to approximately the same edge length.

Outputs (effect) – A structurally informed, double-curved, form-found design surface incorporating additional design inputs and requirements.

Significance – This principle provides the mechanism for the principle of Steering of Form (B.5.5) and is the precondition for adapting the shell to design drivers other than gravity and self-weight.

B.5.4 Principle of Double Curvature

The Principle of Double Curvature associates structural stability of a thin shell surface and TPI conformity with a double–curved surface via the predominance of membrane forces in the shell surface (shell action) and bending stiffness, which avoids buckling in the case of compressive stresses (Fig. 7.33).

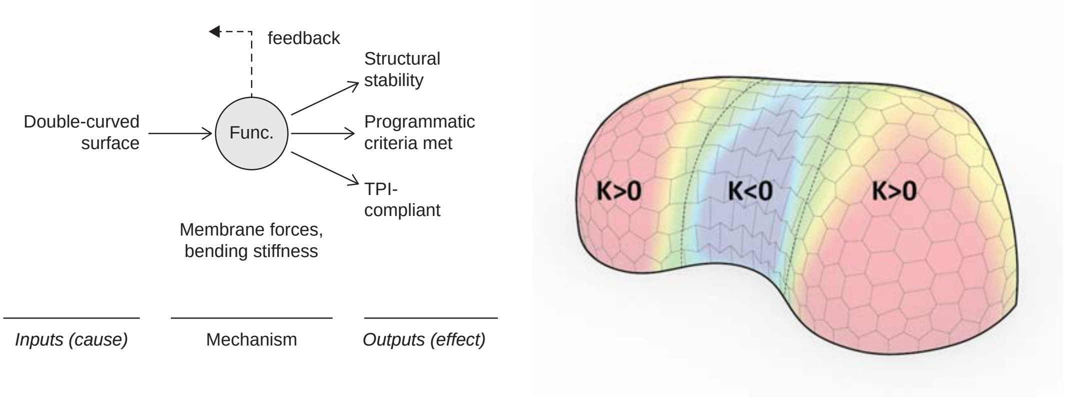


Figure 7.33: Principle of Shell Action. Right image from [172] by Krieg (2014).

Inputs (cause) – A structurally informed, double-curved, form-found design surface incorporating additional design inputs and requirements.

Mechanism (function) – Double curved surfaces inherently perform better than surfaces with Gaussian curvature approaching 0. This applies for structural optimization as well as for individual plate geometry in the plate structure system itself, which in the case of TPI directly depends on the local curvature [172, p. 185].

In order to meet architectural, geometric, programmatic and structural objectives, a feedback mechanism is provided that allows the designer to evaluate and react to the specific outcome of the form-finding process by manipulating reference geometry as part of local controllers and edge control principles.

Outputs (effect) – A computationally form-found design surface that integrates the system's requirements for double-curved geometry [172, p. 184] and meets multi-disciplinary objectives.

Significance – The principle is the precondition for the application of an agent system as the resulting surface provides the environment for the agent system.

B.5.5 Principle of Steering Form

The Principle of Steering Form is a high-level principle that associates a form-found double-curved surface with multiple performance criteria via a feedback process (Fig. 7.34).

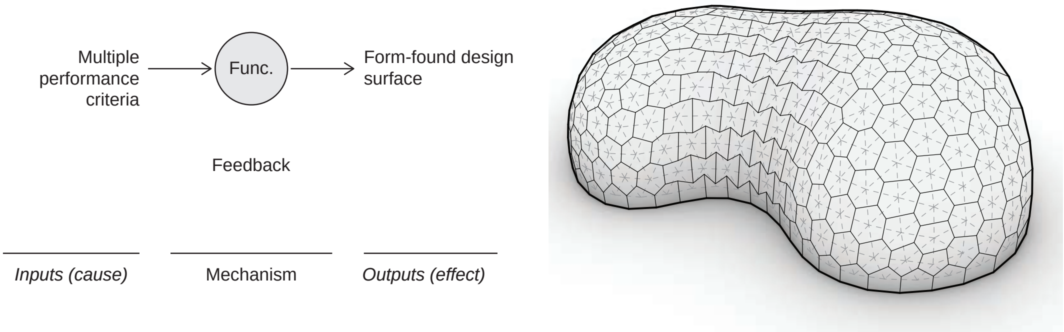


Figure 7.34: Principle of Steering Form.

Inputs (cause) – Structural and TPI-related requirements for double-curved surfaces; architectural and programmatic requirements; mechanisms in the principles of edge control, local controllers and dynamic equilibrium.

Mechanism (function) – With timber being a relatively light-weight building material, self-weight is not the decisive load case. Therefore, in this case, a form-finding approach that yields funicular shapes based purely on self-weight (like the hanging chain) was not required [115, p. 125]. Using physics-based force simulation, a custom digital design approach was developed in order to integrates PS-method and additional external design inputs (see B.5.3).

In the developed approach, the designer empirically and iteratively adapts the global geometry to local plate parameters and vice versa [172, p. 185], while the tool pursues a force equilibrium between top-down design inputs and catenary form (ibid.).

Outputs (effect) – An interactive computational approach for steering form towards a structurally informed global geometry while also responding to top-down design inputs [172, p. 185].

Significance – “This method shows particular advantages for construction systems that do not depend on compression-only geometries, such as 3-valent plate structures, and widely extends the design space while still pursuing a structurally informed solution at all times during the simulation” [172, p. 185].

B.6 Principles of Fabrication Development

B.6.1 Principle of Generative CAM

The Principle of Generative CAM associates the automatic generation of machine tool paths with a geometrically differentiated B-rep design model including topology information and job-specific parameter settings via a generative modeling approach (Fig. 7.35).

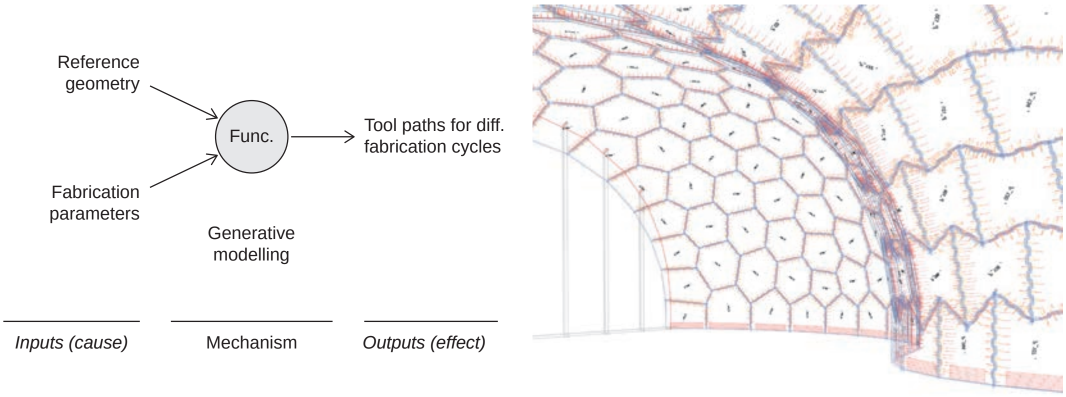


Figure 7.35: Principle of Automated CAM. Right: View of the fabrication model (Krieg, 2014).

Inputs (cause) – Topological and geometric representation of the plate arrangement as Boundary Representation (B-rep) model; geometric attributes and user-defined fabrication parameters.

Mechanism (function) – Crossing screw angles, the number of finger joints and plate assembly vectors were a function of local, edge-specific geometric attributes, such as the connection angle between two adjacent plates and the length of their shared edge [103, p. 120].

User parameters included material thickness, joint width and depth, and tolerances that were defined globally. Specifically, the joint width was set to allow for the space for the crossing screws and thus to meet requirements of building code with respect to material overlap.

The fabrication model with its various tool paths for finger joint fabrication could subsequently be generated automatically for each individual plate edge in the plate structure using trigonometric functions without the need for any further geometric information (see [103, p. 119] and [172, p. 184]).

Outputs (effect) – Tool paths for (1) cutting of the 3-dimensional finger joint contour through consecutive roughing and finishing steps with the shaft of a 20 mm diameter milling tool; (2) auxiliary features such as pockets for the localization of crossing screws; (3) drilling of pilot holes for the crossing screws; and (4) spot facing for planing the teeth of the finger joints in synclastic surface areas where they protruded from the surface of the shell and would thus have interfered with the thermal insulation layer [172, p. 184].

Significance – This principle is the pre-condition for the economical realization of segmented shells due to the large number of geometrically unique building elements and, in this case, the relative novelty of edge-wise connected wood plates, for which no standardized fabrication cycles existed.

B.6.2 Principle of 7-Axis Robotic Fabrication

The Principle of 7-Axis Robotic Fabrication associates a kinematic model of a 7-axis robotic fabrication process with a specific machine setup via a mathematical model of the robot kinematics (see A.5.5).

B.6.3 Principle of Process Validity

The Principle of Process Validity associates the validity of a fabrication process with specific fabrication strategies and a given fabrication setup via kinematic simulation of the process (Fig. 7.36).

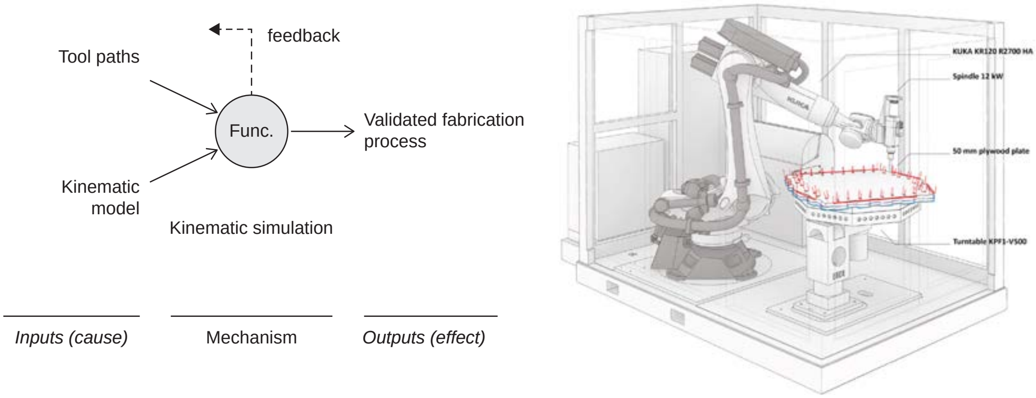


Figure 7.36: Principle of Process Validity. Right image from [172] by Krieg (2014).

Inputs (cause) – Previously generated tool paths; kinematic model of a 7-axis robot system enclosed in a cell with interior dimensions of 3200 x 2100 x 2500 mm (LxWxH) consisting of the models of a 6-axis robot arm (KR120 R2500 Quantec), a milling spindle mounted on the robot flange, an additional external revolute axis used as vertical positioner (turntable); and an automatic tool changer.

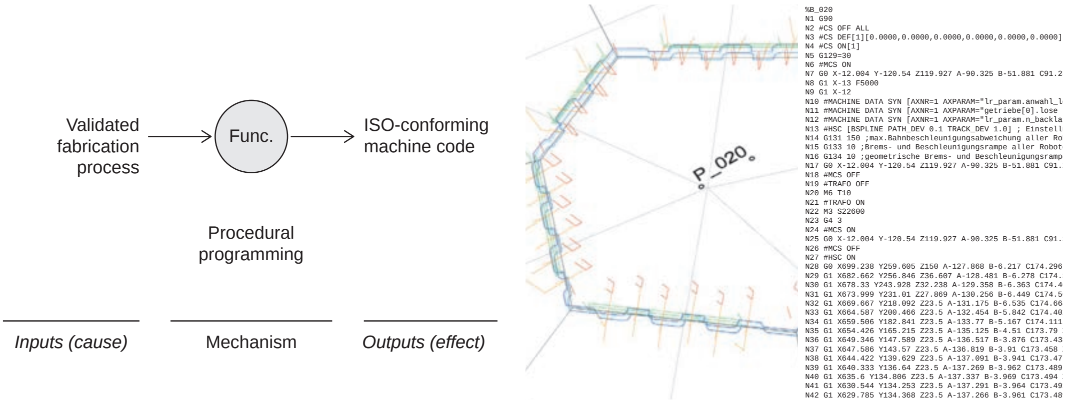
Mechanism (function) – In order to visualize the movement of the 7-axis robot system during the fabrication process, the tool paths, in combination with a kinematic model of the robot cell, were used for the simulation of the robot kinematics within the CAD-environment. This allowed determining if any collisions of the arm with itself or with the environment would occur; and if the robot would encounter any singularities along the path (where Cartesian coordinates cannot be reverse-transformed into axis positions) or out-of-reach positions. If necessary, the robot pose and table orientation could be adjusted interactively.

Outputs (effect) – A fabrication process validated within the ´production-immanent’ design environment.

Significance – The simulation allowed fine-tuning of the robotic fabrication process to ensure that all of the plates could be produced as efficiently as possible on the given machine setup [102, p. 137]. Validation of the fabrication process is the pre-condition for generating valid machine code; the mechanism can be also used to define the Machinic Morphospace associated with a specific fabrication setup.

B.6.4 Principle of Machine Code Generation

The Principle of Machine Code Generation associates the automatic generation of machine code with the tool-paths of a previously validated process via the translation of geometric and alphanumeric attributes into machine-readable text (Fig. 7.37).



```
%B_020
N1 G90
N2 #CS OFF ALL
N3 #CS DEF[1][0.0000,0.0000,0.0000,0.0000,0.0000,0.0000]
N4 #CS ON[1]
N5 G129=30
N6 #MCS ON
N7 G0 X-12.004 Y-120.54 Z119.927 A-90.325 B-51.881 C91.2
N8 G1 X-13 F5000
N9 G1 X-12
N10 #MACHINE DATA SYN [AXNR=1 AXPARAM="lr_param.anwahl_l
N11 #MACHINE DATA SYN [AXNR=1 AXPARAM="getriebe[0].lose
N12 #MACHINE DATA SYN [AXNR=1 AXPARAM="lr_param.n_backla
N13 #HSC [BSPLINE PATH_DEV 0.1 TRACK_DEV 1.0] ; Einstell
N14 G131 150 ;max.Bahnbeschleunigungsabweichung aller Ro
N15 G133 10 ;Brems- und Beschleunigungsrampe aller Robot
N16 G134 10 ;geometrische Brems- und Beschleunigungsramp
N17 G0 X-12.004 Y-120.54 Z119.927 A-90.325 B-51.881 C91.
N18 #MCS OFF
N19 #TRAFO OFF
N20 M6 T10
N21 #TRAFO ON
N22 M3 S22600
N23 G4 3
N24 #MCS ON
N25 G0 X-12.004 Y-120.54 Z119.927 A-90.325 B-51.881 C91.
N26 #MCS OFF
N27 #HSC ON
N28 G0 X699.238 Y259.605 Z150 A-127.868 B-6.217 C174.296
N29 G1 X682.662 Y256.846 Z36.607 A-128.481 B-6.278 C174.
N30 G1 X678.33 Y243.928 Z32.238 A-129.358 B-6.363 C174.4
N31 G1 X673.999 Y231.01 Z27.869 A-130.256 B-6.449 C174.5
N32 G1 X669.667 Y218.092 Z23.5 A-131.175 B-6.535 C174.66
N33 G1 X664.587 Y200.466 Z23.5 A-132.454 B-5.842 C174.40
N34 G1 X659.506 Y182.841 Z23.5 A-133.77 B-5.167 C174.111
N35 G1 X654.426 Y165.215 Z23.5 A-135.125 B-4.51 C173.79
N36 G1 X649.346 Y147.589 Z23.5 A-136.517 B-3.876 C173.43
N37 G1 X647.586 Y143.57 Z23.5 A-136.819 B-3.91 C173.458
N38 G1 X644.422 Y139.629 Z23.5 A-137.091 B-3.941 C173.47
N39 G1 X640.333 Y136.64 Z23.5 A-137.269 B-3.962 C173.489
N40 G1 X635.6 Y134.806 Z23.5 A-137.337 B-3.969 C173.494
N41 G1 X630.544 Y134.253 Z23.5 A-137.291 B-3.964 C173.49
N42 G1 X629.785 Y134.368 Z23.5 A-137.266 B-3.961 C173.48
```

Figure 7.37: Principle of Machine Code Generation. Center-right image from [172] by Krieg (2014).

Inputs (cause) – Tool paths of validated fabrication process.

Mechanism (function) – Custom programmed procedures were used to translate tool-path vertices and tool orientation vectors from the fabrication model including additional machine information, such as feed and spindle speed, into structured, machine-readable text according to specific formatting conventions. These conventions were based on the technology package KUKA.CNC, which offered milling-specific functionality that is not available when programming KUKA robots in KUKA Robot Language (KRL).

Outputs (effect) – ISO-compliant machine code (ISO 6983). One fabrication file was generated per plate containing the NC-code for all four milling job cycles described above. The instructions could be read directly by the robot controller (KR C4) and executed without the need for further post-processing and file format translations [172, p. 184].

Significance – Fabrication information could be generated for all construction layers of the plate structure in all stages of production, including the xml-based fabrication data for cutting stock pieces, wood fibre insulation and Larch plywood cladding layers on a Hundegger Speed-Panel-Machine, ISO-compliant machine code for robot, and the fabrication data for water-jet cutting the waterproofing EPDM layer [172, p. 187]. Automating NC-code generation is a precondition for one-off and small batch production paradigms (see also [15, p. 121]).

B.6.5 Principle of Machinic Morphospace

The Principle of Machinic Morphospace associates the space of producible form with a validated kinematic simulation and given materials via the simulation of the robot kinematics (Fig. 7.38).

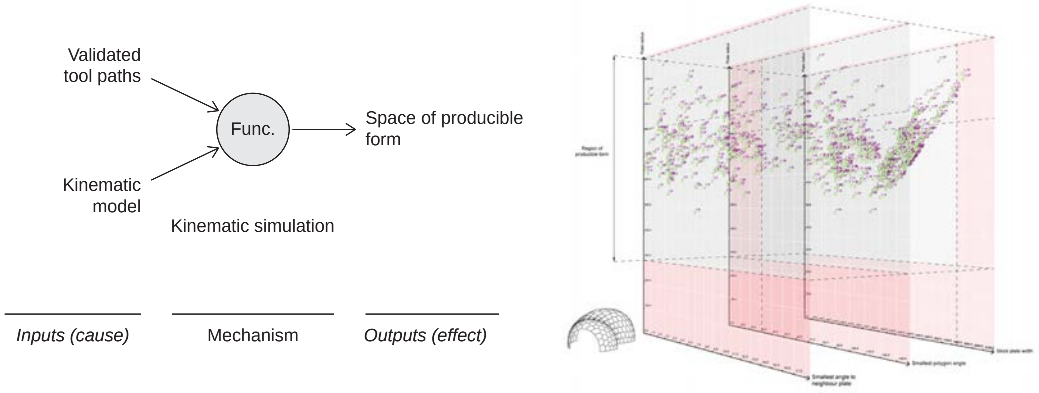


Figure 7.38: Principle of Machinic Morphospace. Right image from [175] by Schwinn (2015).

Inputs (cause) – Validated tool paths; kinematic model of machine setup.

Mechanism (function) – The robot setup defined many fabrication parameters, such as size of the workspace, degrees of freedom of machine kinematics, effector dimensions, and base position and orientation. Simulating the robot kinematics identified the limits of the robotic workspace as the space free of collisions, singularities, and out-of-reach positions.

Considering the space taken by the robot and the tool changer, this defined the minimum and maximum angles between plates and, most notably, the maximum possible plate diameter as a function of the clear width of the robot cell. The limits of this space were further delineated by the parameter ranges defined by the chosen material.

Each geometric parameter of a geometrically differentiated 3-dimensional building element, such as plate radius, stock material length, or minimum and maximum connection angles, corresponded to a dimension in a n-dimensional solution space. Each individual building element can thus be represented as an n-dimensional point [175, p. 95].

Outputs (effect) – A Machinic Morphospace delineating the space of producible geometry, a sub-set of the geometrically possible space, as a function of the specific fabrication setup.

Significance – The principle is the pre-condition for strategically exploring the space of producible form with respect to particularly performative geometry.

B.7 Quality Control Principles

B.7.1 Principle of Fabrication Accuracy

The Principle of Fabrication Accuracy associates the effective accuracy of a fabrication process with the given fabrication setup and the material being machined via the elasticity of the equipment, for example the robot arm, the calibration accuracy of tool and base, and the wear of the tool (Fig. 7.39).

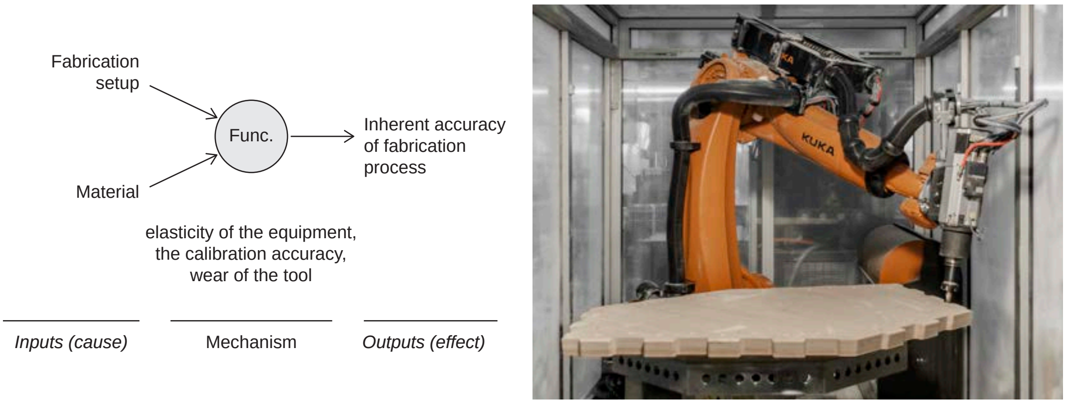


Figure 7.39: Principle of Fabrication Accuracy. Right image from [103] by Krieg (2014).

Inputs (cause) – Given fabrication setup as described in (B.8.1) Principle of Robotic Fabrication; the material to be processed in this case Beech plywood.

Mechanism (function) – The industrial robot arm is a cantilever with a characteristic stiffness. Applying loads on the Tool Center Point (TCP) will invariably cause a slight deflection of the arm. The fabrication accuracy can be improved by limiting the reach of the cantilever during fabrication and by limiting the forces that act on the arm. Forces are a function of the milling parameters spindle speed, feed, and cutting depth, which is the reason for the final finishing pass of the tool that removes only 1 mm of material. The theoretical limit of accuracy is the repeatability accuracy of the arm, specified in this case as ±0.05 mm [105, p. 9]). The resultant accuracy is affected by “slight dimensional changes in the material as well as rough surfaces due to the milling process [which] might also have led to deteriorated deviations” [103, p. 123] as well as the measuring accuracy of the equipment used for measuring (in this case 0.106 mm [169, p.4]).

Outputs (effect) – The inherent maximum accuracy of a fabrication process that is a function of setup and material.

Significance – The accuracy of a fabrication process is an important aspect that has repercussions on the structural behavior and, in the case of segmented shells, on the viability of a proposed structure.

B.7.2 Principle of Fabrication Tolerance

The Principle of Fabrication Tolerance relates tolerance specifications for plates to structural and assembly requirements via a trade-off between assembly requirements (larger gaps) and global shell stiffness (smaller gaps) (Fig. 7.40).

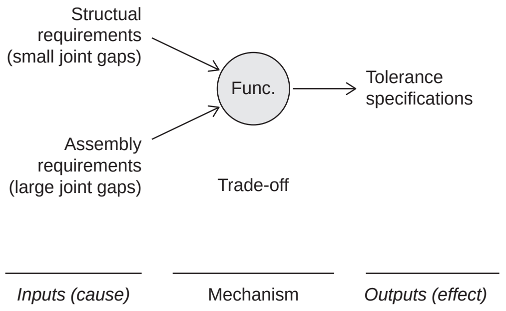


Figure 7.40: Principle of Fabrication Tolerance.

Inputs (cause) – Conflicting requirements of assembly and structure for the production of the joints, which, on the one hand, “have to allow for easy and robust assembly on site but [on the other,] are concomitantly crucial for the integrity of the entire system” [115, p. 138].

Mechanism (function) – According to the industry specification for timber boards, the tolerable deviation is ±2.0mm for dimensions up to 1m (DIN18203-3, 4.2). Due to the structural relevance of the tight-fit connection, the tolerance specification in this case has to be much more restrictive.

“The existence of gaps between panels is decisive for making the installation easier and for handling the [deviations resulting] from fabrication and shrinkage. However, it reduces bending stiffness in the joints and weakens the plate shell structure. The 1 mm gap width chosen in this project has proved to be acceptable” [115, p. 138].

Outputs (effect) – Based on the considerations regarding joint design, tolerance values, that is the deviation of the nominal values that is tolerable, have been defined – in this case 0.5mm (in order to result in a maximum gap width of 1mm) – and implemented in the fabrication model.

Significance – Tolerance specifications are defined in order to ensure the viability and quality of the structure.

B.7.3 Principle of Quality Control

The Principle of Quality Control associates the validation of the fabrication result with the given tolerance requirements via accuracy measurement using laser-tracking and statistical analysis (Fig. 7.41).

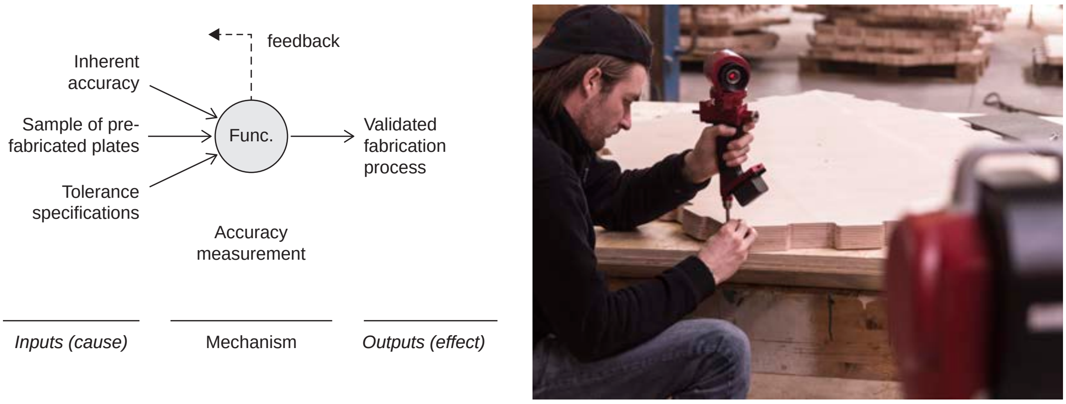


Figure 7.41: Principle of Quality Control. Right image from [103] by Schmitt (2014).

Inputs (cause) – A sample of 24 robotically produced plates; previously defined tolerance specifications that ensure the viability of the structure.

Mechanism (function) – In order to be able to make statistically relevant statements about fabrication accuracy, a sample of 24 out of the overall 243 robotically fabricated timber plates was analyzed using laser-tracking [103, p. 120]. A custom designed probe was produced by the project partner Institute of Engineering Geodesy (IIGS) for the tracking of the plate edges yielding a measurement accuracy S_M of 0.1mm. Using this probe, ten points on each finger joint element were measured. For the analysis, the CAD model of the plates was aligned with the measurements and a Best-Fit transformation performed minimizing the standard deviation S_{FM} over all measured points of one plate [169, p. 5]. By repeating measurement, alignment and minimization for all 24 plates, an average standard deviation can be calculated. The resulting fabrication accuracy S_F can be calculated as

$$S_F = \sqrt{(S_{FM}^2 - S_M^2)},$$

where S_{FM} is the combined accuracy of measurement and fabrication (see [169, p. 6]). In case tolerance specifications are not met, changes can be made to the fabrication setup and/or fabrication parameters, feeding back onto fabrication development.

Outputs (effect) – A 2-dimensional Root Mean Square (RMS) error value of 0.42mm.

Significance – The accuracy of a fabrication process must be higher (the RMS of fabrication lower) than the tolerance specifications. Quality control can then establish whether those specifications have actually been met.

B.7.4 Principle of Dimensional Changes

The Principle of Dimensional Changes associates findings on the dimensional changes of plywood plates with changing environmental conditions via laser-tracking and comparison between the measured and the CAD model (Fig. 7.42).

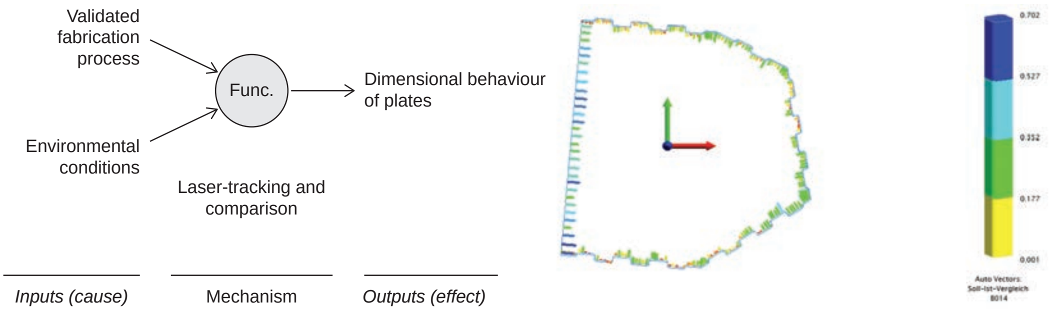


Figure 7.42: Principle of Dimensional Changes. Right image from [169] by Schmitt (2014).

Inputs (cause) – Validated fabrication process; changing environmental conditions between pre-fabrication, storage, and assembly.

Mechanism (function) – Wood is a natural material subject to dimensional changes in response to environmental conditions (most notably an increase in volume relative to increasing relative humidity). In plywood, this effect is expressed as a tendency to buckle and dish, which can have an effect on fabrication accuracy, due to the three-dimensional fabrication process [103, p. 122], but can also continue after fabrication, where it would affect the assemble-ability of the plates and finally the structural behavior. In order to understand the significance of the effect, material behavior of the plates was evaluated over time.

The accuracy measurements and analysis described above were thus repeated for four elements out of the sample of 24, right after fabrication, before transportation, approximately two weeks after fabrication, and right before assembly thereby defining three so-called measurement epochs [103, p. 121] (see also [169, p. 7]). Following the measurements, the statistical significance of the deviations between the different epochs could be tested (see [169]).

Outputs (effect) – The result of the comparison between measurements was that, even though dimensional changes could be detected in the different epochs, the variations were statistically insignificant [103, p. 122] (see also [169, p. 8]).

Significance – The evaluation of the measurement epochs showed that the chosen material under the given handling conditions did not affect the ability of the plates to be assembled nor the structural capacity of the shell.

B.8 Off-site Principles of Pre-fabrication and Pre-Assembly

B.8.1 Principle of Robotic Fabrication

The Principle of Robotic Fabrication associates pre-fabricated plates for finger-jointed plate structures with the use of a 7-axis robotic fabrication setup and valid machine code via off-site fabrication in the context of a workshop providing additional infrastructure (Fig. 7.43).

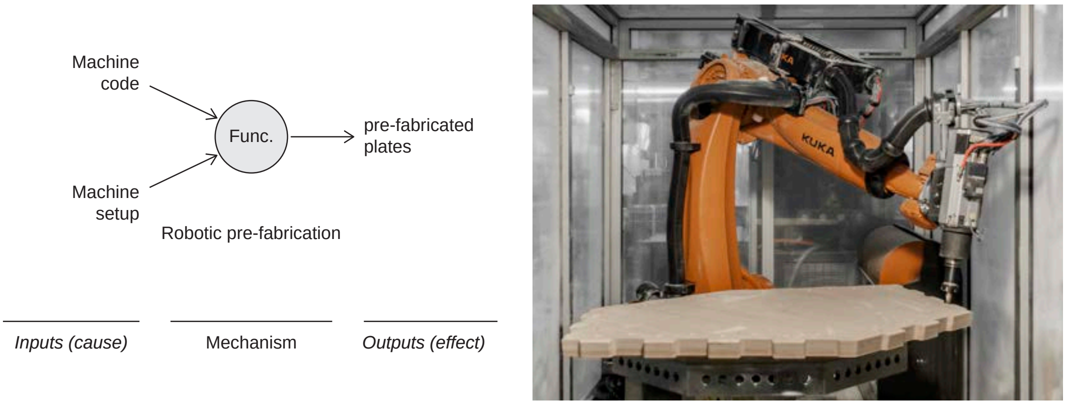


Figure 7.43: Principle of Robotic Fabrication. Right image from [103] by Krieg (2014).

Inputs (cause) – Previously generated machine code; a 7-axis robot system integrated into an enclosed cell (including operator safety) with interior dimensions of 3200 x 2100 x 2500 mm (LxWxH) consisting of: a KR120 R2700 Quantec extra HA, a 12KW milling spindle mounted on the robot flange, an additional external revolute axis used as vertical positioner (turntable), and an automatic tool changer.

Mechanism (function) – One of the main determining factors for pre-fabrication was the co-location at the timber fabricator where in addition to the structural layer, which was robotically produced, also thermal insulation and cladding layers could be fabricated (using the a Hundegger Speed-Panel machine) and pre-assembled.

The fabrication process of the Beech plywood plates consisted of two main fabrication steps: In the first step, the Hundegger Speed-Panel machine was used for pre-formatting the polygonal plates in a three-axis process, cutting them with oversize from rectangular Beech plywood stock panels of dimensions of 2550x1850x50mm. Where possible, up to two plates were nested on one stock panel in order to minimize cut-off [103, p. 120]. As the plates are all geometrically unique, it was crucially important that in this step, the plates were also automatically labelled in order to be able to identify the plates later on in the fabrication and assembly process.

In the following step, the pre-formatted plates were mounted on the turntable in the robotic milling cell where the intricate finger joints where milled down to nominal dimensions and the pockets for screws were indented into the plywood using the industrial robot-arm.

Following fabrication, the plates were palletized in the order of assembly and vacuum-packed before being shipped on-site in order to avoid any dimensional changes due to changing relative humidity.

Outputs (effect) – 243 pre-fabricated, geometrically unique Beech plywood plates, palletized in the order of assembly and ready to be shipped on-site.

Significance – The increased machinic morphospace associated with the 7-axis robotic fabrication setup provided the additional kinematic range necessary for the production of the complex joint details. In this production process of the lightweight timber plate structure, robotic fabrication represented the 'enabling' technology by solving the fabrication of the complex joints [103, p. 120].

B.8.2 Principle of Assemble-ability

The Principle of Assemble-ability associates assemble-ability of the structure with joint geometry and joint gaps via assembly sequence and edge orientation in anticlastic surface areas (Fig. 7.44).

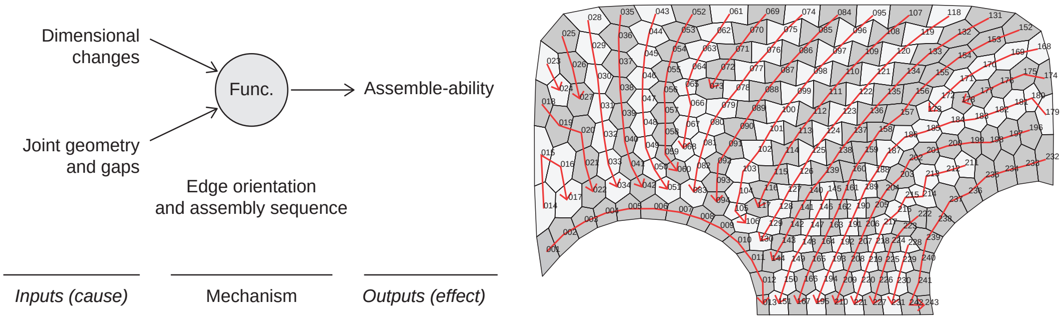


Figure 7.44: Principle of Assemble-ability. Right image by Schwinn (2014).

Inputs (cause) – Specific joint geometry and joint gaps according to the principles of joint development; dimensional changes due to changing environmental conditions between fabrication and on-site assembly.

Mechanism (function) – In order to simplify assembly, the assembly sequence was designed such that a plate would be attached to no more than two plates at a time. Starting from the corner of the building located opposite to the main entrance, the sequence was to proceed row by row, to the last plate in the south-west corner of the building (Fig. 7.44, right).

Outputs (effect) – Assemble-ability as a function of edge orientation and assembly sequence.

Significance – Assemble-ability was a driving factor for the design and development of the joint geometry and therefore a crucial aspect for the structural integrity of the entire structure [115, p. 138].

B.8.3 Principle of Pre-assembly

The Principle of Pre-assembly associates sub-assemblies ready to be shipped on site with the pre-fabricated elements that make up the shell and quality control (Fig. 7.45).

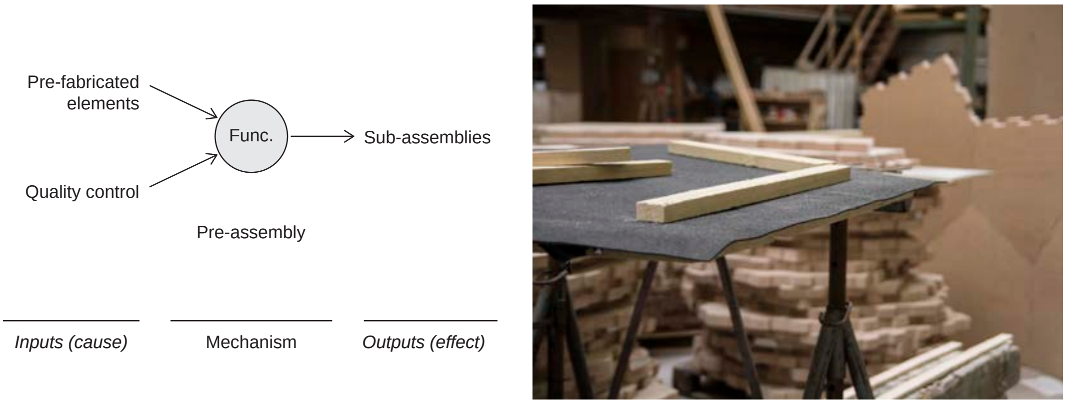


Figure 7.45: Principle of Pre-assembly. Right image by Krieg (2014).

Inputs (cause) – Pre-fabricated elements of all constructive layers of the shell; quality control.

Mechanism (function) – The insulation layer (consisting of a 35mm wood fibreboard) was produced on the previously mentioned Speed-Panel machine in a 5-axis process with mitered edges in order to ensure a tight fit between the segments. The water-proofing layer (consisting of individually tailored sheets of EPDM membrane) was produced by a separate contractor using waterjet-cutting. EPDM and wood fibre board were glued together in the workshop into an insulation panel sub-assembly. Timber slats were screwed to the panel to provide a sub-structure for the subsequent cladding layer (Fig. 7.45, right). The cladding layer (consisting of regionally sourced three-ply larch) was CNC-cut using the Speed-Panel machine in a three-axis process and prepared for on-site assembly.

The different machine codes necessary for the fabrication of all constructive layers were generated automatically from the same digital model according to the Principle of Automated CAM (B.6.1).

Outputs (effect) – Pre-assembled subassemblies ready to be shipped on site.

Significance – Pre-assembly reduced on-site labor and ensured a consistent quality of the glued bond between EPDM and wood fibre board.

B.8.4 Principle of Transport and Delivery

The Principle of Transport and Delivery associates transportability and on-site delivery of building elements with pre-fabrication and pre-assembly (Fig. 7.46).

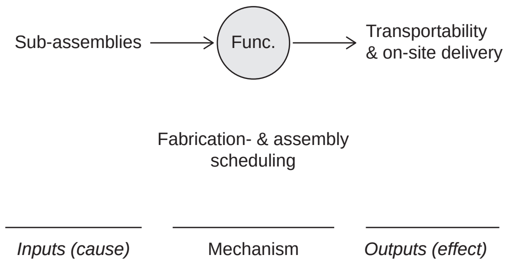


Figure 7.46: Principle of Transport and Delivery.

Inputs (cause) – Plywood plates, pre-assembled insulation subassemblies, and cladding.

Mechanism (function) – Following production all elements and pre-assembly of sub-assemblies in the workshop, they were shipped to the site in Schwäbisch Gmünd, located 65 km from the workshop in Blaustein, near Ulm, and finally being assembled on site [102, p. 138]. All elements and sub-assemblies were produced and pre-assembled in an order to be shipped on-site in the order that they were to be installed. This meant, for example, that the elements that were to be installed first, were loaded last onto the truck; and consequently that the elements which were to be loaded last, needed to be produced first in order to not block other elements by being on top of a stack or in front of a rack. Such pre-fabrication and assembly scheduling was used to eliminate the need for on-site warehousing and to minimize the amount of time that the elements were exposed to the weather.

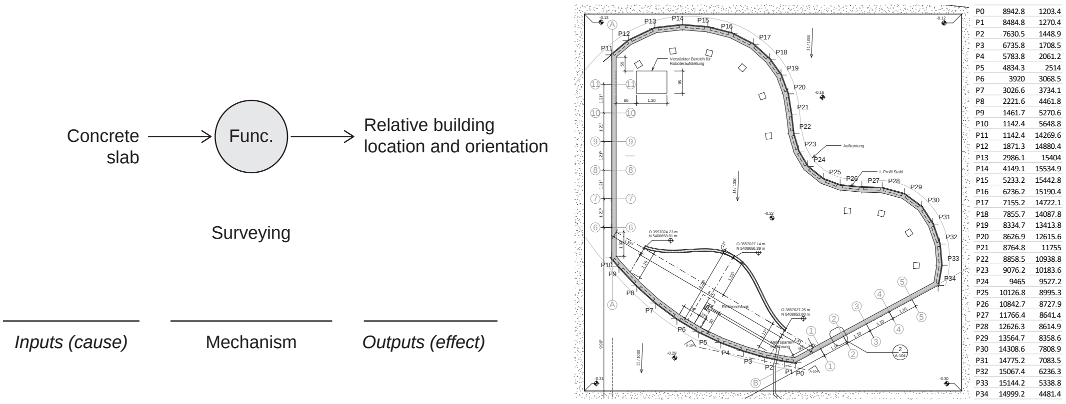
Outputs (effect) – Transportability and on-site delivery.

Significance – While just-in-time delivery was a critical aspect for the on-site assembly of the segmented shell, transportability did not add constraints to the design and development of the shell, as the maximum size (and weight) of the segments was determined by the extents of the fabrication setup.

B.9 On-site Principles of Assembly and Construction

B.9.1 Principle of Staking out

The Principle of Staking out associates building location and orientation on site relative to a local coordinate system with a square concrete slab via surveying with a total station (Fig. 7.47).



P0	8942.8	1203.4
P1	8484.8	1270.4
P2	7630.5	1448.9
P3	6735.8	1708.5
P4	5783.8	2061.2
P5	4834.3	2514
P6	3920	3068.5
P7	3026.6	3734.1
P8	2221.6	4461.8
P9	1461.7	5270.6
P10	1142.4	5648.8
P11	1142.4	14269.6
P12	1871.3	14880.4
P13	2986.1	15404
P14	4149.1	15534.9
P15	5233.2	15442.8
P16	6236.2	15190.4
P17	7155.2	14722.1
P18	7855.7	14087.8
P19	8334.7	13413.8
P20	8626.9	12615.6
P21	8764.8	11755
P22	8858.5	10938.8
P23	9076.2	10183.6
P24	9465	9527.2
P25	10126.8	8995.3
P26	10842.7	8727.9
P27	11766.4	8641.4
P28	12626.3	8614.9
P29	13564.7	8358.6
P30	14308.6	7808.9
P31	14775.2	7083.5
P32	15067.4	6236.3
P33	15144.2	5338.8
P34	14999.2	4481.4

Figure 7.47: Principle of Staking out. Right image by Schwinn (2014).

Inputs (cause) – Prepared construction site, in this case a square reinforced concrete slab with 16m edge length and a slope of 1%. The slab has been poured by a contractor at the behest of the Landesgartenschau project partner in the summer of 2013 to be used as a generic foundation for the still-to-be-designed Exhibition Hall.

Mechanism (function) – Prior to the on-site assembly of the shell, the polygonal building outline was set out by the IIGS on top of the reinforced concrete slab. Coordinates were defined relative to a local coordinate system, which was aligned with the slab edges, with origin at R 3557018.11 East and 5408652.06 North (Gauß-Krüger zone 3) (Fig. 7.47, right).

Outputs (effect) – Building location and orientation relative to an on-site reference coordinate system.

Significance – Staking out is the pre-condition for the setting up support structures, such as edge beam and false work, which allow for correct positioning of the plates relative to each other.

B.9.2 Principle of Support Structures

The Principle of Support Structures associates the permanent edge beam and the temporary false work with a set out building outline on site (Fig. 7.48).

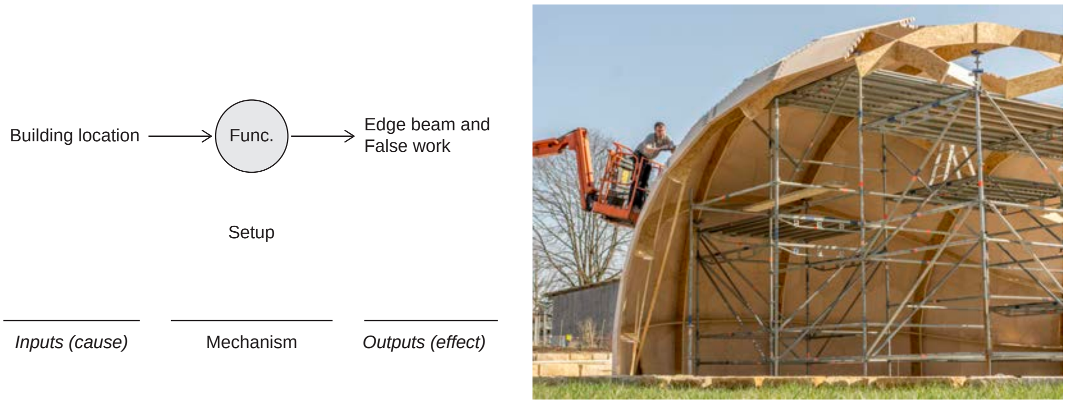


Figure 7.48: Principle of Support Structures. Right image from [101] by Krieg (2014).

Inputs (cause) – Set out building outline on site.

Mechanism (function) – On top of the concrete slab, a timber edge beam was laid out such that the outer faces of the beam's segments were aligned with the polygonal building contour and their upper faces were in the horizontal plane. The edge beam was then anchored to the concrete slab [115, p. 137].

As part of the edge beam, L-shaped steel plates were installed, to which the bottom row of plywood plates could be attached through a dedicated pocket milled into the outer surface of the bottom-row plates.

While the timber edge beam provided a permanent support for the shell, a false work was installed as a temporary support (used only during assembly and later removed). The rips of the false work (produced using CNC-machinery) retraced the inside surface of the plate shell in order to provide a reference for correctly positioning each plywood plate (see [115, p. 137]).

Outputs (effect) – Permanent edge beam providing a level interface between the shell and the concrete slab; temporary false work facilitating assembly.

Significance – The setup of support structures (permanent and temporary) is the pre-condition for precise assembly of the segmented shell.

B.9.3 Principle of On-site Assembly

The Principle of On-Site Assembly associates the assembled shell with assemble-ability, support structures, and on-site delivery of building elements (Fig. 7.49).

Figure 7.49: Principle of On-site Assembly. Right image from [103] by Krieg (2014).

Inputs (cause) – Assemble-ability, set up support structures, and on-site delivery of building segments in the order of assembly.

Mechanism (function) – The 243 Beech plywood plates were assembled over the temporary false work in the predefined assembly sequence thereby ensuring assemble-ability. While the assembly order could be treated flexibly in the synclastic areas of the shell surface, the order had to be respected carefully in the anticlastic areas in order to avoid blockage as described in the Principle of Geometric Compatibility (B.1.4).

Three workers were needed for the installation: one to operate a crane to deliver the plates, which weighted up to 60kg each, to the approximate location of assembly; a second worker standing on the platforms that also supported the false work to receive and carefully insert the plate in the exact location (Fig. 7.49, right); the third person standing on the outside on a boom lift ("cherry picker"), first temporarily fixed the plate in the correct position in relation to the previously installed plates by applying half-threaded screws in the corners before permanently attaching the plate by connecting it to the adjacent plates using full-threaded screws. The screw angle was indicated unambiguously by the screw pockets on the outside surface of each plate.

After completion of the structural layer of the shell, the vapor barrier was installed, followed by the insulation sub-assembly containing thermal insulation and waterproofing. After welding of the edges of the EPDM, the cladding layer was shipped and installed on the timber slats of the insulation panel leaving a shadow gap between the plates.

Outputs (effect) – Through the implementation of specific assembly order and insertion vectors for each plate, the construction process proved to be highly reliable and robust. The entire assembly process, including structural shell, insulation, waterproofing, and cladding layers, took four weeks [102, p. 138].

Significance – Completion of the load-bearing layer of the shell including its weather proofing is the precondition for the subsequent installation of the facade and interior fit-out and finishing.

On-site handling of building elements did not impose any constraints on the design and development of the plate structure due to the relative light weight and small sizes of the segments, which were determined by the extents of the fabrication setup.

B.9.4 Principle of Fit-out and Finishing

The Principle of Fit-out and Finishing associates the finished building with the assembled shell and installed glass façade via interior fit-out and finishing (Fig. 7.50).

Figure 7.50: Principle of Fit-out and Finishing. Right image by Krieg (2014).

Inputs (cause) – Assembled shell and glass façade providing a full enclosure.

Mechanism (function) – Completion of the main shell and installation of the glass facade including its columns, were followed by interior fit-out and finishing including electricity and lighting (Fig. 7.50, right). The columns were necessary to hold the glass façade and to limit the deflection of the free edge of the shell surface in order to avoid exceeding permissible stresses on the glass.

Additionally, an interior wall made of plaster board and supported by of aluminum profiles was installed to provide a separation between exhibition space and storage. Plumbing installations were provided for fresh water and waste water to allow for the subsequent use of the Hall as event space.

Finally, in order to maximize the utilization of the available building material, the off-cut generated during the pre-formatting of the beech plywood plates was reused in the hardwood flooring as parquet lamellas [172, p. 187].

Fit-out and finishing took four weeks to complete such that the Exhibition Hall and the surrounding landscaping were finished just in time for the opening of the landscaping show at the end of April 2014.

Outputs (effect) – The result is a fully enclosed and insulated building providing floor area for exhibitions and educational activities including amenities.

Significance – Meeting architectural requirements that go beyond the demonstration of building technology, such as the provision of a full enclosure and the interior fit-out and finishing, distinguishes the Exhibition Hall from other research into segmented shells.

7.4 Second-level Principle Chains of Case B

On the second-level of abstraction, individual low-level principles are connected into domain-specific principle chains, where one principle's output becomes another principle's input. In this way, a causal dependency between input factors and results of a development domain can be illustrated. The notation of principle chains thereby follows a similar approach as in Case Study 1 (see 6.4).

7.4.1 Joint Development

Joint development associates an adaptive joint model for edge-wise connecting timber plates with geometrical (finger joint geometry, surface curvature), fabricational (subtractive fabrication), and constructive (metal fasteners, assembly sequence) inputs through the specific sequence of low-level functional principles (Fig. 7.51):

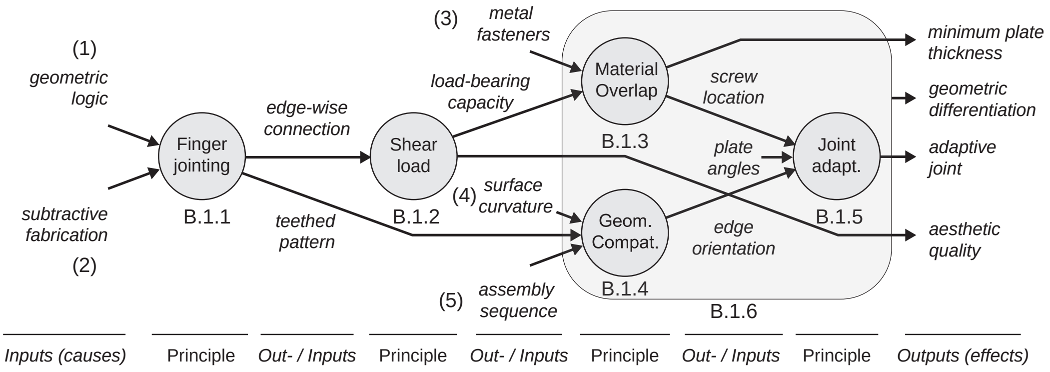


Figure 7.51: Chain of Joint Development principles.

7.4.1.1 Inputs

As stated above, an important driver in the plate system's development was the objective [i] to further develop integrated timber joints in order to meet the requirements of the building industry in terms of building code and regulations, efficient pre-fabrication, and on-site assembly, including the efficient and economic use of material and financial resources. The objectives for joint development thus related to the structural performance of the joint, to building code requirements for edge-wise connecting plywood plates under different load conditions, to keeping production times short, and to facilitating straightforward on-site assembly. A special concern were the requirements for assembly of segments in the areas of negative Gaussian curvature.

7.4.1.2 Methods

Structural objectives were addressed by articulating the plate edges with the teethed pattern of finger-joints in order to interlock timber plates along their edges in the in-plane shear direction. Additionally, crossing screws were inserted in the joint to accommodate axial forces and out-of-plane shear forces and to a lesser degree bending moments between the plates. Given the use of screws in edge-wise connected plates, building code requires a minimum material overlap as a multiple of the screw diameter, which resulted in a plate thickness of 50 mm.

Assembly-related objectives were implemented as two features of the plate edge: (1) screw pockets indicate the location and direction of the crossing-screws during assembly; and (2) local curvature directions define the orientation of the edge faces. This is to ensure geometric compatibility and to avoid blockage by other finger joints along the plate insertion vector. A special assembly sequence was therefore defined in order to ensure assemble-ability in anticlastic surface areas.

Finally, a generative, parametric modeling approach allowed implementing these structure- and assembly-related functionalities in Rhino and Grasshopper, thereby making every detail adaptable to locally varying requirements and to global parameters. In terms of joint development, this led to an adaptive joint model that allows for the integration of multiple performance criteria into the plate connection. Functional integration, as seen in biological structures, thus led to geometric differentiation on the level of the plate connection.

7.4.1.3 Principle chain

Discrete development steps, which have been identified and defined as low-level functional principles in B.1, can thus be connected to form a principle chain that relates the results of joint development with the inputs mentioned above via the following sequence of functional principles:

- (B.1.1) the Principle of Finger Jointing associates the teethed pattern of finger joints with their geometric logic and the fabrication logic of subtractive fabrication;
- (B.1.2) the Principle of Shear-load transfer associates the shear-load bearing capacity of the joint with a teethed pattern in an edge-wise connection via interlocking between the plates;
- (B.1.3) the Principle of Material Overlap associates a minimum shell thickness with the use of metal fasteners, such as screws, in edge-wise connected timber plates via specific code requirements;
- (B.1.4) the Principle of Geometric Compatibility associates the orientation of edge-faces with the surface curvature at the location of the associated segments and a given assembly sequence via aligning edges faces with the subsequently installed plate;
- (B.1.5) the Principle of Joint Adaptation associates an adaptive joint model with varying plate angles, principle curvature directions, and varying structural

requirements via an integrative parametric model of the joint;

- (B.1.6) the general, high-level Principle of Geometric Differentiation associates a geometrically differentiated outcome with a diverse set of input factors via the integration of multiple performance criteria into the plate connections.

7.4.1.4 Result

The result of this principle chain is a geometrically adaptive, locally differentiated large-scale finger joint model for timber plates with embedded functionalities based on principles of assembly, building code and structure (Fig. 7.52).

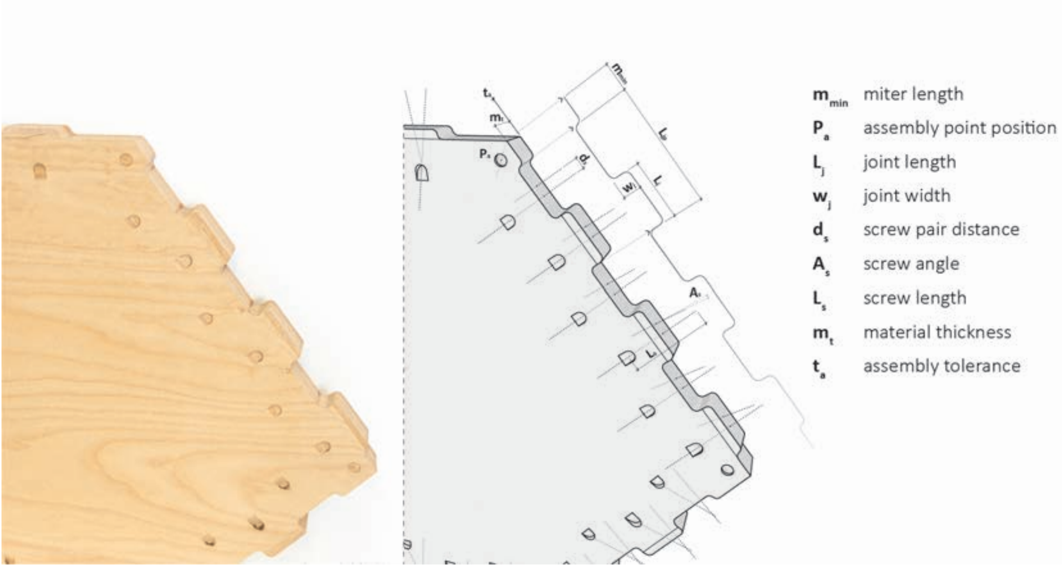


Figure 7.52: Parameters of the geometrically adaptive finger joint model. From [172] by Krieg (2014).

7.4.1.5 Significance

Joint development met the objective related to structural performance and to applicability in building practice in that it kept production times short, allowed straightforward assembly on site, and obtained technical approval. It also provided the input for the further development of the segmented shell structure.

7.4.2 Structural Joint Development

Structural joint development relates increased in-plane stiffness in a plate shell, compared to shells without finger-joints, with locally bending-weak joints between edge-wise connected plates and the need for gaps between the joints through the specific sequence of low-level functional principles (Fig. 7.53):

7.4.2.1 Inputs

The structural performance of a plate shell is to a large extent influenced by the properties of the connections between the plates [115, p. 132]—in this case the

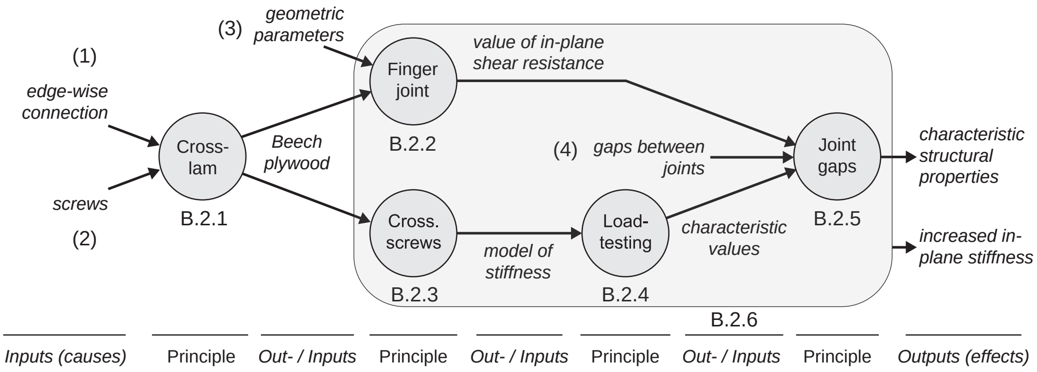


Figure 7.53: Chain of Structural Joint principles.

hinged connections that provide shear load-bearing capacity. On the other hand, joint forces and segment layout are highly correlated, as demonstrated by the structural segmentation principles (see 7.4.4): the segmentation interrupts the stiffness continuity of the shell [114, p. 123] and the joints need to translate the forces between the segments. Due to the trivalent plate layout, predominantly shear forces occur along the edges, but also axial forces and, to a lesser degree, bending moments.

The objective in the structural design of the joints that [iii] the connection should ultimately be applicable to the building industry meant for the development process to enable the required load transfer while also following building code [103, p. 116]. Input factors for structural joint principles predominantly relate to the properties of the material, the parameters of the edge-wise connection, and the gaps between the joints stemming from fabricational and assembly requirements.

7.4.2.2 Methods

Regarding the material aspect, the choice was limited to a material that has similar structural properties in all directions, such as plywood and cross-laminated timber, as connections between plates occur at all edges of a plate in a timber plate structure and membrane forces might come from various directions. Given the use of screws, grain directions parallel to the screws generally have to be avoided, therefore plywood/LVL was chosen, where the thickness of the individual layers of veneer is much lower than that in cross-laminated timber, which ensures that the screws penetrate multiple layers of different grain directions. Finally, Beech plywood was chosen as its stiffness and strength are about twice as high as plywood made from softwood, and therefore less material and fewer screws are needed [115, p. 125].

The novel connection for plate structures made from Beech plywood, which was developed as part of this research, consists of alternating finger joints and crossing screws. According to Li and Knippers [115, p. 117], finger joints function similarly to step joints in a traditional truss: the thrust force is taken by the contact surface and transferred to the shear plane. Using an analytical approach, the in-plane shear stiffness and capacity of the joint were calculated, which were later used to determine spring coefficients in the FE-analysis and to check whether any joint exceeds its design capacity.

The axial forces and the out-of-plane shear forces are taken by crossing screws [115, p. 126]. Using a similar analytical approach, Li and Knippers first developed a mathematical model of the stiffness and capacity of the crossing screw connection as a function of the withdrawing capacity of the screw from the plywood. In a second step, the withdrawing stiffness and capacity were determined through a series of physical load tests and the characteristic values of the connection were calculated using the developed mathematical model. Similar to the step joint model, these values were then used in the subsequent FE-analysis to determine spring coefficients and to check whether any screw connection exceeds its design capacity.

In a detailed analysis, Li and Knippers [115] examined the structural mechanism, the effect of finger joints on the transfer of axial and shear forces between the plates, and the role that the gaps play in the overall load-bearing capacity. In the FE-model, the stiffness of the connections is simulated by spring elements. Simulation of the gap behaviors showed that “the in-plane shear in a connection will be completely taken up by the crossing screw joint if the in-plane displacement of the connection is smaller than the gap width, but once the displacement reaches the gap width, contact occurs, and the finger joint starts to take the force” [115, p. 133]. After in-plane contact in the shear direction, the high in-plane resistance of finger joints changes the force transfer in the structure so that more forces are transferred between the plates as in-plane shear forces, instead of axial forces; as a result, the axial forces, which are completely taken up by screw joints, are reduced. With regards to the relationship between finger joints and shell stiffness, the analysis showed that “the application of finger joints effectively increases the in-plane stiffness and the strength of the connections and thus makes the plate structure stiffer and stronger” [115, p. 138].

7.4.2.3 Principle chain

Six discrete development steps have been identified in Structural Joint Development and defined as low-level functional principles in B.2. Their inputs and outputs can now be connected to form a principle chain that relate domain-specific development results to the inputs mentioned above via the following sequence of functional principles:

- (B.2.1) the Principle of Cross-laminated Veneer associates the use of multi-layer Beech plywood (multiplex) with the use of screws in timber plates that are edge-wise connected at the perimeter;
- (B.2.2) the Principle of Finger Joints associates characteristic values of a finger

joint with geometric and material parameters via an analytical model;

- (B.2.3) the Principle of Crossing Screws associates an analytical model of stiffness and capacity of a crossing screw connection with the application of crossing screws in edge-wise connected timber plates;
- (B.2.4) the Principle of Physical Load Testing associates characteristic values of the crossing screw connection with an analytical model of the connection;
- (B.2.5) the Principle of Joint Gaps associates characteristic structural properties of the connection in both in-plane shear and axial load directions with gaps between the fingers and spring coefficients;
- (B.2.6) the Principle of Higher Stiffness associates increased in-plane stiffness with the high stiffness of finger joints once the segmented shell is fully loaded.

7.4.2.4 Result

The result of structural joint development is an increased in-plane stiffness in the plate shell, compared to shells without finger-joints, after in-plane contact of the finger joints in shear direction (Fig. 7.54).

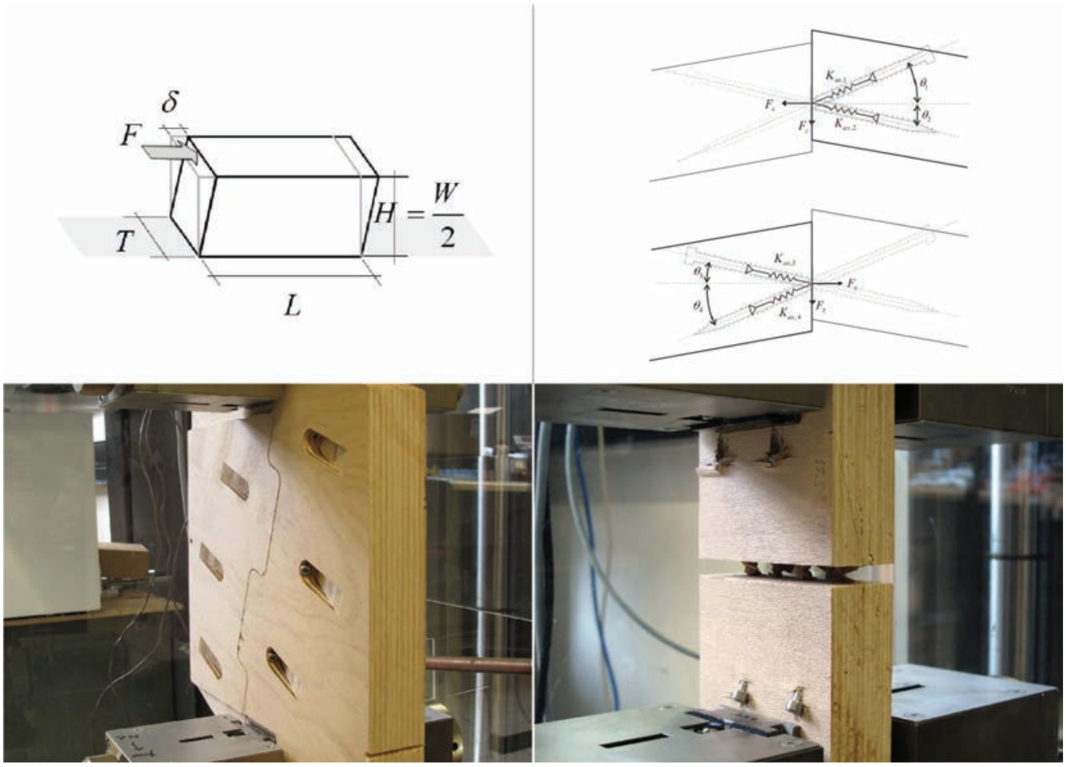


Figure 7.54: Components of finger joint model and validation. Top row: models of step joint (left) and crossing screw connection (right). From [115] by Li (2015). Bottom row: Physical load test of joint samples. Bottom left: Shear load test. From [115] by Li (2014). Bottom right: Withdrawal test (Li, 2014).

7.4.2.5 Significance

The physical and computational experiments that were conducted in structural joint development showed that the structural behavior of the plate connection is an influential factor in the global structural behavior of a segmented plate shell. Together with the geometric pattern of the segmentation, the connection determines "where and how the internal forces are transferred in segmental plate shells" [114, p. 7].

7.4.3 Segmentation Development

Segmentation development relates planar timber plates in a trivalent configuration with a double curved surface and input points on the surface through a specific sequence of low-level functional principles (Fig. 7.55).

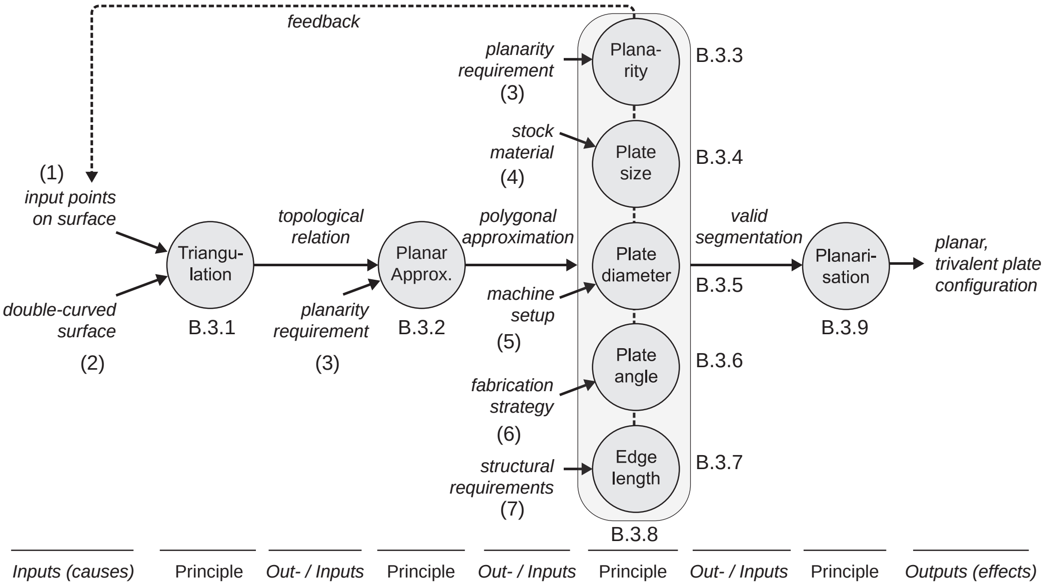


Figure 7.55: Chain of Segmentation Development principles.

7.4.3.1 Inputs

Similar to Case A, taking the planarity of the chosen building material as a given, in this case Beech plywood, required the ability to cover double-curved freeform surfaces with planar polygonal segments. In view of objective [i] to meet the requirements of the building industry not only on the level of the joint, but also on the level of the segmentation, designing a viable plate system necessitated a reduction of the number of segments per unit area relative to the ICD/ITKE Research Pavilion 2011 investigated in Case Study I (see 6). Furthermore, the segmentation needed to consider structural requirements (see 7.4.4), material parameters and specific fabrication requirements (7.4.6) involving workspace and boundary conditions, such as stock material, and building part handling and assembly.

7.4.3.2 Methods

The objective to reduce the number of building elements was addressed by increasing the average plate size through a single-layer approach where one plate corresponds to one segment instead of a polyhedral approach where many plates constitute one segment as in Case A. The planarity constraint and the objective to reduce the number of building elements led to the choice of TPI as a segmentation method for approximating double-curved surfaces with planar polygonal faces as introduced above.

Starting from a series of input points on a double-curved surface, a topological map was generated representing the neighborhood of each input point through edges in a triangulated mesh. For each vertex in the mesh a closed planar polyline could be generated using the TPI method, which led to a "watertight", trivalent configuration of polylines: convex in the case of positive Gaussian curvature ($K > 0$), concave in the case of negative Gaussian curvature ($K < 0$). A special case, however, were parabolic areas (where Gaussian curvature $K = 0$), which occur, for example, in the transition zone between synclastic and anticlastic surface areas. In these areas, customized TPI resulted in temporarily non-planar polylines.

Following the preliminary segmentation, a series of validity checks were conducted with the aim to meet different performance criteria related to geometry (planarity), material (available material stock sizes), fabrication (jointing strategy and working envelope), and structure (minimum edge lengths to allow for the width of a finger joint and its crossing screws). In a feedback loop, specific rules were then used to update the positions of the input points in order to meet the performance criteria, using a prioritized sequence and weighting of rules.

As the system of rules was not guaranteed to converge to a stable solution where all constraints were met (especially planarity of plates in parabolic areas), a post-process was implemented that minimized the remaining non-planarity of the polylines, in a trade-off, by relaxing the tangency constraint stemming from TPI. Planar polylines that have passed the validity check could then represent the outlines of planar plates in the segmentation of double-curved surfaces.

7.4.3.3 Principle chain

Nine discrete development steps have been identified in segmentation development and defined as low-level functional principles in B.3. Their inputs and outputs can thus be connected to form a principle chain that relates planar timber plates in a trivalent configuration with the inputs mentioned above through the following sequence of low-level functional principles:

- (B.3.1) the Principle of Triangulation associates the generation of a topological neighborhood with a series of input points on a free-form surface via a triangulation mechanism;

- (B.3.2) the Principle of Planar Approximation associates a mostly planar trivalent configuration of segments with a topological network of input points and the planarity requirement via a customization of TPI;
- (B.3.3) the Principle of Planarity associates increased planarity of non-planar plates with a given segmentation and the planarity requirement.
- (B.3.4) the Principle of Valid Plate Size associates a valid plate size with a given segmentation and material constraints;
- (B.3.5) the Principle of Valid Plate Diameter associates a valid plate diameter with a given segmentation and machine constraints;
- (B.3.6) the Principle of Valid Plate Angle associates valid plate connection angles with a given segmentation and a fabrication strategy;
- (B.3.7) the Principle of Valid Edge Length associates valid plate edge lengths with a given segmentation and structural requirements;
- (B.3.8) the high-level Principle of Segmentation Validity associates a valid planar approximation with a multi-criteria-informed segmentation;
- (B.3.9) the Principle of Planarization associates a planar segmentation with a validated planar approximation of a double-curved surface.

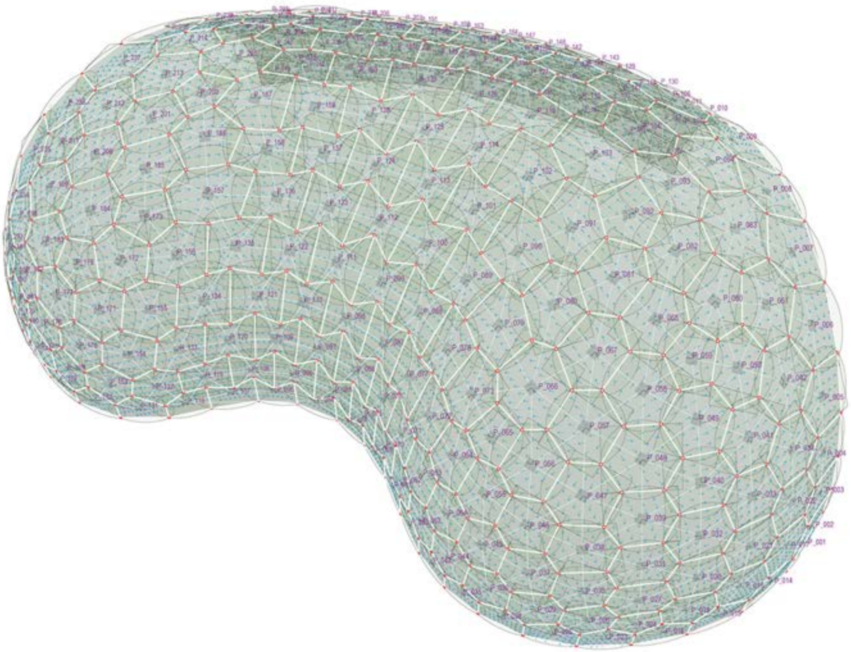


Figure 7.56: Validated segmentation model. From [175] by Schwinn (2015).

7.4.3.4 Result

The result is a fully planar segmentation even in areas where the TPI method typically fails (Fig. 7.56).

7.4.3.5 Significance

The combination of the principles of Triangulation, Planar Approximation, Segmentation Validity, and Planarization provides an approach for the interactive manipulation of the segmentation pattern by adjusting the positions of the input points on the double-curved surface based on multiple performance criteria.

This ability raises the question of an adequate method for automatically updating the positions of input points in order to meet the various performance criteria mentioned above. In this case, ABMS has been chosen.

7.4.4 Structural Segmentation Development

Structural segmentation development relates a bending bearing, force-aligned segmentation that meets the requirements for permanent buildings with joint characteristics, segmentation pattern, global geometry, and building code through a specific sequence of low-level functional principles (Fig. 7.57).

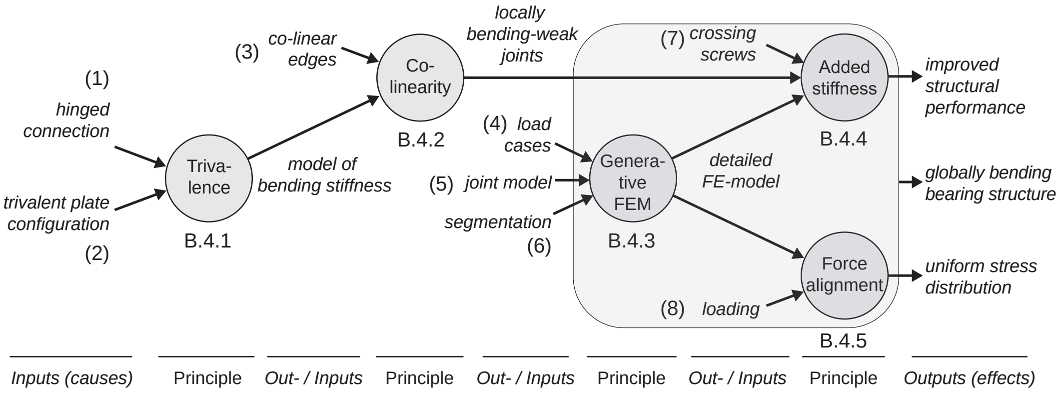


Figure 7.57: Chain of Structural Segmentation principles.

7.4.4.1 Inputs

The structural performance of segmented shells is determined on multiple levels of scale: on the level of the joint, the joint stiffness affects the force path, “because when the joints are stiffer, larger forces will be attracted to flow through” [114, p. 7] as demonstrated above (see 7.4.2); on the level of the segmentation, segment pattern and topology also play a crucial role as the segmentation interrupts the stiffness continuity of the shell and the pattern of segmental plates affects the force transfer; on the level of the global geometry, every area in the surface needs to be double-curved to generate shell action. For a given shell shape, “geometric pattern and joint stiffness thus determine where and how the internal forces are transferred in segmental plate shells" [114, p. 7].

In addition to connection properties, segmentation pattern, and global geometry, input factors for structural design and analysis were building code, such as Eurocode 5, as well as the client's brief for the Exhibition Hall to stand on site for at least 5 years, which determined wind and snow load requirements for the shell applicable to permanent buildings.

7.4.4.2 Methods

As stated before, the importance of a trivalent segmentation pattern where "three plates meet at one point and are hinged at the three intersection lines" is that "each plate is constrained by the other two" [114, p. 3], which is the precondition for plate action and kinematic stability of the overall structure (see 4.4.2).

In an analytical approach, Li and Knippers [114] analyzed the relationship between segment layout and kinematic stability using a simplified mathematical model: they showed on the one hand, that the trivalent segment pattern does provide sufficient overall bending stiffness in areas of positive and negative Gaussian curvature and that the bending resistance of the pattern is independent of type and direction of the pattern. On the other hand, they showed that if angles between consecutive plate edges become close to 180° (co-linear), such as in the transition zone between areas of positive and negative Gaussian curvature, the bending stiffness generated by the trivalent geometry disappears. In the case of the Exhibition Hall, the local rotation between plates in the transition zone would thus become too large to be acceptable [114, p. 11].

In order to be able to conduct multiple structural studies and to link structural analysis to the design process of the shell, most importantly to the development of the segmentation, a custom digital workflow was implemented that automatically generated structural models from trivalent, polygonal B-rep surface models [103, p. 117]. The workflow included the discretization of the plate model using specific meshing algorithms, the generation of shell elements and springs connecting opposing edges between the plate segments, the definition of support conditions and load assumptions: with timber being a relatively light-weight material and given the requirements for permanent buildings, the critical load cases thus were combinations of unsymmetrical snow loads and the wind loads [115, p. 125]. In this way, structural analysis could be directly linked to the design process and the configuration of connections could easily be varied for finding a locally optimal solution [103, p. 117].

In a comparative FE-analysis with and without added bending stiffness, for example provided by crossing screws, Li and Knippers [114, p. 11] showed that by adding a comparatively small amount of bending stiffness (in comparison with the bending stiffness provided by trivalent geometry) the rotations between the plates become acceptable.

Furthermore, four FE-simulations were conducted, one for each combination of the two principal pattern types and directions, in order to investigate the re-direction of the forces as they pass through the connections, and the effect of the segment pattern on the force transfer. The authors showed that even though the bending stiffness of the four patterns is the same, the stresses and reaction forces at the supports are more uniform if the pattern is aligned with the principal load direction [114, pp. 8, 12].

7.4.4.3 Principle chain

The structural experiments have led to five findings, which have been identified as systematic or functional principles in B.4. They form a second-level principle that relate domain outputs to the inputs mentioned above via the following sequence of low-level functional principles:

- (B.4.1) the Principle of Trivalence associates the kinematic stability of a segmental plate shell with hinged connections and a trivalent segment layout via geometric locking;
- (B.4.2) the Principle of Co-linearity associates zero local bending stiffness of hinged joints in a trivalent plate layout with co-linear consecutive edges and bending in the structure via the hinge mechanism;
- (B.4.3) the Principle of Generative FE-Modeling associates a detailed FE-model for structural analysis with specific load cases, a validated joint model, and a B-rep model of a segmented shell via a generative structural modeling approach;
- (B.4.4) the Principle of Added Bending Stiffness associates increased bending resistance of a structure with locally bending-weak joints and crossing screws via increased bending stiffness in the joints;
- (B.4.5) the Principle of Force Alignment associates a more uniform stress distribution in a trivalent pattern with a given loading via alignment of the pattern to the principal load direction.

7.4.4.4 Result

The result of structural segmentation development is a bending bearing, force-aligned segmentation that meets the requirements for permanent buildings (Fig. 7.58).

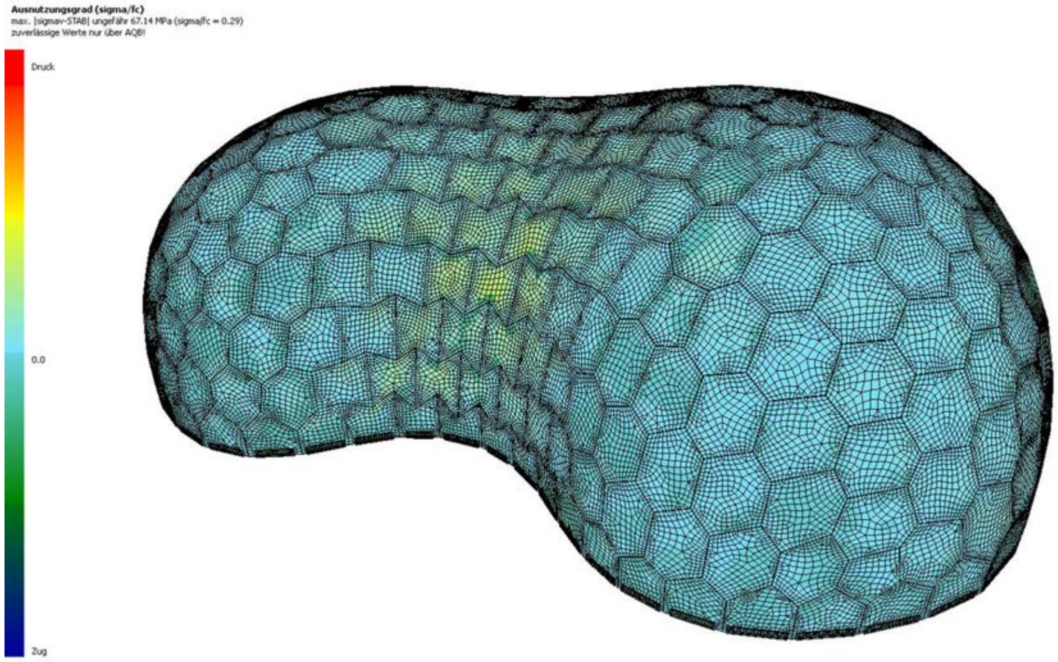


Figure 7.58: Global FE model showing utilization of Beech plywood plates (Li, 2014).

7.4.4.5 Significance

The chain of principles illustrates how joint design, the plate layout, and global geometry each contribute to the structural performance of a segmented shell: while the global stability is generated by its double-curved shape and the binding strength in the connections between the segments [103, p. 115], the force flow is highly dependent on joint stiffness and segmentation pattern. Similarly, the segmentation pattern is highly dependent on global geometry as demonstrated by segmentation principles (see 7.4.3). Addressing these three aspects is the pre-condition for the development of a lightweight segmented shell that meets the requirements of permanent buildings.

7.4.5 Shell Form-finding

The development area of shell form-finding relates a double-curved NURBS-based surface used as a reference for segmentation with architectural and structural design criteria, fabrication, and programmatic requirements via a specific sequence of low-level functional principles (Fig. 7.59).

7.4.5.1 Inputs

Closely related to structural simulation and analysis of the segmentation is form-finding of the global shell shape. Next to the design brief defining (1) global requirements for the shell, for example to cover the desired usable floor area, input factors for shell development were (2) boundary conditions, such as the maximum plate diameter based on the given fabrication setup and transportability resulting in the number of required plates to cover the shell surface; and (3) the structural

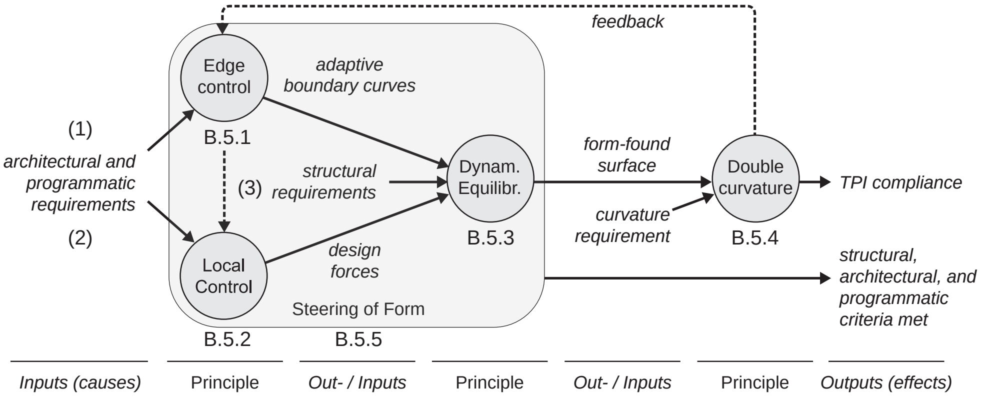


Figure 7.59: Chain of Form-finding principles.

requirement for double-curvature in order to enable shell action and to facilitate the use of the TPI method.

7.4.5.2 Methods

With timber being a relatively light-weight building material, self-weight was not the decisive load case, therefore in this case a form-finding approach that yields funicular shapes based purely on self-weight (like the hanging chain) was not required [115, p. 125].

Surveying the existing literature on the Landesgartenschau Hall turned out that the computational form-finding approach pursued in this project is only marginally described and, apart from the statement above, only in publication [172]. The analysis is therefore largely based on the latter publication.

According to Schwinn, Krieg, and Menges [172, p. 185], the computational design approach aimed for a force equilibrium between top-down design inputs and catenary form by employing additional local controllers that acted as design forces in the realm of physical simulation, similar to Case A. The approach was based on the physics-based simulation of forces, in this case using the PS-method, as well as on the definition and manipulation of additional forces representing design inputs such as site context and program through controllers.

For example, by interactively adjusting the control points of a boundary curve, which acts as a guide for the boundary of a reference mesh during form-finding, the designer could control the spatial outcome of the form-finding process in order to meet architectural and programmatic requirements, and of plate generation: “the custom digital design approach used an iterative process for the adaptation of the global geometry to local plate parameters and vice versa” [172, p. 185].

The design approach thus enabled the introduction of top-down design inputs in order to address the input factors mentioned above, while computational form finding sought for a structurally informed, double-curved global geometry, allowing the designer to steer the global form of the shell (see also [89]). The result is a shell shape which, while being funicular with respect to the chosen design forces (see [210, p. 26]), effectively requires bending stiffness in the shell surface for real world load cases.

7.4.5.3 Principle chain

Five main development steps have been identified as part of the computational design approach for the segmented timber shell and defined as low-level functional principles in B.5. Together, they form a principle chain that relates a double-curved NURBS-based surface used as a reference for segmentation with the inputs mentioned above via the following sequence of principles:

- (B.5.1) the Principle of Edge Control associates adaptive boundary curves with architectural design inputs and programmatic requirements via a 3D reference curve that can be interactively manipulated by the designer;
- (B.5.2) the Principle of Local Controllers associates force vectors representing attraction or repulsion forces with architectural and programmatic inputs via a physics-based conceptualization of design inputs;
- (B.5.3) the Principle of Dynamic Equilibrium associates a form-found, double-curved shell shape in dynamic equilibrium with an adaptive user-defined boundary curve, attraction and repulsion forces that act on the particles during the form-finding process, and structural requirements via a dynamic equilibrium approach.
- (B.5.4) the Principle of Double Curvature associates structural stability of a thin shell surface and TPI-conformity with a double–curved surface in dynamic equilibrium via the predominance of membrane forces in the shell surface and bending stiffness, which avoids buckling in the case of compressive stresses (see [210, p. 26]);
- (B.5.5) the Principle of Steering Form is a high-level principle that associates a form-found double-curved surface with multiple performance criteria via a feedback process.

7.4.5.4 Result

The resultant double-curved, free-form design surface integrates multiple criteria ranging from structural design and fabrication, to program and architectural design (Fig. 7.60). It is also highly correlated with the plate system itself, where the individual plate's geometry directly depends on the local curvature.

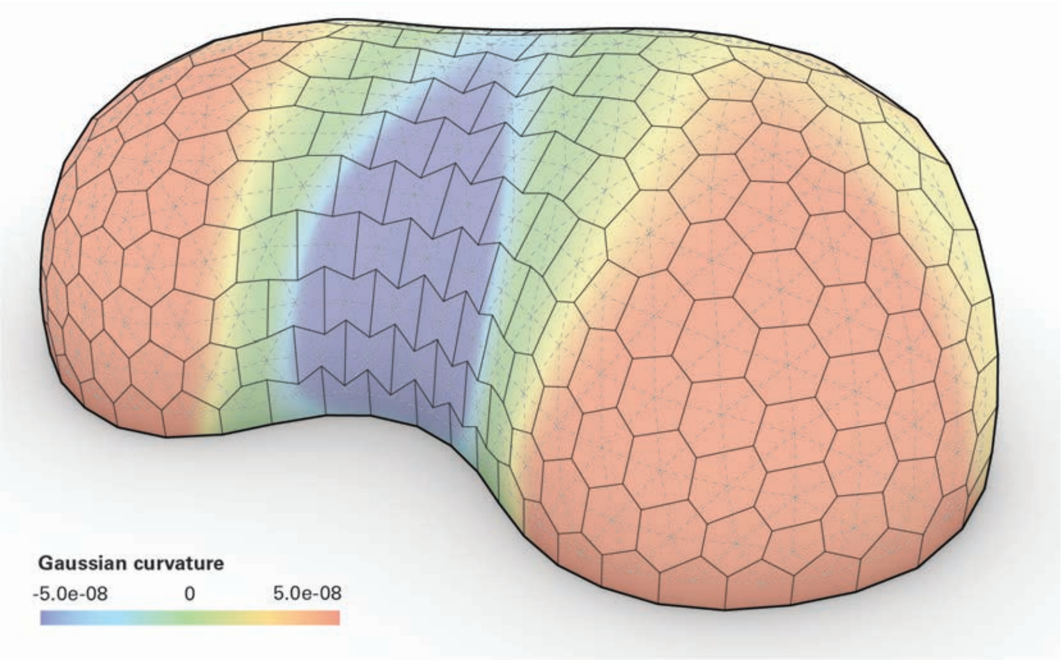


Figure 7.60: The double-curved, free-form design surface integrates multiple criteria and is correlated with the plate system.

7.4.5.5 Significance

This approach shows particular advantages for light-weight construction systems that do not depend on compression-only geometries. In combination with a trivalent plate layout, the approach thus widely extends the design space while still pursuing a structurally informed solution during form-finding [172, p. 185].

7.4.6 Fabrication Development

Fabrication development associates a validated fabrication process, the space of producible form (Machinic Morphospace), and KUKA.CNC-specific NC-code with plate geometry, fabrication and material parameters, and machine setup via a specific sequence of low-level functional principles (7.61).

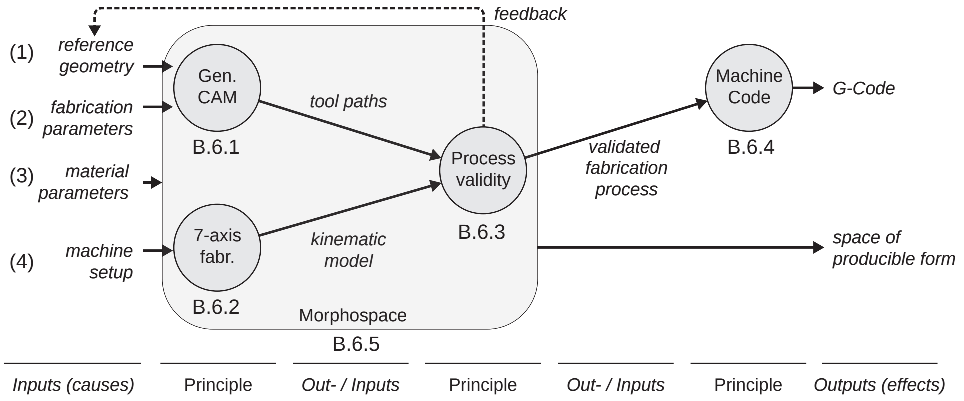


Figure 7.61: Chain of Fabrication Development principles.

7.4.6.1 Inputs

The design of a segmented timber shell is to a large extent influenced by the structural and geometry-related considerations outlined above. Equally important aspects, however, are related to fabrication, such as the equipment used, the dimensions of the desired material stock, and considerations regarding fabrication cycles and one-off production (batch size one).

In contrast to serial production, where the same machine code can be executed multiple times with the aim of producing identical elements, the challenge of one-off production is that each element requires its own unique set of fabrication instructions. The large number of geometrically unique building elements, from which segmented shells are composed, together with the relative novelty of edge-wise connected wood plates, for which no standardized fabrication cycles existed, meant that, similar to Case A, a generative programming approach towards CAM was required for the fabrication modeling of the thousands of plate joints.

In practice, this modeling step, where tool paths are generated based on reference geometry and job-specific parameters (such as tool diameter, tool path offset for roughing and finishing, tolerance values, tool approach and retraction strategies) is typically conducted by a contractor by parametrically configuring pre-defined fabrication cycles. In this case, however, such an approach was not applicable due to the lack of viable fabrication cycles. Therefore, development of custom fabrication cycles and the integration of this modeling step in the design environment were part of the research objectives.

In addition to this context, specific input factors were (1) the structurally determined gap width between panels of 1 mm (both in local x and y directions); (2) tolerance specifications for panel fabrication and tool path generation. Given a specific fabrication setup, tolerance specifications are necessary for the tool path generation in order to meet structural and assembly requirements; and (3) the fact that pre-fabrication of the segments was to be performed in cooperation with and on location at Müllerblaustein GmbH, where existing infrastructure could be used in addition to the robotic milling cell, which was provided by KUKA Roboter GmbH for the duration of the project.

7.4.6.2 Methods

A computationally lightweight B-rep model was used as the geometric representation of the plate configuration, which maintained the same topological relation between plates and edges as the triangular dual between vertices and edges. In this way, the adjacency information of edges and plates and of plates and neighboring plates was contained in the model. Based on the topological and geometric representation of the plate arrangement, tool-paths could be generated through trigonometric operations based on geometric attributes and based on user parameters [172, p. 183-4].

Crossing screw angles, the number of finger joints (as a function of the joint width) and edge face directions were a function of local, edge-specific geometric attributes, such as the connection angle between two adjacent plates and the length of their shared edge. User parameters included material thickness, joint width and depth, and tolerances that were defined globally. Specifically, the joint width was set to allow for the space for the crossing screws and thus to meet requirements of the building code with respect to material overlap. Based on these parameters, several custom CAM strategies were developed for different purposes:

- contour cutting of the 3-dimensional finger joints using robotic milling through consecutive roughing steps and one final finishing step with the shaft of a 20mm diameter milling tool;
- auxiliary features such as milling pockets for the localization of crossing screws;
- drilling of pilot holes for the crossing screws; and, finally,
- spot facing for planing the teeth of the finger joints in synclastic surface areas where fingers would protrude from the surface of the shell and would thus interfere with the thermal insulation layer.

The fabrication model with its various tool paths for finger joint fabrication could subsequently be generated automatically for each individual plate edge in the plate structure without the need for any further geometric information. The tool paths formed the basis for the simulation of the robot kinematics within the CAD-environment in order to visualize the movement of the 7-axis robot system including turn table and to allow for adjustments of the robot pose if necessary.

While the fabrication setup consisted of a numerically controlled 5-axis Hundegger Speed-Panel machine and a robotic milling cell, in the subsequent step, only the kinematics of the robotic cell were simulated in the design environment. On the Hundegger, the plywood plates were to be ‘pre-formatted’, that is cut to shape with oversize. In this case, the work space of the machine was larger than the maximum available Beech plywood panel used and therefore did not impose any constraints relating to plate size onto shell design. Simulation of its kinematics was performed by a dedicated operator using proprietary software and an Extensible Markup Language (XML)-based exchange format generated from the design model.

After pre-formatting, the robotic milling cell was to be used for the milling of the complex and intricate joint geometry. The cell was equipped with a KUKA industrial robot arm of the Quantec series with a reach of 2500 mm and 120 kg payload. A 12 kW milling spindle was mounted on the robot flange as an end-effector and an additional external revolute axis was used as vertical positioner (turntable), on which the work pieces were affixed. The setup also included an automatic tool changer, which allowed changing pre-configured milling chucks during a milling cycle without interrupting the process. The 7-axis setup was integrated into a robot system (including operator safety) as an enclosed cell with interior dimensions of 3200 x 2100 x 2500 mm (LxWxH). Considering the space taken by the robot and the tool changer, the resultant available workspace was significantly smaller than the maximum available beech plywood panel and thus imposed a limit on the producible plate size.

Based on the successful simulation, ISO-conforming machine code was exported from the fabrication model by translating tool path vertices, tool orientation vectors and additional machine information as structured, machine-readable text according to specific formatting conventions. These conventions were based on the technology package KUKA.CNC, which offered milling-specific functionality that is not available when programming KUKA robots in KRL. In contrast to Case A, the instructions could thus be read directly by the robot controller (KR C4) and executed without the need for further post-processing and file format translations.

The robot setup defined many of the fabrication parameters such as size of the workspace, degrees of freedom of machine kinematics, dimensions and power of the spindle, and base position and orientation. Simulating the robot kinematics consequently allowed the definition of the limits of the robotic workspace as the space free of collisions, singularities, and out-of-reach positions. Most notably, the setup defined the maximum possible plate diameter as a function of the clear width of the robot cell (see 7.4.3), as well as the minimum and maximum angles between plates.

The result was a *Machinic Morphospace* delineating the space of producible geometry, a sub-set of the geometrically possible space, as a function of the specific fabrication setup. Each building element thus constituted simultaneously a geometrically differentiated building element in 3-dimensional model space and a position in the n-dimensional design solution space/fabrication-constraint space. The limits of this space were delineated by the parameter ranges defined in response to the chosen material and fabrication approach.

During the design of the shell and its segmentation, the limits of the machinic morphospace could thus be explored while ensuring that all of the plates could be produced as efficiently as possible. Within the spatial limits of the robotic milling cell, the seven-axis robot then provided the degrees of freedom and the kinematic range necessary for the production of the complex joint details.

In order to meet the requirements of the project brief for an exhibition space and for an extended usage period from spring to fall, the build-up of the shell included, in addition to the structural layer, a vapor barrier, insulation, water-proofing, and a cladding layer held by timber slats. The insulation layer made from wood fibre board was to be produced in a 5-axis process and the cladding layer consisting of regionally sourced three-ply larch in a 3-axis process. The water-proofing layer, which consisted of individually tailored sheets of EPDM membrane, was to be produced in a 2-axis process using waterjet cutting.

The machine codes and fabrication information for all constructive layers were generated automatically from the same digital model according to the Principle of Generative CAM, ultimately allowing the entire shell to be digitally fabricated.

7.4.6.3 Principle chain

Five distinct principles have been identified in the fabrication development domain and defined as functional principles in B.6. Connected into a principle chain, they link the domain-specific inputs and outputs described above via the following sequence of functional principles:

- (B.6.1) the Principle of Generative CAM associates the automatic generation of machine tool paths with geometric variation represented by a B-rep design model, topology information and job-specific parameters via a generative modeling approach;
- (B.6.2) the Principle of 7-Axis Robotic Fabrication associates a kinematic model of a 7-axis robotic fabrication process with a specific machine setup via a mathematical model of the robot kinematics;
- (B.6.3) the Principle of Process Validity associates the validity of a fabrication process with specific fabrication strategies and a given fabrication setup via kinematic simulation of the process;
- (B.6.4) the Principle of Machine Code Generation associates the automatic generation of machine code with the tool paths of a previously validated process via the translation of geometric and alphanumeric attributes into machine-readable text;
- (B.6.5) the Principle of Machinic Morphospace associates the space of producible building elements with a validated kinematic simulation and given materials via the simulation of the robot kinematics.

7.4.6.4 Results

The analysis illustrates how the use of robotic fabrication in combination with other established CNC-equipment enabled the development of a custom, project-specific pre-fabrication sequence. In this sequence, the robotic fabrication setup in combination with the available material had the most significant effect on what could be produced and consequently on what could be designed. At the same time, the generative approach was paramount for producing what has been designed: enabled

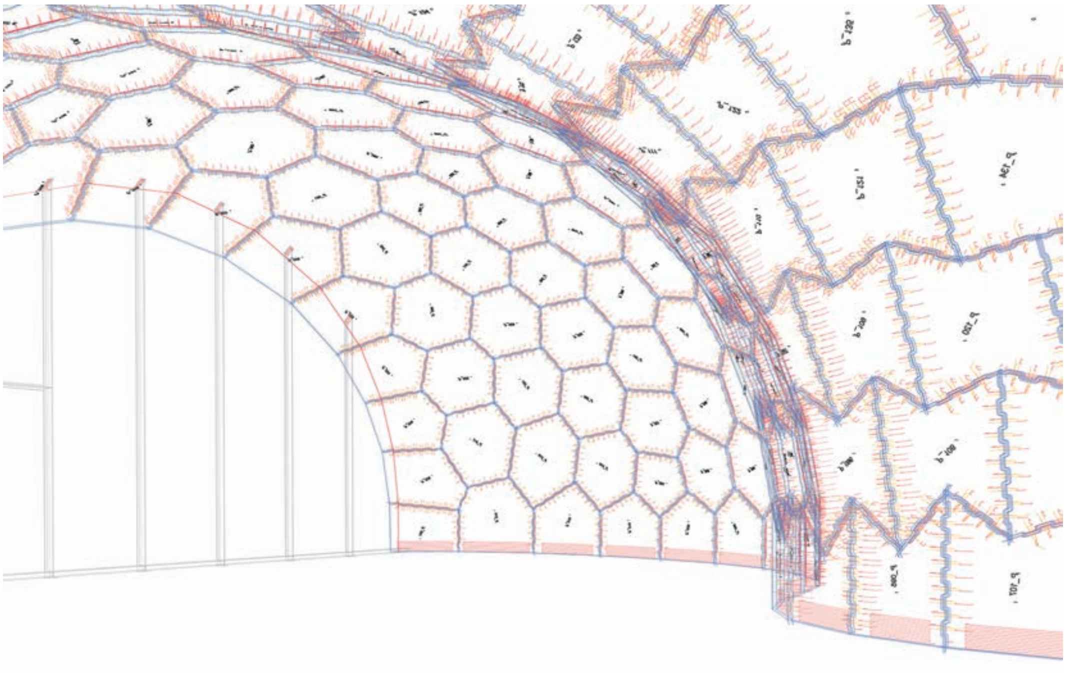


Figure 7.62: View of fabrication data model (Krieg, 2014).

by the automated CAM and machine code mechanisms, the information necessary for the production of all individual 243 plates in the shell and their 7,356 joints could be validated and generated automatically from the same digital model by writing directly to the respective machine code interfaces (Fig. 7.62). Using such a generative approach, in which the respective tool path logic could be applied to all geometrically unique plates simultaneously, not only highly facilitated fabrication modeling, but also ensured a consistent connection performance while staying inside the machine's morphospace [103, p. 120].

7.4.6.5 Significance

It is worth noting that due to the fabrication-oriented modeling approach, no more geometry information than the B-rep surface representation was needed throughout the whole digital design and fabrication process [172, p. 183]. The solid model representing the actual plates with finger joints has been generated after-the-fact, "down-stream" from the fabrication model and was only needed for visualization and quantity take-off purposes [172, p. 184]. This should be seen in contrast to established processes where the fabrication model is usually derived from a detailed geometric description of the final result representing the finished surface. The conventional approach however is problematic as the geometric model is then typically not informed by fabrication logic.

7.4.7 Quality Control Principles

The domain of quality control relates the validation of a fabrication process in terms of accuracy with fabrication setup and tolerance requirements, as well as findings on the dimensional behavior of Beech plywood with changing environmental conditions between fabrication and assembly and material handling via a specific sequence of low-level principle (Fig. 7.63).

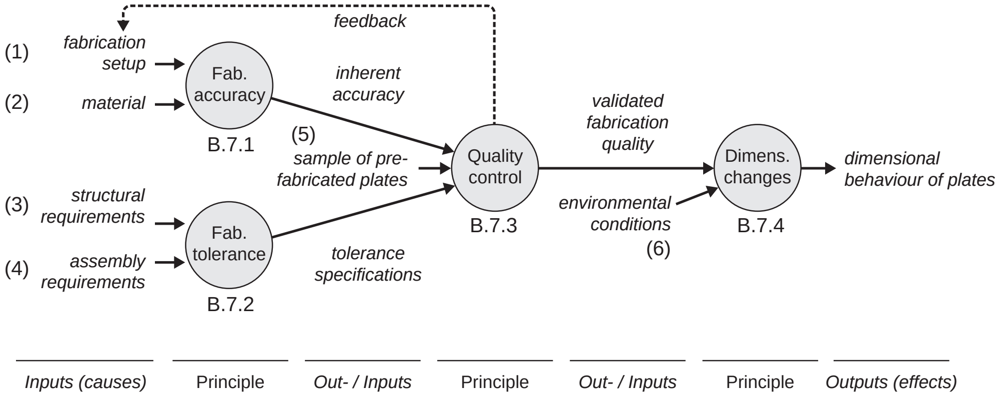


Figure 7.63: Chain of Quality Control principles.

7.4.7.1 Inputs

As mentioned before, the accuracy of a fabrication process is an important aspect that has repercussions on the structural behavior and, in the case of segmented shells, on the viability of a proposed structure. Based on structural and assembly requirements, tolerance values, that is the deviation from the nominal values that is tolerable, are defined – in this case 0.5 mm (in order to result in a maximum gap width of 1 mm) – and implemented in the fabrication model. The accuracy of the fabrication process therefore must be higher (the RMS error of fabrication lower) than the tolerance specifications. Quality control then establishes whether those specifications have actually been met. It is therefore particularly important when developing new fabrication and construction processes and, again, a crucial aspect for the relevance of the developed plate system in the building industry.

It is worth noting that these specifications are significantly higher than the typical requirements in timber construction where 4 mm tolerance per length of 1 m are acceptable (DIN18203-3). By consequence, the realization of timber plate structures requires much higher precision fabrication equipment than that which is currently established in the industry.

Input factors for quality control thus were the given fabrication setup and process, as well as joint gap requirements stemming from structural design and assembly. Furthermore, changing environmental conditions between workshop and construction site were expected to have effects on the highly responsive Beech wood.

7.4.7.2 Methods

In order to be able to make statistically relevant statements about fabrication accuracy, a sample of 24 out of the overall 243 robotically fabricated timber plates was analyzed using laser-tracking.

A custom designed probe was produced by the project partner IIGS for the tracking of the plate edges. The methods and tools for tracking, the comparison of the measurement data with the digital model, and the statistical analysis are described in detail in Krieg et al. [103, pp. 120-22] and Schmitt and Schwieger [169]. It is important to note that the fabrication accuracy, that is the 2-dimensional RMS error value, stated in [103, pp. 123] as 0.7 mm was revised to 0.42 mm 2D-RMS [169, p. 6] after further evaluation of the measurement data.

According to Krieg et al. [103, p. 123]), rough surfaces due to the milling process as well as slight dimensional changes in the material in response to changing environmental conditions might also have impacted accuracy. Notably, plywood's tendency to buckle and dish can have an effect on fabrication accuracy, due to the three-dimensional fabrication process [103, p. 122]. The behavior of the plates thus was to be monitored over time.

Measurements were repeated for four elements right after fabrication, before transportation (approximately two weeks after fabrication) and right before assembly thereby defining three so-called measurement epochs [103, p. 121]. The result of the comparison of the point measurements was that, even though dimensional changes could be detected in the different epochs, the variations were statistically insignificant [103, p. 122].

7.4.7.3 Principle chain

In the domain of quality control, four distinct development steps could be identified and defined as functional principles in B.7. When connected into a principle chain, they relate the validation of a fabrication process in terms of accuracy as well as findings on the dimensional behavior of Beech plywood with the inputs mentioned above via the following sequence of functional principles:

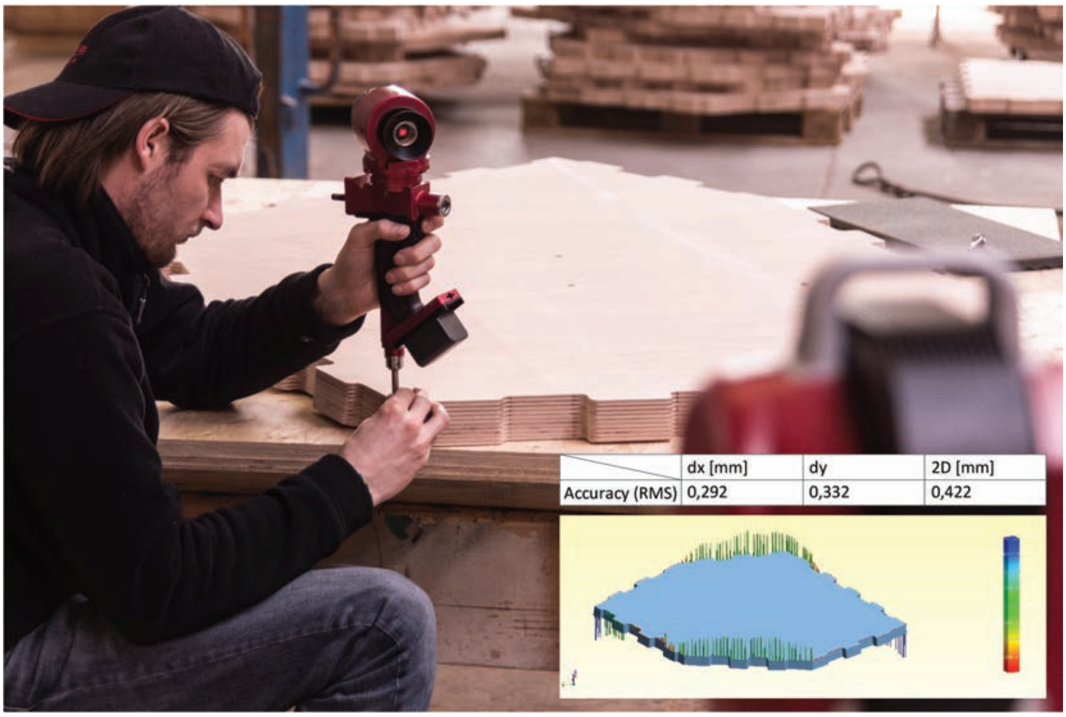
- (B.7.1) the Principle of Fabrication Accuracy associates the inherent accuracy of a fabrication process with the given fabrication setup and the material being machined via the elasticity of the equipment, for example the robot arm, the calibration accuracy of work piece and tool, and the wear of the tool;
- (B.7.2) the Principle of Fabrication Tolerance associates tolerance specifications for plate segments with structural and assembly requirements via a trade-off between assembly requirements (larger gaps) and global shell stiffness (smaller gaps);
- (B.7.3) the Principle of Quality Control associates the validation of the fabric-

ation result with the given tolerance requirements via accuracy measurement using laser-tracking and statistical analysis;

- (B.7.4) the Principle of Dimensional Changes associates findings on the dimensional changes of Beech plywood plates with changing environmental conditions via laser-tracking and comparison between the measured and the CAD model.

7.4.7.4 Result

The general conclusion of the quality control of the plates was that the resultant fabrication accuracy (RMS error of 0.4 mm) is not only significantly higher than the typical requirements in timber construction, but also within the specific tolerance specifications of joint development regarding assembly and structure: lower than the structurally required maximum gap width of 1 mm at the same time large enough so that the plates could be assembled (Fig. 7.64).



	dx [mm]	dy	2D [mm]
Accuracy (RMS)	0,292	0,332	0,422

Figure 7.64: Laser tracking of plate contours after fabrication. From [103] by Schmitt (2014).

7.4.7.5 Significance

Meeting the tolerance requirements of assembly and structure is the pre-condition for the viability of the shell. The findings thereby confirm that robotic fabrication is a suitable approach for meeting its advanced accuracy requirements.

It is interesting to note that due to the limited timeframe, the accuracy of the fabrication process could only be determined after fabrication had occurred. In other words, it was unknown if the accuracy of the process was high enough to be within the tolerance requirements until fabrication had been completed.

7.4.8 Off-site Pre-fabrication and Pre-assembly

The development domain of off-site pre-fabrication and pre-assembly associates assemble-ability and on-site delivery with machine setup, fabrication instructions, environmental conditions and quality control through the specific sequence of low-level principles (Fig. 7.65).

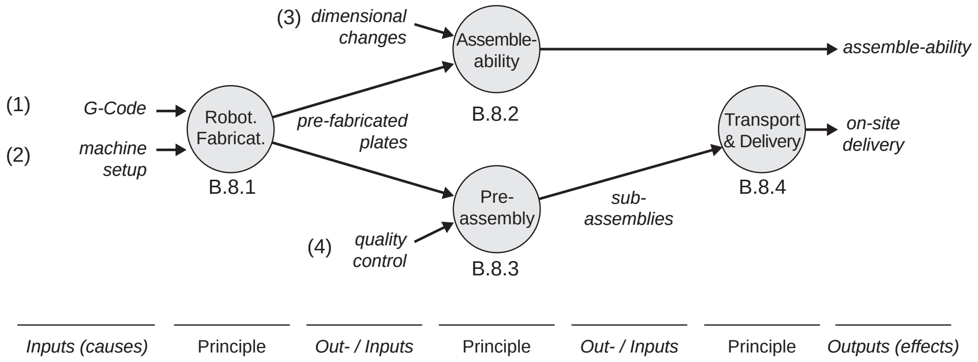


Figure 7.65: Chain of Off-site Pre-fabrication and Pre-assembly principles.

7.4.8.1 Inputs

Inputs to the domain are (1) the validated fabrication process including machine code, (2) the specific machine setup provided by KUKA, (3) dimensional changes of the plates due to changing environmental conditions, and (4) the findings of quality control.

7.4.8.2 Methods

In the first step of the fabrication process, the Hundegger Speed-Panel machine was used for pre-formatting the polygonal plates: they were cut with oversize in a three-axis process from rectangular beech plywood stock panels of dimensions of 2550 x 1850 x 50 mm. Where possible, up to two plates were nested on one stock panel in order to minimize cut-off. As the plates are all geometrically unique, it was crucially important that in this step the plates were also automatically labelled in order to be able to identify the plates later on in the fabrication and assembly process.

In the following step, pre-formatted plates were mounted on the turntable in the robotic milling cell where intricate finger joints where milled down to nominal dimensions, screw pockets were indented into the plywood, and fingers were spot-faced using the industrial robot-arm. Following fabrication, the 243 plywood plates were palletized in the order of assembly and vacuum-packed before being shipped on-site in order to avoid any dimensional changes due to changing relative humidity.

The insulation layer (consisting of a 35mm wood fibre-board) was produced on the previously mentioned Speed-Panel machine in a 5-axis process with mitered edges in order to ensure a tight fit between the segments. The water-proofing layer (consisting of individually tailored sheets of EPDM membrane) was produced by a separate contractor using waterjet-cutting. In order to reduce on-site labor and to ensure a consistent quality of the glued bond between EPDM and wood fibre board, they were pre-assembled in the workshop into an insulation panel sub-assembly. Additionally, timber slats were screwed to the panel as a sub-structure for the subsequent cladding layer. The cladding layer (consisting of regionally sourced three-ply Larch wood) was CNC-cut using the Speed-Panel machine in a three-axis process and prepared for on-site assembly. Ultimately all constructive layers in the shell could be digitally fabricated.

Off-site pre-fabrication entails transportation of building elements from the workshop to the construction site where elements and subassemblies are assembled into the full structure. Consequently, transportability and assemble-ability are a defining criteria for the viability of any pre-fabricated structure. In this case, assemble-ability is incorporated into the design process (i.e. joint development), but ultimately, it is also affected by dimensional changes due to changing environmental conditions between the workshop and the construction site. The effects of which were the subject of investigation in (7.4.7) Quality Control.

7.4.8.3 Principle chain

In the analysis described in B.8, four discrete development steps have been identified and defined as low-level functional principles. When connected into a principle chain, they associate assemble-ability and on-site delivery with with the domain inputs mentioned above via the following sequence of low-level principles:

- (B.8.1) the Principle of Robotic Fabrication associates pre-fabricated plates for finger-jointed plate structures with the use of a 7-axis robotic fabrication setup and valid machine code;
- (B.8.2) the Principle of Assemble-ability associates assemble-ability of the structure with joint geometry, joint gaps, and dimensional changes;
- (B.8.3) the Principle of Pre-assembly associates sub-assemblies ready to be shipped on site with the pre-fabricated elements that make up the shell and quality control;
- (B.8.4) the Principle of Transport and Delivery associates on-site delivery of building elements with pre-fabrication and pre-assembly.

7.4.8.4 Results

The developed fabrication process included all stages of fabrication for all construction layers of the plate structure [172, p. 187], which were pre-fabricated off-site and partially pre-assembled into sub-assemblies in the controlled production environment of the project partner's workshop in Blaustein near Ulm, Germany (Fig. 7.66).

Figure 7.66: Robotic fabrication of plates (Krieg, 2014).

Since the different machine codes necessary for the individual fabrication steps were generated automatically from the same digital model according to the Principle of Generative CAM (B.6.1), ultimately all constructive layers in the shell could be digitally fabricated.

7.4.8.5 Significance

While digital pre-fabrication was a highly automated process, pre-assembly still occurred completely manually. Further research, which followed the Landesgartenschau project, consequently focused on increasing the level of automation in timber construction particularly in the areas of pre-assembly through Human-Robot-Cooperation (HRC).

Nevertheless, off-site pre-fabrication confirmed the two main benefits that are typically associated with pre-fabrication such as higher quality (confirmed by quality control and through the perfect fit of all elements during assembly) and productivity in comparison with conventional on-site fabrication (through the high degree of automation), which consequently lead to a reduction of on-site labor and unnecessary down-time (through just-in-time delivery) and a clean construction site. Pre-fabrication (and pre-assembly), together with digital design and digital fabrication, can thus be identified as the critical concepts that enable the development of segmented shells and that are at the source of the renewed interest in shell structures.

Furthermore, it is important to note that as part of the realization of the project, a project-specific fabrication, assembly, and construction sequence was developed combining robotic fabrication with other numerically controlled equipment, in this case panel machine and waterjet cutter. In this fabrication sequence, robotic fabrication was strategically used for the accurate fabrication of the complex finger joints and thus represented the 'enabling' technology for the fabrication and assembly of the lightweight timber plate structure.

7.4.9 On-site Assembly and Construction principles

The domain of assembly and construction associates the fully assembled and finished building with stake-out data, a generic concrete slab as foundation, assemble-ability, and on-site delivery via the specific sequence of functional principles (Fig. 7.67).

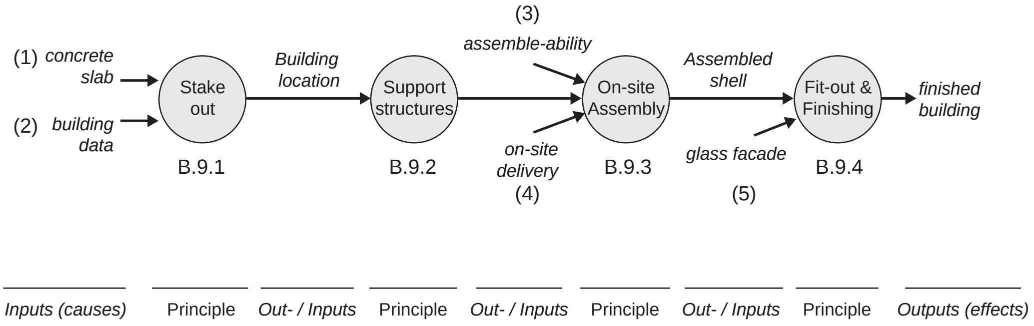


Figure 7.67: Chain of On-site Assembly and Construction principles.

7.4.9.1 Inputs

Inputs to the domain are (1) the concrete slab provided by the project partner Landesgartenschau Schwäbisch-Gmünd 2014 GmbH, (2) building data used for staking out the building on site, (3) assemble-ability of the pre-fabricated plates and pre-assembled insulation packages, (4) on-site delivery of building elements, and (5) the glass facade needed for providing an enclosed and weather-proof exhibition space.

7.4.9.2 Methods

In parallel to prefabrication and pre-assembly, the site, which is located just outside the city of Schwäbisch Gmünd (about 65 km from the fabricator's workshop in

Blaustein) was prepared for construction: a square slab of reinforced concrete with 16 m edge length and a slope of 1% for drainage was cast by a contractor on behalf of the Landesgartenschau project partner in the summer of 2013 prior to design and fabrication development of the shell. The slab was to provide a generic foundation for the Exhibition Hall.

Prior to the on-site assembly of the shell, the polygonal building outline was staked out by the IIGS on top of the concrete foundations. In the next step, a timber edge beam was laid out such that the outer faces of the beam's segments were aligned with the polygonal building contour and their upper faces were in the horizontal plane. The edge beam was then anchored to the concrete slab. Finally, L-shaped steel profiles were fastened to the edge beam, to which the bottom row of plywood plates could be attached through a dedicated pocket milled into the outer surface of the bottom-row plates.

While the timber edge beam provided a permanent support for the shell, a false work was installed as a temporary support (used only during assembly and later removed) retracing the inside surface of the plate shell in order to provide a reference for positioning each plywood plate.

The Beech plywood plates could then be assembled over the false work in the predefined assembly sequence: starting from the corner of the building located opposite to the main entrance, the installation proceeded, row by row, until the last plate was installed in the south-west corner of the building. The sequence was designed such that a plate would be attached to no more than two already installed plates in order to simplify assembly. While the assembly order was relatively flexible in the synclastic areas of the shell surface, the order had to be respected carefully in the anticlastic areas in order to avoid blockage as described in Joint Development (7.4.1).

Three workers were needed for the installation: one to operate a crane to deliver the plates, which weighted up to 60 kg, to the approximate location of assembly; a second worker standing on the platforms that also supported the false work received and carefully inserted the plate in the exact location; the third worker standing on the outside on a boom lift ("cherry picker"), first temporarily fixed the plate in the correct position in relation to the previously installed plates by applying half-threaded screws in the corners before permanently attaching the plate by connecting it to the adjacent plates using 120x6mm full-threaded screws. The screw angle was indicated unambiguously by the milled screw pockets on the outside surface of each plate. After completion of the structural layer of the shell, the vapor barrier was installed, followed by the insulation sub-assembly containing thermal insulation and waterproofing. After welding of the edges of the EPDM, the cladding layer was shipped and installed on-site leaving a shadow gap between the plates.

Finally, the remaining parts of the building were installed, such as lightning protection, columns that hold the glass façade followed by the glass façade itself, and finally the interior fit-out. The columns were necessary to provide support for the glass façade against wind loads and to limit the deformation of the free edges of the shell surface in order to avoid exceeding permissible stresses on the glass façade due to deflections of the shell.

In line with the overall goal of maximizing the utilization of the available building material, the off-cut generated during the pre-formatting of the beech plywood plates could be reused in the hardwood flooring as parquet lamellas.

On-site assembly and construction of the Landesgartenschau Exhibition Hall lasted only three weeks and proved to be highly reliable and robust. The completion of the additional interior fit-out and finishing added four more weeks to the construction schedule, such that the Hall and the surrounding landscaping were finished just in time for the opening of the landscaping show at the end of April 2014.

7.4.9.3 Principle chain

Four distinct development steps could be identified in the domain of assembly and construction and defined as low-level functional principles in B.9. When connected into a chain, they associate the fully assembled and finished building with the domain inputs mentioned above via the following sequence of functional principles:

- (B.9.1) the Principle of Stake Out associates the relative building location and orientation on site with corresponding data and a square concrete slab via surveying;
- (B.9.2) the Principle of Support Structures associates the permanent edge beam and the temporary false work with a staked out building outline on site;
- (B.9.3) the Principle of On-Site Assembly associates the assembled shell with assemble-ability, support structures, and on-site delivery of building elements; and finally
- (B.9.4) the Principle of Fit-out and Finishing associates the finished building with the assembled shell and installed glass façade via interior fit-out and finishing.

7.4.9.4 Result

All building elements and sub-assemblies were then assembled on-site during March 2014. In April 2014 interior fit-out and finishing was completed such that finished building demonstrator could be opened to the public at the end of the same month (Fig. 7.68).

Figure 7.68: Interior view of the finished Exhibition Hall (Reichert, 2014).

7.4.9.5 Significance

It is worth noting that assemble-ability, transportability and on-site handling of building elements usually impose constraints on the design and development of a structure. In this case, however, this applied only to assemble-ability: transportability and on-site handling were no constraining criteria to consider, as the maximum size and, consequently, weight of the segments was determined by the extents of the fabrication setup.

The realization of the project showed how robotic fabrication can become a valuable component in a fabricator's tool kit that not only integrates into existing fabrication technologies but, due to its flexibility, also opens up new applications [103, p. 124].

7.5 Third-level Abstraction

On the third level, principle chains are subsumed under high-level domain-specific functional principles thereby placing the focus on the interfaces between development domains. Using this approach, the 'vertical' relationship between low- and high-level principles and the 'horizontal' relationships between domains in the design, development, and realization process can be described.

The functional principles and their associations identified in the analysis of the design, development, and realization of the Landesgartenschau Exhibition Hall form a network of dependencies with multiple interfaces between different development domains and cross-domain feedback.

The following section aims to identify interfaces between domains and, based on the summary of principles, forms a dependency graph of functional principles that is specific to the design development of the Landesgartenschau Exhibition Hall. In a second step, feedback between development domains is identified based on the analysis of the dependency graph and described.

7.5.1 Summary of principles

In the analysis above (7.3), a total number of 48 functional principles have been identified. The following table (7.1) gives an overview on the related inputs and outputs. System-external inputs, such as machine setup, can be considered global parameters of the system, which not only includes (alpha-)numerical parameters and requirement definitions, but also geometric parameters, such as points and curves, that allow user interaction. Principle outputs that do not affect other domains, such as the resultant aesthetic quality of a finger joint connection or the finished building, are considered system-external outputs. In the table, both system-external inputs and outputs, are marked as ‘ex’ unless otherwise noted.

Table 7.1: Summary table of functional principles in Case Study II.

No.	Principle	Inputs from	Outputs to
B.1	Joint development principles		
B.1.1	Finger Jointing	ex	B.1.2+4, B.2.1, B.3.6
B.1.2	Shear-load Transfer	B.1.1	B.1.3, ex
B.1.3	Material Overlap	B.2.3, B.1.2	B.1.5, B.1.6, B.4.3
B.1.4	Geometric Compatibility	B.1.1, B.5.3, B.8.2	B.1.5, B.1.6
B.1.5	Joint Adaptation	B.1.3, B.1.4, B.3.9	B.1.6, B.6.1, B.2.2
B.1.6	Geometric Differentiation	B.1.3-5	ex
B.2	Structural joint principles		
B.2.1	Cross-laminated veneer	B.1.1, screws	B.2.2, B.2.3, B.7.1
B.2.2	Finger Joints	B.1.5, B.2.1	B.2.5, B.2.6
B.2.3	Crossing Screws	B.2.1	B.1.3, B.2.4+6, B.4.4
B.2.4	Load-testing	B.2.3	B.2.5
B.2.5	Joint gaps	B.2.2, B.2.4, B.8.2	B.2.6, B.7.2
B.2.6	Higher Stiffness	B.2.2, 3+5	B.4.3
B.3	Segmentation principles		
B.3.1	Triangulation	ex, B.4.5, B.5.5, B.6.3	B.3.2

Continued on next page

No.	Principle	Inputs from	Outputs to
B.3.2	Planar Approximation	B.3.1, planarity requ.	B.3.3-7, B.5.4
B.3.3	Planarity	B.3.2, planarity requ.	B.3.8
B.3.4	Valid Plate Size	B.3.2, stock material	B.3.8
B.3.5	Valid Plate Diameter	B.3.2, machine setup	B.3.8
B.3.6	Valid Plate Angle	B.3.2, B.1.1	B.3.8
B.3.7	Valid Edge Length	B.3.2, structural requ.	B.3.8
B.3.8	Segmentation Validity	B.3.3-7	B.3.9
B.3.9	Planarisation	B.3.8	B.1.5, B.4.3, B.6.1
B.4	Structural segmentation principles		
B.4.1	Trivalence	-	B.4.2
B.4.2	Co-linearity	B.4.1	B.4.4
B.4.3	Generative FE-Modelling	load cases, B.1.3, B.2.6, B.3.9	B.4.4, B.4.5
B.4.4	Added Bending Stiffness	B.2.3, B.4.2, B.4.3	ex
B.4.5	Force Alignment	B.4.3, loading	B.3.1, ex
B.5	Form-finding principles		
B.5.1	Edge Control	arch. & progr. requ., B.5.4	B.5.3, B.5.5
B.5.2	Local Controllers	arch. & progr. requ., B.5.4	B.5.3, B.5.5
B.5.3	Dynamic Equilibrium	B.5.1, B.5.2, structural requ.	B.1.4, B.5.4, B.5.5
B.5.4	Double Curvature	B.3.2, B.5.3	B.5.1, B.5.2, ex
B.5.5	Steering Form	B.5.1-3	B.3.1, B.9.1, ex
B.6	Fabrication development principles		
B.6.1	Generative CAM	B.1.5, B.3.9, ex	B.6.3
B.6.2	7-axis robotic fabrication	Machine setup	B.6.3, B.6.5
B.6.3	Process Validity	B.6.1, B.6.2	B.6.4-5, B.3.1
B.6.4	Machine Code	B.6.3	B.8.1
B.6.5	Morphospace	B.6.2-3, mat. parameters	producible form
B.7	Quality control principles		
B.7.1	Fabrication Accuracy	fabr. setup, B.2.1	B.7.3
B.7.2	Fabrication Tolerance	B.2.5, assembly requ.	B.7.3
B.7.3	Quality Control	B.7.1-2, B.8.1	B.7.4
B.7.4	Dimensional Changes	B.7.3, env. conditions	B.8.2
B.8	Offsite Fabrication and Pre-Assembly Principles		
B.8.1	Robotic Fabrication	B.6.4, machine setup	B.7.3, B.8.2-4
B.8.2	Assemble-ability	B.7.4, B.8.1	B.1.4, B.2.5, B.9.3
B.8.3	Pre-assembly	B.8.1	B.8.4

Continued on next page

No.	Principle	Inputs from	Outputs to
B.8.4	Transport & Delivery	B.8.1, B.8.3	B.9.3
B.9	Onsite Assembly and Construction Principles		
B.9.1	Stake-out	B.5.5, concrete slab	B.9.2
B.9.2	Support structures	B.9.1	B.9.3
B.9.3	On-site assembly	B.8.2, B.8.4, B.9.2	B.9.4
B.9.4	Fit-out and finishing	B.9.3, glass facade	finished building

Principle chains within development domains and their associations across domains form a network of mutual dependencies, including global inputs and outputs, that is visualized as a directed graph in Fig. 7.69.

Similar to the corresponding visualization in Case Study I (Fig. 6.55), the graph reveals that some of the domains that in conventional architectural planning would be considered ‘down-stream’, such as segmentation development, are shown to have had a significant influence on domains that are typically considered ‘upstream’, such as global design (in this case part of the Form-finding domain). On the other hand, joint development, in conventional architectural design considered ‘down-stream’ but in this case one of the starting points of the development process, is shown to have been influenced by what would typically be considered ‘up-stream’ domains, such as global design and segmentation development. The terms ‘up-stream’ and ‘downstream’ are used here in the relative sense: ‘up-stream’ denotes aspects that have been determined earlier in the development process and that influence current developments; conversely, ‘down-stream’ denotes aspects that are going to be determined later on in the development process and that are influenced by current developments. Where current developments are influenced by ‘down-stream’ aspects, a bottom-up relationship can be established as part of a feedback loop as described in 7.5.3.

7.5.2 Cross-domain integration

In the following, influence across domains, as described above, is termed *integration* such that domains are said to be ‘integrating’ domain-external aspects, if they are influenced by them. Defined in this sense and illustrated in Figure 7.69, integration can now be explicitly stated from the perspective of each development domain.

7.5.2.1 Joint development

The Joint Development domain (B.1) integrates aspects of structural joint development, segmentation development, form-finding, and off-site fabrication (B.2, B.3, B.5, and B.8) and itself influences structural joint development (B.2), segmenta-

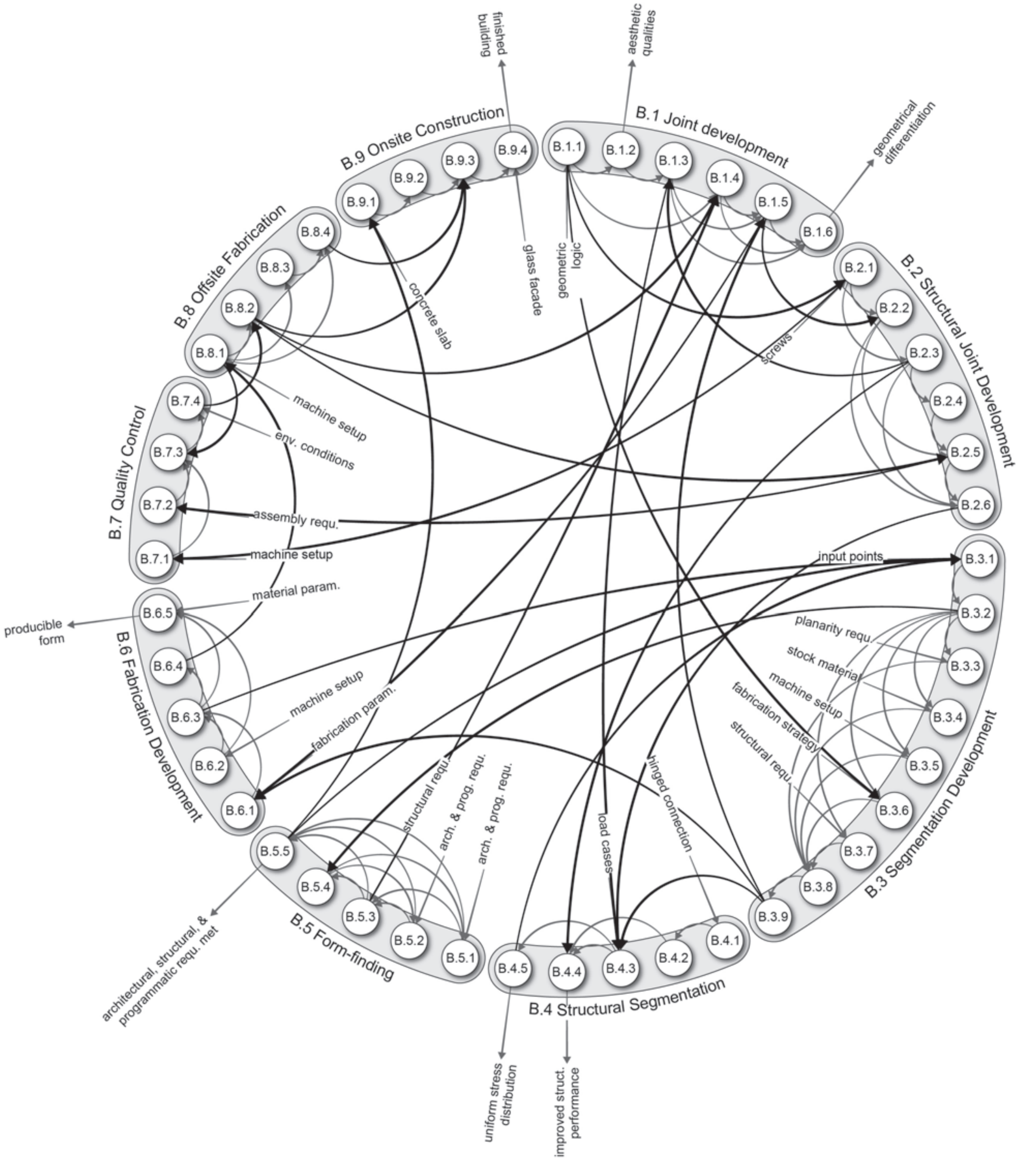


Figure 7.69: Interfaces and cross-domain feedback loops in Case B.

tion development (B.3), structural segmentation (B.4), and fabrication development (B.6). Joint development has a total of nine cross-domain connections and is in a bi-directional cross-domain relationship with structural joint development and segmentation development.

Specifically, the principle of Material Overlap (B.1.3) integrates the structural screw model of the principle of Crossing Screws (B.2.3); the principle of Geometric Compatibility (B.1.4) integrates double curvature of the principle of Dynamic Equilibrium (B.5.3) and the requirement to be assemble-able of the principle of Assemble-ability (B.8.2); and the principle of Joint Adaptation (B.1.5) integrates the geometrically differentiated planar segmentation of the principle of Planarization (B.3.9).

The geometric logic that is inherent in finger joints has been identified as a system-external input to the joint development domain. System-external outputs are the aesthetic quality of the connection as well as geometrical differentiation of the joint resulting from functional integration and adaptation.

It is worth noting that visualizing the dependencies, for example regarding assemble-ability, in Case A and B, illustrates the different roles of assemble-ability in both cases. It then becomes apparent that assemble-ability was a driver of joint development in B, whereas it was the result of joint and connection geometry on both levels of hierarchy in A.

7.5.2.2 Structural joint development

The domain of Structural Joint Development (B.2) integrates aspects of joint development (B.1) and Off-site fabrication (B.8) and itself influences joint development (B.1), structural segmentation (B.4), and quality control (B.7). It has a total of eight cross-domain connections and is in a bi-directional cross-domain relationship with joint development.

Specifically, the principle of Cross-laminated veneer (B.2.1) integrates the resulting teethed pattern of the principle of Finger jointing (B.1.1); the principle of Finger joints (B.2.2) integrates geometric parameters (width and depth) of the principle of Joint Adaptation (B.1.5); and the principle of Joint Gaps (B.2.5) integrates the requirement for gaps between the plates to ensure assemble-ability (B.8.2).

As a system-external input to the structural joint development domain the use of screws has been identified; no system-external outputs have been identified.

7.5.2.3 Segmentation development

The Segmentation Development domain (B.3) integrates aspects of joint development (B.1), structural segmentation (B.4), form-finding (B.5), and fabrication development (B.6) and itself influences joint development (B.1), structural segmentation (B.4), form-finding (B.5), in this case dictating double-curvature, and fabrication development (B.6). It has a total of eight cross-domain connections and is in a bi-directional cross-domain relationship with joint development, structural segmentation, form-finding, and fabrication development.

Specifically, the principle of Triangulation (B.3.1) integrates the requirement for pattern alignment of the principle of Force Alignment (B.4.5), the resulting double-curved surface of the principle of Steering of form (B.5.5), as well as the outcome of the validity check of the principle of Process Validity (B.6.3); and the principle of Valid Plate Angles (B.3.6) integrates the fabrication strategy of the principle of Finger Jointing (B.1.1).

System-external inputs of the domain are input points as the basis of triangulation as well as the various requirements stemming from the choice of material (planarity, stock sizes), machine setup, and structural requirements. No system-external outputs of the segmentation domain have been identified.

7.5.2.4 Structural segmentation

The Structural segmentation domain (B.4) integrates aspects of joint development (B.1), structural joint development (B.2), and segmentation development (B.3) and itself influences segmentation development. It has a total of five cross-domain connections and is in a bi-directional cross-domain relationship with segmentation development.

Specifically, the principle of Generative FE-Modeling (B.4.3) integrates the trivalent plate configuration of the principle of Planarization (B.3.9), the structural joint model of the principle of Higher Stiffness (B.2.6), and the resultant material thickness of the principle of material overlap (B.1.3); and the principle of Added Bending Stiffness (B.4.4) integrates the structural model of the principle of Crossing Screws (B.2.3).

The load cases that are applied in the FE-model (B.4.3) (based on the requirements for permanent buildings) and the hinged connection as an input to the principles of Trivalence (B.4.1) have been identified as a system-external inputs to the domain. For clarification, while the particular joint type developed in Case B is a hinged connection (with added bending stiffness), the “hingedness” that the Trivalence principle receives as input is independent of a particular joint type, hence the declaration as external input. System-external outputs are the improved structural performance as a result of the principle of Added Stiffness (B.4.4) and the more uniform stress distribution as a result of the principle of Force Alignment (B.4.5).

7.5.2.5 Form-finding

The domain of Form-finding (B.5) integrates aspects of segmentation development (B.3) and itself influences joint development (B.1), segmentation development (B.3), and On-site construction (B.9). It has a total of four cross-domain connections and is in a bi-directional cross-domain relationship with segmentation development.

Specifically, the principle of Double curvature (B.5.4) integrates the requirement for double curvature resulting from the use of the TPI method in the principle of Planar Approximation (B.3.2).

System-external inputs are architectural, structural, and programmatic requirements that the building has to meet. Conversely, the system-external output of the Form-Finding domain is that those system external requirements are actually met as the result of the principle of Steering of Form (B.5.5).

7.5.2.6 Fabrication development

The fabrication development domain (B.6) integrates aspects of joint development (B.1) and segmentation development (B.3), and itself influences off-site fabrication (B.8). It has a total of four cross-domain connections and is in a bi-directional cross-domain relationship with segmentation development.

Specifically, the principle of Generative CAM (B.6.1) integrates the trivalent plate configuration of the principle of Planarization (B.3.9), and the adaptive joint model of the principle of joint adaptation (B.1.5).

System-external inputs are fabrication parameters and machine setup. A system-external output is the space of producible geometry, aka Machinic Morphospace.

7.5.2.7 Quality Control

The domain of Quality control (B.7) integrates aspects of structural joint development (B.2), off-site fabrication (B.8), and itself influences off-site fabrication (B.8). It is thus in a bi-directional cross-domain relationship with off-site fabrication. It has total of four cross-domain connections.

Specifically, the principle of fabrication accuracy (B.7.1) integrates the material of the principle of cross-laminated veneer (B.2.1); the principle of Fabrication tolerances (B.7.2) integrates structural requirements of the principle of Joint gaps (B.2.5); and the principle of quality control (B.7.3) integrates a sample of the prefabricated plates of the principle of robotic fabrication (8.1).

System-external inputs are machine setup, assembly requirements, and environmental conditions. Assembly requirements are considered system-external because the exact assembly tolerances were set a priori and were not based, for example, on experiments or simulations such as in the case of structural requirements. No system-external outputs have been identified.

7.5.2.8 Off-site fabrication

The domain of Off-site Fabrication (B.8) integrates aspects of fabrication development (B.6) and quality control (B.7) and itself influences joint development (B.1), structural joint development (B.2), quality control (B.7), and on-site construction (B.9). It has a total of seven cross-domain connections and is in a bi-directional cross-domain relationship with Quality Control.

Specifically, the principle of Robotic Fabrication (B.8.1) integrates machine code of the principle of Machine Code Generation (B.6.4); and the principle of Assemble-ability (B.8.2) integrates dimensional changes of the plate due to changing environment conditions (B.7.4).

No system-external inputs or outputs have been identified.

7.5.2.9 On-site construction

Lastly, the domain of On-site Construction (B.9) integrates aspects of Form-finding (B.5) and Off-site fabrication (B.8). It does not influence any other domain in the system and is thus not involved in any bi-directional cross-domain relationship. It has a total of three cross-domain connections.

Specifically, the principle of Stake-out (B.9.1) integrates building data (such as its footprint and relative coordinates) of the principle of Steering of Form (B.5.5); the principle of On-site Assembly (B.9.3) integrates the ability to be assembled of the principle of Assemble-ability (B.8.2), as well as the building elements delivered on-site based on Delivery and Transportation (B.8.4).

System-external inputs are the square concrete slab used as a foundation and the glass facade. The domain also features the system's principal output, which is the finished building as the result of the principle of Fit-out and Finishing (B.9.4).

7.5.3 Cross-domain Feedback Loops and Concurrent Developments

The analysis of the dependency graph now allows for the identification of feedback loops between development domains. Similar to Case Study I, a feedback loop is defined as a bi-directional relationship between two domains A and B, where an output O_A of domain A is integrated by domain B at time t and an output O_B of domain B is, in turn, re-integrated into domain A at time $t + 1$ such that it changes output O_A in what effectively forms a loop in the flow of information (see 2.1).

Based on this definition, the analysis of integration between domains reveals that not all bi-directional relationships between domains can be considered feedback loops. In these cases, rather than the outputs of domain A effectively changing its own inputs, the outputs O_B of domain B feed back into domain A at a point where they do not affect the corresponding inputs of domain B. In this case, developments in different domains occur concurrently and are subsequently termed 'concurrent relationships'. The following description distinguishes between these two cases.

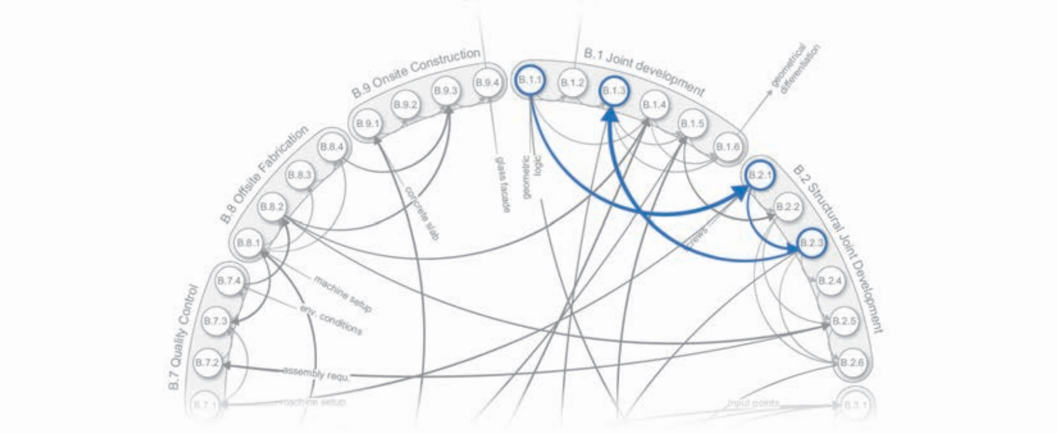


Figure 7.70: Concurrent relationship between joint development and structural joint development domains.

A **first concurrent relationship** exists between the domains of joint development (B.1) and structural joint development (B.2): the principle of Cross-laminated veneer (B.2.1) integrates the resulting teethed pattern of the principle of Finger jointing (B.1.1), while the principle of material overlap (B.1.3) integrates the structural screw model of the principle of Crossing Screws (B.2.3) without affecting B.1.1 (Fig. 7.70).

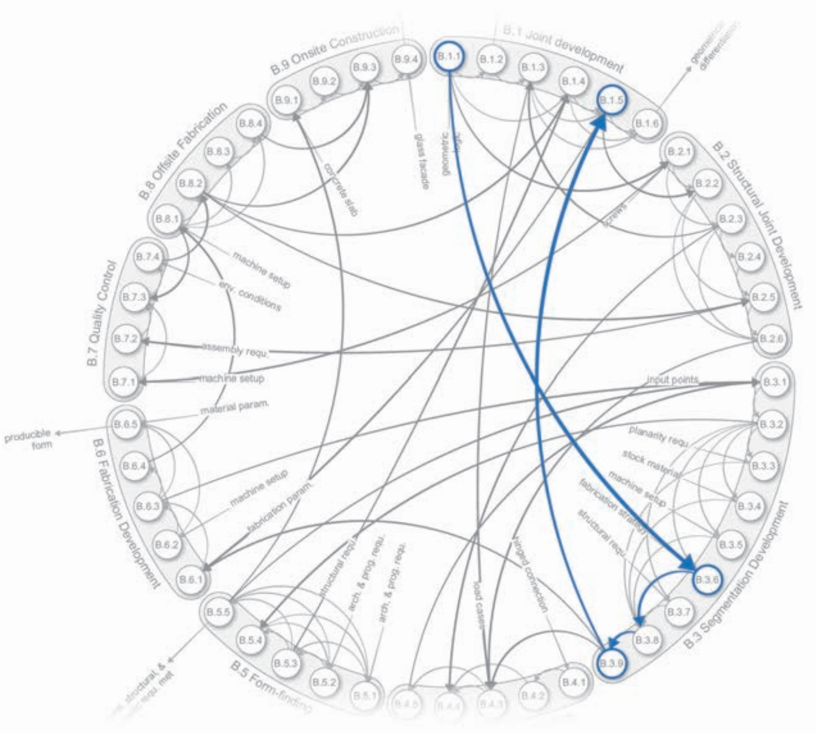


Figure 7.71: Concurrent relationship between joint development and segmentation development domains.

A **second concurrent relationship** exists between the domains of joint development (B.1) and segmentation development (B.3): the principle of valid plate angles (B.3.6) integrates the fabrication strategy of the principle of finger jointing (B.1.1) while the principle of joint adaptation (B.1.5) integrates the geometrically differentiated planar segmentation of the principle of Planarization (B.3.9) without affecting B.1.1. (Fig. 7.71).

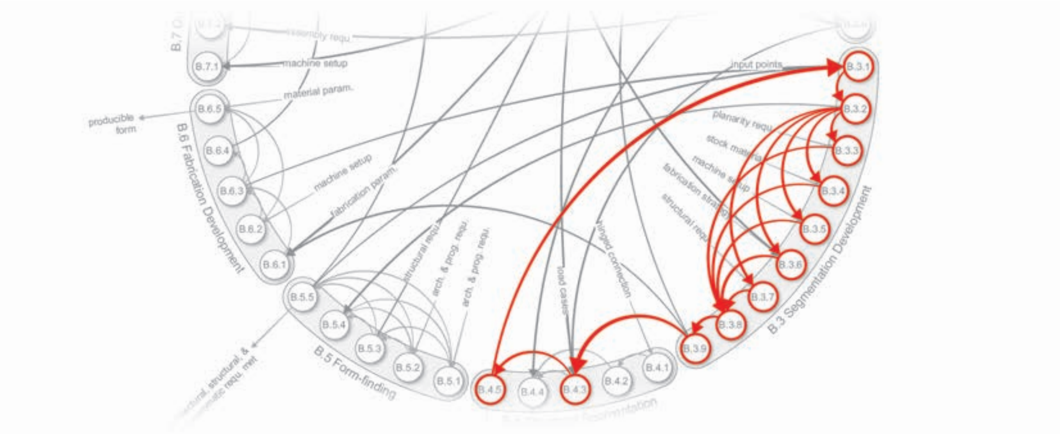


Figure 7.72: Feedback loop between geometric segmentation and structural segmentation development domains.

A **first feedback loop** exists between the domains of segmentation development (B.3) and structural segmentation (B.4): the principle of Generative FE-Modeling (B.4.3) integrates the geometrically differentiated planar segmentation of the principle of Planarization (B.3.9), while the principle of Triangulation (B.3.1) integrates the edge alignment rules of the principle of Force Alignment (B.4.5) thereby affecting B.3.9 (Fig. 7.72).

Figure 7.73: Feedback loop between segmentation development and form-finding domains.

A **second feedback loop** exists between the domains of segmentation development (B.3) and form-finding (B.5): the principle of Double Curvature (B.5.4) integrates the requirement for double curvature of the principle of Planar Approximation (B.3.2), while the principle of Triangulation (B.3.1) integrates the resulting double-curved surface of the principle of Steering of Form (B.5.5) thereby affecting B.3.2 (Fig. 7.73).

A **third feedback loop** exists between the domains of segmentation development (B.3) and fabrication development (B.6): the principle of Generative CAM (B.6.1) integrates the geometrically differentiated planar segmentation of the principle of Planarization (B.3.9), while the principle of Triangulation (B.3.1) integrates geometrical adjustments due to the principle of Process Validity (B.6.3) thereby affecting B.3.9 (Fig. 7.74).

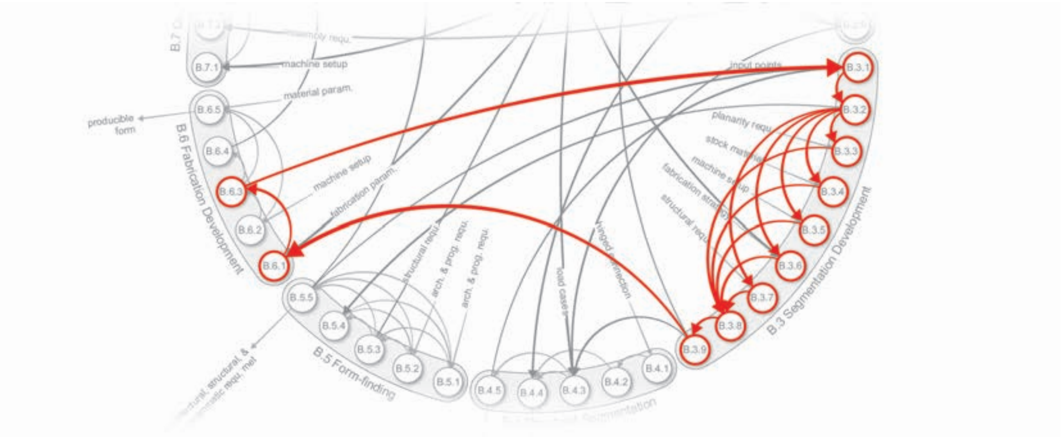


Figure 7.74: Feedback loop between segmentation development and fabrication development domains.

A **third concurrent relationship** exists between the domains of quality control (B.7) and off-site fabrication (B.8): the principle of Quality Control (B.7.3) integrates samples of pre-fabricated plates of the principle of Robotic Fabrication (B.8.1), while the principle of Assemble-ability (B.8.2) integrates dimensional changes of the plate (B.7.4) due to changing environment conditions without affecting B.8.1 (Fig. 7.75).

Furthermore, tracing the flow of information also reveals **multi-lateral feedback loops** in the development process of Case B, where more than two development areas are involved. This is the case between (1) the areas of structural joint development, quality control, and off-site fabrication; (2) joint-, fabrication development, and off-site fabrication; and (3) between joint-, structural joint-, and structural development (Fig. 7.76).

As stated before, the relevance of tracing the flow of information lies not only in the ability to discover feedback loops or to pinpoint feedback loops that would be expected from experience, but also in identifying the absence of relationships that would be expected. One example is the absence of a feedback loop between fabrication development (B.6) and quality control (B.7): generally, it would be expected

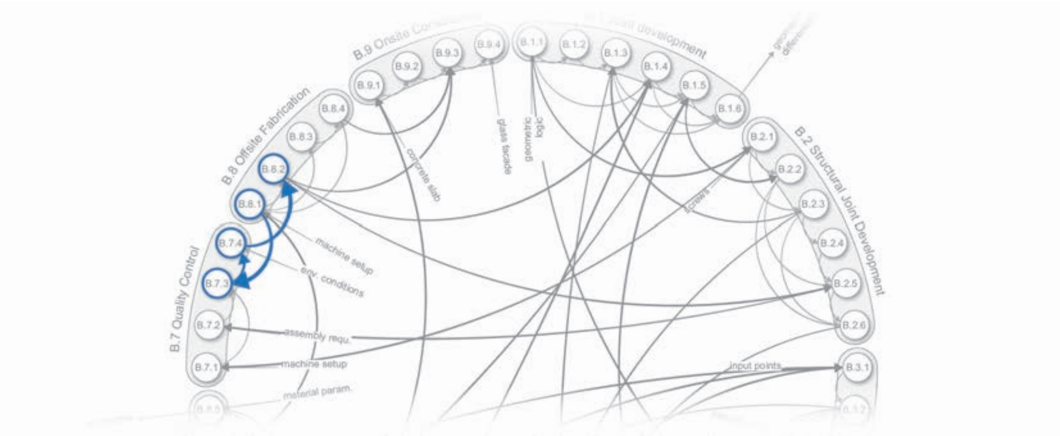


Figure 7.75: Concurrent relationship between quality control and off-site fabrication domains.

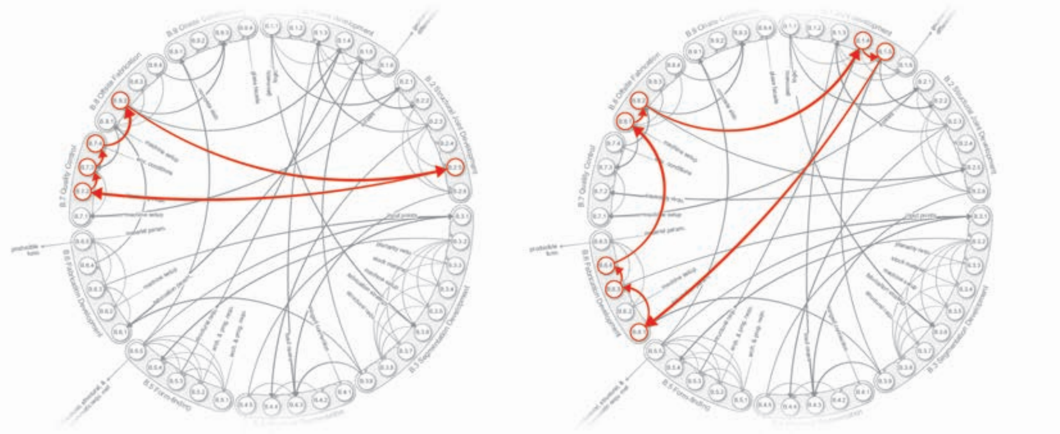


Figure 7.76: Multi-lateral feedback loops. Left: between cell-, design-, and structural development domains. Right: between cell-, design-, and fabrication development domains.

that tool paths were defined, as part of the principle of Generative CAM (B.6.1), based on the measured accuracy of the fabrication process as part of Quality Control (B.7.3). As mentioned before, however, the accuracy of the fabrication process was not determined until after fabrication had occurred due to the compressed project schedule. The flow diagrams above thus confirm that fabrication development did not receive feedback from quality control.

Again, these observations are specific to Case B and should not be mistaken for generally valid principles in segmented shell design. Rather, they constitute the specific findings from which a generalization will be attempted in Chapter 8 as part of an inductive research approach.

7.6 Results and Discussion

At the centre of this case study was the analysis of a full-scale segmented shell, the prototype building Landesgartenschau Exhibition Hall (Case B), with regard to functional principles in its development. Functional principles and their relationships within development domains and across domains have been identified based on information gathered from literature (see 7.1.3). As part of the analysis, high-level principles that are relevant beyond the specific case and bi-directional relationships between domains (feedback loops and concurrent relationships) could be identified. Specifically, the case study led to the following two principal findings:

1. A total of 48 functional principles have been identified, defined and categorized, and their interactions illustrated and discussed.
2. Based on the analysis of the network, six bi-directional, cross-domain relationships could be explicitly identified that were at the core of design development, of which three are feedback loops.

In addition, high-level principles could be found in the domains of structural joint development (B.2), segmentation design (B.3), Global design (B.5), and fabrication development (B.6). For example, the high-level principle of Higher Stiffness (B.2.6) increases the in-plane stiffness of edge-wise connections through the combination of finger joints (B.2.2), crossing screws (A.2.3), and joint gaps (B.2.5) that are necessary for assembly, using the mechanisms described in the respective lower-level design principles. A second instance is the high-level principle of Segmentation Validity (B.3.8), which ensures the validity of the segmentation by combining the mechanisms of the individual lower-level design principles, such as planarity (B.3.3), plate diameter (B.2.4), etc. A third instance is the high-level principle of Steering of form (B.5.5), through which structural, architectural and programmatic parameters are met by combining mechanisms of the lower-level principles of Edge Control (B.5.1), Local Control (B.5.2) and Dynamic Equilibrium (B.5.3) in order to steer the formation process of the shell. The fourth instance of a high-level principle is the principle of Machinic Morphospace (B.6.5), which defines the space of producible form by combining the lower-level design principles of Generative CAM (B.6.1), 7-axis robotic fabrication (B.6.2), and Process validity (B.6.3) with given material parameters. It is a pre-requisite for fabrication-oriented design integration.

Two of the most important findings of the case study, however, are the identification of six bi-directional cross-domain relationships, which have been made explicit, and the corresponding distinction between ‘concurrent relationships’ and ‘feedback loops’ in the development of the Landesgartenschau Exhibition Hall. Specifically, the feedback loops between the domains of segmentation development (B.3) and structural segmentation (B.4), between the domains of segmentation development (B.3) and form-finding (B.5), and between the domains of segmentation development (B.3) and fabrication development (B.6) constitute important findings for the further development of segmented shells for two reasons. First, they illustrate how segmentation development was located at the centre of the development of the Exhibition

Hall and, correspondingly, the role of synthesizing interdisciplinary dependencies that it played in its development. Second, the identification of feedback loops is particularly important as their implementation requires a computational method capable of feedback.

As described in Case Study I, while top-down relationships can be implemented through hierarchical modeling approaches, such as state-of-the-art associative parametric modeling, bottom-up relationships that incorporate feedback are difficult to implement (see 6.6). Consequently, agent-based modeling as an inherently bottom-up modeling approach that adopts a "microscopic perspective" has been proposed for the design of segmented shells. However, it is only now, through the analysis of the relationships in the development of segmented shells, that this proposition can be corroborated.

In other words, in the case of the Landesgartenschau Exhibition Hall, agent-based modeling was used hypothetically and feedback loops have been implemented intuitively based on implicit knowledge gained from previous work as opposed to based on explicitly stated functional principles and their relationships (as they have not been known at the time due to the lack of a methodology for the analysis of previously built segmented shells and their general scarcity).

7.7 Conclusion and Outlook

While the research that was conducted within the Robotics in Timber Construction research project, which led to the Exhibition Hall, already showed the viability of agent-based modeling for fabrication-oriented design of segmented shells, this case study now demonstrated a systematic approach for identifying and abstracting functional principles from real-world segmented shells. It thereby lays the foundation for defining validated design principles as the pre-condition for developing agent-based design models.

The next step towards instrumentalizing the identified functional principles therefore is to transfer functional principles to design principles and, further, to make recommendations for the development of agent-based design models for segmented shells. These recommendations address the questions of which design principles should be abstracted and transferred to the ABMS approach; and how design principles and their mechanisms map to the constituent elements of agent-based models, such as agents, interactions, topology, environment, and behaviors.

Both transfer and recommendations will be presented in the following chapter by combining findings from both case studies as part of defining *design principles for segmented timber shells*.

8

Design Principles for Segmented Timber Shells

8.1 Introduction

The aim of Part II of this dissertation was the identification of design principles for the design of segmented shells. This constitutes the pre-condition for the subsequent development, in which design principles will be implemented using the ABMS method.

Part II started with a definition of the term ‘principle’ in chapter 5 focussing on the distinction between systematic and axiomatic uses, which in the context of case studies were termed ‘functional’ and ‘design principles’ respectively. As part of the two case studies in chapters 6 and 7, functional principles were identified and categorized and their associations within and across development domains illustrated.

This chapter starts with synthesizing observations made in the case studies regarding the hierarchical relationships between development steps. In the following sections, the approach for identifying functional principles in the development of existing segmented shells is first summarized and then the translation from functional principles to design principles described. Transferring functional principles to design principles results in a preliminary catalogue of theoretical or ‘proto’-design principles that are specific to each case and that are evaluated with respect to their transferability. While the evaluation of transferability is to a certain degree subjective, it provides a filter for identifying those principles from this list of theoretically possible design principles that have the potential to be abstracted and generalized as design principles for segmented shells.

In order to establish generality of principles, the catalogues of design principles then are compared and recurring principles identified. Consequently, a selection of general principles for the design of segmented shells and their relationship to other principles is presented. This leads to the consideration of patterns of design principles, so-called “design pattern” for segmented shells.

Finally, recommendations are made as to how design principles map to the constituent elements of agent-based models. The Case Study part of the dissertation concludes with a discussion of the findings and the outlook to the agent-based implementation.

8.2 Bottom-up vs. Top-down Relationships

The conception of design as being influenced by ‘up-stream’ decisions influencing ‘down-stream’ aspects, borrowing the metaphor of water flow to describe the flow of information, suggests a uni-directionality of design from a higher to a lower level in a ‘top-down’ manner. However, as the case studies have shown, many ‘down-stream’ aspects can have an impact on ‘up-stream’ design decisions in a ‘bottom-up’ manner. Therefore, the terms ‘top-down’ and ‘bottom-up’ are considered to be a better characterization of these relationships by not only indicating uni- vs. bi-directionality of relationships, but also feedback loops.

In Case Study I, an example of a top-down relation is the principle of Machinic Morphospace (A.5.8) controlling the principle of Modular Plate Cell (A.2.4): the range of producible geometry defines the range of valid plate cells; however, the range of producibility is not affected by the results of plate cell generation. In other words, while considering aspects of fabrication suggests a bottom-up approach, by setting fabrication first in the development process, it can effectively end-up in a top-down relationship with other aspects of shell development if it is not affected by them.

Conversely, the example of a bottom-up relationship describes a feedback loop: the principle of Anisotropy (A.3.6) structurally evaluates the result of the principle of Steering of Form (A.3.4), using the principle of Automated FEM (A.4.3), and subsequently affects form generation parameters, such as the number and location of input points in the principle of Topological Control (A.3.1).

In Case Study II, an example of a top-down relationship is the principle of Planarization (B.3.9) affecting Joint Adaptation (B.1.5): the adaptive joint model implements the geometric variation resulting from segmentation design; however, segmentation is not affected by the adaptive joint model, but rather by the fabrication sequence defined in Finger Jointing (B.1.1). This leads to the surprising observation that, instead of starting the design and development process of the segmented shell with joint development, joint development could as well have occurred after global design and concurrently with segmentation development without affecting the degree of integration in the development of the structure. This, in turn, leads to the conclusion that the design and realization of segmented shells might well be implemented within the established stages of project delivery, which would significantly lower the threshold of adoption.

These examples show that the order number of a principle does not necessarily indicate its hierarchical relationship to other principles. In Case Study II, for example, the numbering of principles reflects the fact that joint development constituted one of the starting points of the project: Joint development (B.1) integrates many aspects of the development process, e.g. from domains B.3 and B.5, in a top-down way without affecting them. Segmentation development (B.3), on the other hand is in a bottom-up relationship with structural development (B.4) and fabrication development (B.5). In other words, this suggests that the relationship between principles is not subject to the specific premises that were made at the beginning of a research project and feedback loops occur independently of those premises.

8.3 From Functional Principles to Design Principles

Defining principles for the design of segmented timber shells is conceived of as a three step process consisting of analysis, abstraction, and translation. The development of this process is aimed at generalization, both in terms of moving from the specific to the general and in the sense that it is aimed to be applied to different cases of built segmented shells as a method for inducing generally valid design principles from specific cases.

In the **analysis** step, a chosen segmented shell is analyzed aiming at the identification of individual functional principles through the study of appropriate source material, such as literature or project data, by mapping the outcome of individual development steps to required inputs via specific mechanisms. A general example of such a functional relationship is output $z = f(x, y)$, where z is a function of x and y variables, which is parametrized by a, b, c, etc. as in the example of a quadratic function $f(x) = ax^2 + bx + c$ (see definition of functional principles in 5).[1]

In the **abstraction** step, functional principles are abstracted by grouping them into development domains and by connecting their outputs and inputs to form inference chains that link the outputs of a development domain to its inputs. On a higher level of abstraction, relationships between domains are then described using a similar graph-based approach. The resulting graph is termed ‘dependency graph’ as it describes the dependency of principles within the network. Contrary to existing

---

[1] Such a function can be considered a model of a real-world phenomenon, as in the example of fitting parameters of a polynomial curve through data points representing housing prices.

uses of the term ‘dependency graph’ in associative modeling software packages, such as Maya®, it is used as an analysis tool that allows for identifying feedback loops (circular causality) within the graph and for identification of ‘first principles’ (in the original sense of the word), from which the development process originates. Building a dependency graph forms a unique representation of a case, which can be a tool for further topological comparison between different segmented shells.

The abstraction steps thereby not only identify vertical relationships between low-level principles and high-level domains and but also horizontal relationships between domains. As part of the horizontal relationships, bi-directional relationships between domains can be identified and, specifically, a distinction can be made between concurrent relationships and feedback loops:

- A ‘feedback loop’ is defined in this context as a bi-directional relationship between two domains A and B, where an output O_A of domain A is integrated by domain B at time t, and an output O_B of domain B is re-integrated into domain A at time $t+1$ such that it changes output O_A in what effectively forms a loop in the flow of information (circular causality).
- The term ‘concurrent relationship’, on the other hand, describes bi-directional relationships between domains where developments occur concurrently, as the name suggests, in a way where the outputs O_B of domain B feed back into domain A at a point where they do not affect the corresponding inputs of domain B.

Establishing principle relationships across development domains and disciplines has made implicit connections explicit. Both aspects, the identification of feedback loops and making them explicit, are particularly important for three reasons:

1. Together with the number of cross-domain connections, they provide a quantitative measure of centrality of a specific development domain to the development of the entire segmented shell and thus illustrate the role of the domain in synthesizing interdisciplinary dependencies.
2. The identification of feedback loops is particularly important as their implementation requires a computational method capable of feedback; in other words, only when their locations in the development process, their attributes, and their connections are known can they be implemented.
3. Identifying feedback loops in the flow of information also allows identifying the absence of feedback loops in the conventional design process that contribute to the challenges that the building industry is facing.

The first two steps of analysis and abstraction, which have been described as part of two previous case studies, essentially form a bottom-up approach towards identifying functional principles and functional relationships between them and, on a higher level of abstraction, between development domains.

The third step in the definition of design principles is the **translation** step at which point the output of a functional principle is defined as a design objective: while, conceptually, the causality of the functional principle is inverted, the principle at this point becomes a vehicle that allows matching a given design objective to actionable mechanisms and minimum prerequisites. Furthermore, its prerequisites then become design objectives that might suggest other design principles that can be connected to the current principle to form a chain of design principles. In this way, individual principles or parts of principle chains can be appropriated in different design contexts as 'design patterns' and recombined as long as the corresponding inputs and outputs can be matched.

The translation step and the forming of chains of design principles is the subject of the following three sections.

8.4 Design Principles of Case A

In order to instrumentalize systematic functional principles observed in Case A for the design of segmented shells, relevant principles need to be identified that have the potential to be used as axiomatic design principles. 'Axiomatic' means that they don't need to be proved again every time they are used.

Since functional principles are defined by their mechanism and the minimum number of inputs that consistently produce the observed effect, a way to define design principles is to ask: "which outcome of a functional principle corresponds to a design objective for segmented shells?" and consequently to invert the causality on the level of the individual functional principle. An objective can then be matched to a mechanism of a functional principle that produces the desired result while also stating the minimum prerequisites to achieve the result.

The following tables in this section show 'theoretical' design principles organized by development domain that can be defined by applying this approach to the list of functional principles defined in 6.3. Additionally, the last column indicates the transferability T of a given design principle: h corresponds to a high transferability meaning that it is considered generic enough for the design of segmented shells; l corresponds to a low transferability meaning that the principle is considered specific to the case and therefore unlikely to be used in other segmented shell designs.

It is worth noting that, in a similar but inverse logic to the principle chains above, prerequisites of a design principle define the objectives of subsequent principles in the principle chains. For example, if the objective is to have a finished segmented shell then prerequisites are to have pre-assembled plate cells and an assembly sequence, which in turn become objectives that can be assigned to corresponding design principles. In this way, a designer would step through the chains of functional principles in reverse order.

Starting with **Joint Development** (6.4.1), five theoretical design principles have been identified, all of which have a high transferability as they have a high potential to be used in different contexts (Tab. 8.1). The corresponding flow diagram, which is inverted, is figure 6.43.

Table 8.1: Theoretical joint design principles.

Objectives	**Principle**	**Prerequisites**	*T*
Bending-free edges	Bending-free (in-plane forces)	Shear-load transfer capacity, trivalence	h
Shear-load transfer capacity	Shear-load (Interlocking)	Integral plate connection, range of valid angles	h
Range of valid angles	Connecting Angles	Kinematic model, given fabrication sequence	h
Integral plate connection	Joint Production	Edge-wise connection, fabrication sequence	h
Fabrication sequence	Finger Jointing	Geometric logic, subtractive fabrication	h

In the domain of **Cell Development** (6.4.2), the principles of Parametric Proxy and of Cellular morphology have a high transferability, while the remaining principles have a low transferability meaning that they are specific to this particular case and are unlikely to be used in different contexts (Tab. 8.2). The corresponding flow diagram, which is inverted as part of the definition of design objectives, is figure 6.45.

Table 8.2: Theoretical cell design principles.

Objectives	**Principle**	**Prerequisites**	*T*
Two-level hierarchy	Hierarchy	Polyhedral plate cell, trivalent 3D cellular model	l
Polyhedral plate cell	Plate cell	Trivalent 3D cellular model, trivalent plate configuration	l
Trivalent plate configuration	Frustum	Reference mesh, range of valid angles	l
Trivalent 3D cellular model	Parametric proxy	Trivalent 2D cellular model	h

Continued on next page

Objectives	**Principle**	**Prerequisites**	*T*
Trivalent 2D cellular model	Cellular morphology	Input points in 2D plane, logistic requirements	h

In the **Design Development** domain (6.4.3), all principles are considered to have a high transferability (Tab. 8.3). The corresponding flow diagram, which is inverted as part of the definition of design objectives, is figure 6.47.

Table 8.3: Theoretical global design principles.

Objectives	**Principle**	**Prerequisites**	*T*
Steering of form	Steering of form	Control parameters, design objectives	h
Given resources	Plate Coverage	Full polyhedral model	h
Morphological adaptation	Cell sizes	Full polyhedral model	h
Morphological adaptation	Anisotropy	Structural evaluation, full polyhedral model	h
Full polyhedral model	Global Cell Arrangement	Trivalent 3D cellular model, polyhedral plate cell	h
Trivalent 3D cellular model	Dynamic Equilibrium	Triangular mesh, Adaptive boundary	h
Adaptive boundary	Edge control	Programmatic requirements, 3d curve	h
Triangular mesh	Topological control	Input points in 2D plane	h

In the domain of **Structural Design** (6.4.4), all principles have high transferability with the exception of the principle of Stiffness Increase, which is tied to the specific hierarchical design of the plate cell (Tab. 8.4). The corresponding flow diagram, which is inverted as part of the definition of design objectives, is figure 6.49.

Table 8.4: Theoretical structural design principles.

Objectives	**Principle**	**Prerequisites**	*T*
Global bending bearing structure	Free-form shells	Convex plates, complex FE-model, two-level hierarchy	h
Uniform stress distribution	Stress reduction	Complex FE-model, convex plates	h
Increased stiffness	Stiffness increase	Two-level hierarchy, complex FE-model	l

Continued on next page

Objectives	**Principle**	**Prerequisites**	*T*
Complex FE-model	Automated FEM	Validated structural model, complex polyhedral model	h
Validated Structural model	Structural validation	Structural model, finger joint samples	h
Structural model	Trivalence	Biological plate structure	h

In the domain of **Fabrication Design** (6.4.5), all principles have a high transferability with the exception of Open CNC (Tab. 8.5). The corresponding flow diagram, which is inverted as part of the definition of design objectives, is figure 6.51.

Table 8.5: Theoretical fabrication design principles.

Objectives	**Principle**	**Prerequisites**	*T*
Range of producible geometry	Morphospace	Machine setup, Fabrication model	h
Machine-specific code, Validated fabrication process	Fabrication validation	Kinematic model, ISO Code	h
Brand independence	Open CNC	ISO Code	l
Kinematic model	7-axis fabrication	Machine setup	h
ISO Code	NC-Code generation	Fabrication model	h
Fabrication model	Automated CAM	Generic fabrication model, Data structure, Polyhedral model	h
Data structure	Scalability	Geometric complexity	h
Generic fabrication model	Finger joint modelling	Fabrication sequence	h

Finally, in the domain of **Production Design** (6.4.6), all principles also have a high transferability (Tab. 8.6). The corresponding flow diagram, which is inverted as part of the definition of design objectives, is figure 6.53.

Table 8.6: Theoretical production design principles.

Objectives	**Principle**	**Prerequisites**	*T*
Finished building, minimal form-/falsework	On-site assembly	Pre-assembled plate cells, Assembly sequence	h

Continued on next page

Objectives	**Principle**	**Prerequisites**	T
Pre- assembled plate cells	Pre-assembly	Pre-fabricated plates, Assemble-ability	h
Assemble-ability	Design for assembly	Finger-joint geometry, Plate & cell arrangement	h
Pre-fabricated plates	Robotic pre-fabrication	Validated fabrication process, 7-axis robot setup	h

In the above evaluation of 36 theoretical design principles, a total of 32 principles of high transferability have been identified. With their design objective stated, their mechanisms described in the methods section of functional principles, and the necessary pre-requisites (inputs) defined (see 6.3), those principles can now be considered validated *design principles*.

8.5 Design Principles of Case B

Similar to Case A above, in order to instrumentalize functional principles observed in Case B for the design of segmented shells, those functional principles need to be identified that have the potential to be used as design principles. Design principles are similarly defined by ‘inverting the causality’ of functional principles such that the outcome of a principle is stated as a design objective. An objective is thereby matched to a mechanism that produces the desired result while stating the minimum prerequisites to achieve it.

In the following tables (8.7-8.15) this inversion is applied to all functional principles that have been identified above as the basis for an evaluation of the design potential of each principle. The last column indicates the transferability T of a given design principle: h corresponds to a high transferability meaning that it is considered a design principle of general validity for the design of segmented shells; l correspondingly signifies a low transferability meaning that the principle either is specific to the case and unlikely to be used in other segmented shell designs or the outcome of the functional principle is not desirable as an design objective, as in the case of the principle of Co-linearity (B.4.2). In such instances, a short explanation is given as to why a principle is considered of low transferability.

Starting with the **Joint Development** domain (7.4.1), five theoretical design principles of high transferability have been identified (Tab. 8.7). The corresponding flow diagram, which is inverted as part of the definition of design objectives, is figure 7.51.

It is worth noting that geometric differentiation, which is not limited to the joint expression but also includes the geometry of the plates, is not considered a design objective per se as it is, in this case, the result of the shape adaptability of the joint model to the various constraints. Therefore, while it is an important functional principle, in this context it is considered of low importance as design principle and, consequently, of low transferability.

Table 8.7: Theoretical joint design principles derived from Case Study II.

Objectives	**Principle**	**Prerequisites**	*T*
Geometrically varied joint	Geometric Differentiation	Adaptive Joint model	*l*
Adaptive joint model	Joint Adaptation	Varying plates angles; variable edge orientation; Screw locations	*h*
Variable edge orientation	Geometric Compatibility	Edge-wise connection with teethed pattern; assembly sequence; surface curvature	*h*
Minimum plate thickness	Material Overlap	Shear-load bearing capacity; metal fasteners	*h*
Shear-load bearing capacity	Shear-load Transfer	Edge-wise connection with teethed pattern	*h*
Edge-wise connection with teethed pattern	Finger Jointing	Geometric logic; subtractive fabrication	*h*

In the **Structural Joint Development** domain (7.4.2), six theoretical design principles of high transferability have been identified (Tab. 8.8). The corresponding flow diagram, which is inverted as part of the definition of design objectives, is figure 7.53.

Table 8.8: Theoretical structural joint principles.

Objectives	**Principle**	**Prerequisites**	*T*
Increased in-plane stiffness	Higher Stiffness	Combination of finger joint, crossing screws, and joint gaps	*h*
Characteristic joint properties	Joint Gaps	Characteristic value of in-plane shear resistance and of withdrawing capacity	*h*
Characteristic values of withdrawing capacity	Load Testing	Model of stiffness	*h*
Model of connection stiffness	Crossing Screws	Material parameters (Beech)	*h*
Characteristic value of in-plane shear resistance	Finger Joints	Geometric parameters; material parameters (Beech)	*h*

Continued on next page

Objectives	**Principle**	**Prerequisites**	*T*
Use of Beech plywood	Cross-laminated Veneer	Edge-wise connection; metal fasteners	*h*

In the **Segmentation Development** domain (7.4.3), nine theoretical design principles of high transferability have been identified (Tab. 8.9). The corresponding flow diagram, which is inverted as part of the definition of design objectives, is figure 7.55.

Table 8.9: Segmentation design principles.

Objectives	**Principle**	**Prerequisites**	*T*
Planar, trivalent plate configuration	Planarisation	Valid segmentation	*h*
Segmentation validity	Segmentation Validity	Combined validity of below validity principles	*h*
Edge length validity	Valid Edge Length	Structural requirements; polygonal approximation	*h*
Angle validity	Valid Plate Angle	Fabrication strategy; polygonal approximation	*h*
Diameter validity	Valid Plate Diameter	Machine setup; polygonal approximation	*h*
Size validity	Valid Plate Size	Stock material; polygonal approximation	*h*
Planar validity	Planarity	Planarity requirement; polygonal approximation	*h*
Polygonal approximation of double curved surface	Planar Approximation (using TPI)	Topological relation; planarity requirement	*h*
Topological relation	Triangulation	Input points on surface; double-curved surface; segmentation validity	*h*

In the **Structural Segmentation Development** domain (7.4.4), four theoretical design principles of high transferability have been identified. The flow diagram, which is inverted as part of the definition of design objectives, is figure 7.57.

It is important to note that locally bending-weak joints do not constitute a design objective per se. Rather, they are the outcome of the functional principle of Co-linearity. The subsequent application of crossing screws is a measure to mitigate the effects of co-linearity by adding bending stiffness, which, consequently, is a design principle of high transferability (Tab. 8.10).

Table 8.10: Structural segmentation principles.

Objectives	Principle	Prerequisites	*T*
Uniform stress distribution	Force Alignment	Detail FE-model; loading	*h*
Improved structural performance	Added Bending Stiffness	Locally bending-weak joints; crossing screws; detailed FE-model	*h*
Detailed FE-model	Generative FE-Modelling	Joint model; planar segmentation; load cases	*h*
Locally bending-weak joints	Co-linearity	Co-linear edges; model of bending stiffness	*l*
Model of bending stiffness	Trivalence	Hinged connections; trivalent plate configuration	*h*

In the **Shell Form-finding** domain (7.4.5), five theoretical design principles of high transferability have been identified (Tab. 8.11). The corresponding flow diagram, which is inverted as part of the definition of design objectives, is figure 7.59.

Table 8.11: Form-finding principles.

Objectives	Principle	Prerequisites	*T*
Meeting structural, architectural and programmatic requirements	Steering Form	Combined principles of Edge control, local control, and dynamic equilibrium	*h*
TPI compliance	Double Curvature	Form-found surface; curvature requirement	*h*
Form-found surface	Dynamic Equilibrium	3d boundary curves; structural requirements; design forces	*h*
Design forces	Local Controllers	Architectural and programmatic requirements; curvature validity	*h*
3-dimensional boundary curves	Edge Control	Architectural and programmatic requirements; curvature validity	*h*

In the domain of **Fabrication Development** (7.4.6), five theoretical design principles of high transferability have been identified (Tab. 8.12). The corresponding flow diagram, which is inverted as part of the definition of design objectives, is figure 7.61.

Table 8.12: Fabrication design principles.

Objectives	**Principle**	**Prerequisites**	*T*
Space of producible form	Machinic Morphospace	Robotic fabrication, and Process validity; material parameters	*h*
G-Code	Machine Code	Validated fabrication process	*h*
Validated fabrication process	Process Validity	Tool paths; kinematic model	*h*
Kinematic model of fabrication equipment	7-axis Robotic Fabrication	Machine setup; defined quality criteria	*h*
Machine tool paths	Generative CAM	Reference geometry; fabrication parameters; valid process	*h*

In the domain of **Quality Control** (7.4.7), three theoretical design principles of high transferability have been identified (Tab. 8.13). The flow diagram, which is inverted as part of the definition of design objectives, is figure 7.63.

Similar to the previous instance of “locally bending-weak joints”, the dimensional behavior of plates is the result of the functional principle of Dimensional Changes (B.7.4), which is an important principle, however, it is not considered a design principle of high transferability. Rather the measures to mitigate dimensional changes, such as keeping environmental conditions as constant as possible, for example by shrink-wrapping batches of plates before they are shipped on-site as it was done in this case, are considered important in this context.

Table 8.13: Quality control principles.

Objectives	**Principle**	**Prerequisites**	*T*
Dimensional behaviour of plates	Dimensional Changes	Vetted fabrication quality; environmental conditions	*l*
Vetted fabrication quality	Quality Control	Prefabricated plates; tolerance specifications	*h*
Tolerance specifications	Fabrication Tolerance	Structural and assembly requirements	*h*
Inherent accuracy	Fabrication Accuracy	Fabrication setup; material	*h*

In the domain of **Off-site Fabrication and Pre-Assembly** (7.4.8), four theoretical design principles of high transferability have been identified (Tab. 8.14). The corresponding flow diagram, which is inverted as part of the definition of design objectives, is figure 7.65.

Table 8.14: Off-site fabrication and pre-assembly principles.

Objectives	**Principle**	**Prerequisites**	*T*
On-site delivery	Transport and Delivery	Sub-assemblies	*h*
Sub-assemblies	Pre-Assembly	Pre-fabricated plates; quality control	*h*
Assemble-ability of plates	Assemble-ability	Pre-fabricated plates; Dimensional changes	*h*
Pre-fabricated plates	Robotic Fabrication	G-Code; machine setup	*h*

In the domain of **On-site Assembly and Construction** (7.4.9), four theoretical design principles of high transferability have been identified (Tab. 8.15). The corresponding flow diagram, which is inverted as part of the definition of design objectives, is figure 7.67.

Table 8.15: On-site Assembly and Construction.

Objectives	**Principle**	**Prerequisites**	*T*
Finished building	Fit-out and Finishing	Assembled shell; glass façade	*h*
Assembled shell	On-site Assembly	Assemble-ability; support structures; on-site delivery	*h*
Support structures	Support Structures	Building location	*h*
Building location on site	Stake Out	Concrete slab, building data	*h*

In the above evaluation of 48 theoretical design principles, a total of 45 principles of high transferability have thus been identified. With their design objective stated, their mechanisms described in the methods section of functional principles, and the necessary pre-requisites (inputs) defined, those principles can now be considered validated *design principles*.

8.6 Summary of Design Principles for Segmented Shells

This section now provides a summary and comparison of design principles that were identified in the two case studies: first the respective design domains are described, then the principles within the corresponding domains are compared and generalized. The design principles for segmented shells of a given domain are summarized in domain-specific tables: principles that span both case studies can be considered general design principles, other design principles are case-specific. Similar to the functional principles, these principles are shown to be in a causal relationship with each other: in the case of design principles, one principle's output (pre-requisite) is derived from the principle's mechanism and defines the following principle's input (objective).

8.6.1 Design Domains for Segmented Shells

Design principles that were defined in the two previous sections fall into the main categories of joint design, segment design, global design, structural design, fabrication development, and production.

In the case study of the Landesgartenschau Exhibition Hall, the structural design category distinguishes between structural joint design and structural segmentation, which includes global structural analysis. Also within this case study, the additional category of quality control has been established, which wasn't present in the ICD/ITKE Research Pavilion 2011 where quality control of the fabrication process has not been conducted. Similar to Case B, the 3D-laser scanning in Case A was primarily aimed at monitoring the behavior of the entire structure over time and was not part of the analysis of functional principles. Finally, the production category distinguishes between off-site fabrication and on-site construction (Tab. 8.16).

Table 8.16: Design domains for segmented shells.

<table>
<tr><th>Case Study I (Double-layer)</th><th>Case Study II (Single-layer)</th></tr>
<tr><td colspan="2">Joint design (A.1, B.1)</td></tr>
<tr><td colspan="2">Segment design (A.2, B.3)</td></tr>
<tr><td rowspan="2">Structural design (A.4)</td><td>Structural joint design (B.2)</td></tr>
<tr><td>Structural segmentation (B.4)</td></tr>
<tr><td colspan="2">Global design / Form finding (A.3, B.5)</td></tr>
<tr><td colspan="2">Fabrication development (A.5, B.6)</td></tr>
<tr><td>-</td><td>Quality control (B.7)</td></tr>
<tr><td rowspan="2">Production (A.6)</td><td>Off-site Fabrication (B.8)</td></tr>
<tr><td>On-site Construction (B.9)</td></tr>
</table>

8.6.2 Joint design principles

As a generalization, the common objective within joint design for single and double-layered segmented plate shells is to achieve an adaptive model of an edge-wise connection for discrete segments that leads to a globally bending-bearing structure via the design principles of Bending-Free Edges (A.1.5, B.1.5).

The mechanisms in the corresponding design principles lead to the ensuing objectives of a trivalent segment layout and of using a connection that generates shear-load bearing capacity, e.g., through interlocking along the perimeter of edge-wise connected plates, as in Shear-Load Transfer (A.1.4, B.1.2).

In both cases, this leads to the objective of developing an integral timber joint type, such as finger joints, which can be achieved through teethed joint patterns in plate edges as described in the principles of Joint Production (A.1.2, B.1.1). The corresponding prerequisite is a fabrication strategy for this joint type as described in the principle of Finger Jointing (A.1.1, B.1.1)

In both cases, this leads to the objective of developing an adaptive joint model that can incorporate varying angles between adjacent plates, varying thicknesses between plates, and varying number of teeth along each edge via the design principle of Joint Adaptation (A.1.3, B.1.5).

In the case of single layered shells (Case B), the joint model also includes the requirement for Joint Gaps (B.2.5) to ensure assemble-ability and for variable edge orientation in anticlastic surface regions via the principle of Geometric Compatibility (B.1.4). Joints gaps, together with the effect of Co-linearity (B.4.2), lead to the objective of increasing stiffness in the connection by using additional fasteners via the corresponding design principle of Higher Stiffness. In Case B, the increase in stiffness is based on the combined roles of fasteners, finger joints, and gaps between joints whose characteristic stiffness values can be derived through a combination of physical load testing and analytical models as described in the corresponding design principles. In both cases, physical Load Testing (A.4.2, B.2.4) allows for structural validation and the calibration of the FE-model with respect to the design capacity of the joints.

The principle objective in Case B of using an edge-wise connection that includes additional fasteners can be realized via the design principle of Cross-Laminated Veneer (B.2.1), which in Case B leads to the use of Beech plywood: code requirements dictate that screws have to penetrate multiple layers of veneer with grain running perpendicular to the screw direction. Additional requirements leading to the use of Beech plywood are the need to accommodate loads that can come from all directions around the plate perimeter; and the higher stiffness of the material compared to LVL from softwoods.

Finally, the objective in Case B to minimize plate thickness is limited only by the use of additional fasteners as described in the principle of Material Overlap (B.1.3). Table 8.17 shows a summary of the mentioned design principles and figure 8.1 an example of how selected design principles can be arranged to form a chain of design principles.

Table 8.17: Joint design principles.

Design objective	**Case Study I (Double-layer)**	**Case Study II (Single-layer)**
Bending-free edges	Bending-free edges (A.1.5, [N.N.])	
Shear-load transfer	Shear-load transfer (A.1.4, B.1.2)	
Integral joint	Joint production (A.1.2, B.1.1)	
Fabrication strategy	Finger jointing (A.1.1, B.1.1)	
Adaptive joint model	Joint adaptation (A.1.3, B.1.5)	
Variable edge orientation	-	Geometric Compatibility (B.1.4)
Assemble-ability	-	Joint gaps (B.2.5)
Increased in-plane stiffness	-	Higher stiffness (B.2.6)
Structural validation	Load testing (A.4.2, B.2.4)	
Edge-wise connection incl. fasteners	-	Cross-laminated veneer (B.2.1)
Minimum plate thickness	-	Material overlap (B.1.3)

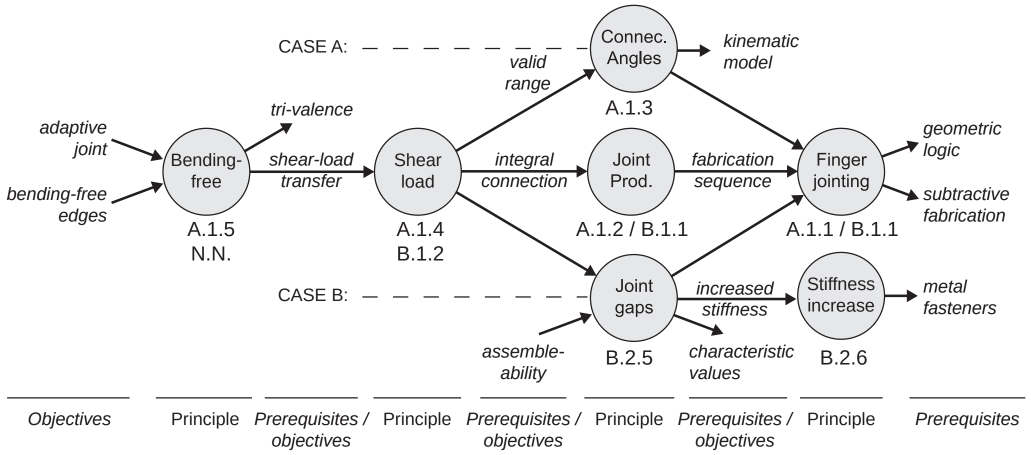


Figure 8.1: Chain of joint design principles

8.6.3 Segment design principles

Similarly generalizing from the case-specific segment design principles, the common objective in segment design is the design of discrete segments, from which the segmented shell can be constructed, that are geometrically adaptive to varying requirements in the shell. In both cases, this requires a segmentation strategy that leads to valid segments.

In Case A, the high-level design objective of achieving such a segmentation is met on the local level of the segments through the design principle of Frustum (A.2.3), which meets the requirement of planarity and of a trivalent plate configuration. A corresponding non-planar polygonal segment outline is achieved through dynamic equilibrium methods as part of the principle of Parametric Proxy (A.2.2). On the global level of the segmentation, the objective of a trivalent segment layout is met through the principle of Cellular Morphology (A.2.1). The segmentation objectives on global shell and local segment levels thus result in a case-specific segmentation on two levels as described in the high-level design principle of Hierarchy (A.2.5).

An alternative design approach towards segment design can be generalized from Case B for single-layer shells. In this approach, segment design is also principally based on the objective of achieving planarity of segments in a trivalent configuration (B.3.9) (due to the planarity requirement of the stock material and due to the geometric locking objective mentioned above). However, the validity of the segmentation is additionally evaluated based on the requirements of plate size, plate diameter, and plate angles. This evaluation occurs on the results of Planar Approximation (B.3.2) of a double curved surface. The mechanism used for planar approximation (TPI) leads to the lowest level objective in segmentation design of defining a topological relation between input points on a design surface through Triangulation (B.3.1).

Table 8.18 shows a summary of the mentioned design principles and figure 8.2 an example of how selected principles can be arranged to form a chain of design principles.

Table 8.18: Segment design principles.

Design objective	**Case Study I (Double-layer)**	**Case Study II (Single-layer)**
Discrete segment	Segment model (A.2.4, B.3.9)	
Planarity	Frustum (A.2.3)	Planarisation (B.3.9)
Valid segmentation	-	Validity (B.3.8)
Polygonal outline	Parametric Proxy (A.2.2)	Planar approximation (B.3.2)
Trivalent segment layout	Cellular morphology (A.2.1)	Triangulation (B.3.1)
Two-level segmentation	Hierarchy (A.2.5)	-

8.6.4 Global design principles

Generalizing design principles for global design of segmented shells, both cases pursue the high-level objective of a segmented shell model that meets architectural, programmatic, logistic, and structural requirements, which at the same time allow for design exploration. This can be achieved through a design principle evaluating the Validity of the Segmentation (A.3.5-7, B.3.8) while being embedded in a feedback loop with the design principle of Steering of Form (A.3.8, B.5.5).

In Case A, logistic requirements lead to objectives for meeting available resources

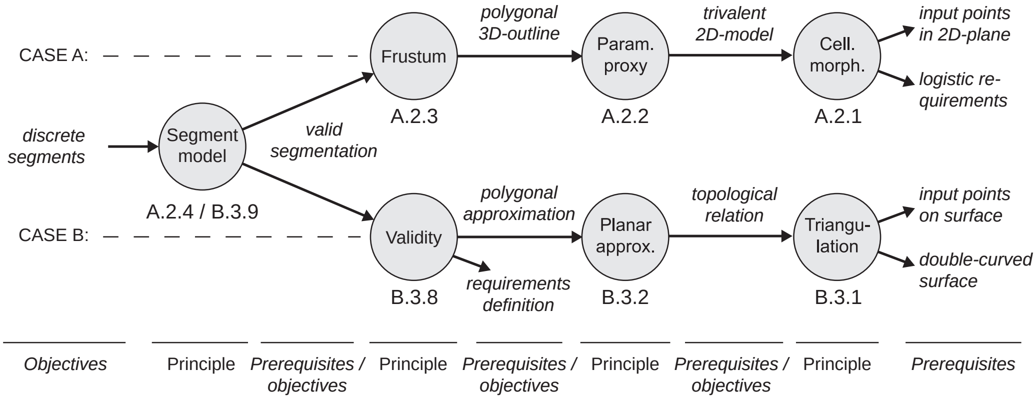


Figure 8.2: Chain of segment design principles.

addressed through the principle of Plate Coverage (A.3.7), and for morphological adaptation based on structural and architectural requirements through the principles of Anisotropy (A.3.5) and Cell Sizes (A.3.6). In Case B, the corresponding objectives are addressed through the principles of Plate Size (B.3.4), Plate Diameter (B.3.5), and Force Alignment (B.4.5).

The design objectives for a fully segmented shell are addressed in Case A through the principles of Global Cell Arrangement (A.3.4), where the segment design principles described above are applied; and in Case B by applying the corresponding segment design principles defined in Case B (B.3.1-2).

Both cases require a form-found reference geometry for the instantiation of shell segments, which can be achieved through the application of Dynamic Equilibrium methods (A.3.3, B.5.3): in Case A closed polylines act as a reference for the plate cells thereby linking segmentation design and global design; in Case B, the result of form-finding is a design surface, which needs to meet the distinctive objective of TPI compliance through the principle of Double Curvature (B.5.4).

The high-level design principle of Steering of Form (A.3.8, B.5.5) is based on similar lower level mechanisms that are used in both cases: Dynamic Equilibrium (A.3.3, B.5.3), Edge Control (A.3.2), and Local Controllers (in Case A a controller of design forces that feed into A.3.3; B.5.2). However, one notable difference between the two cases is that, while in Case A Topological control also influences the form-finding of the shell, in Case B form-finding is independent of the topological relationship between segments in the shell. In the further development it is therefore proposed to use form-found triangular meshes as input for the TPI method in the design of single-layer segmented shells, thereby directly linking form-finding and segmentation for single-layer shells.

Table 8.19 shows a summary of the mentioned design principles and figure 8.3 a high-level example of a chain of selected design principles.

Table 8.19: Global design principles.

Design objective	Case Study I (Double-layer)	Case Study II (Single-layer)
Validated segmentation	Validation (A.3.5-7, B.3.8)	
Available resources	Plate coverage (A.3.7)	Plate size (B.3.4)
Morphological adaptation	Cell sizes (A.3.6, B.3.5)	
	Anisotropy (A.3.5)	Force alignment (B.4.5)
Full segmented model	Cell arrangement (A.3.4)	Segmentation design (B.3.1-2)
TPI compliance	-	Double curvature (B.5.4)
Design exploration	Steering of form (A.3.8, B.5.5)	
Form-found surface	Dynamic equilibrium (A.3.3, B.5.3)	
Design forces	Local controllers ([N.N.], B.5.2)	
Adjustable boundaries	Edge control (A.3.2, B.5.1)	
Pre-programmed mesh	Topological control (A.3.1)	-

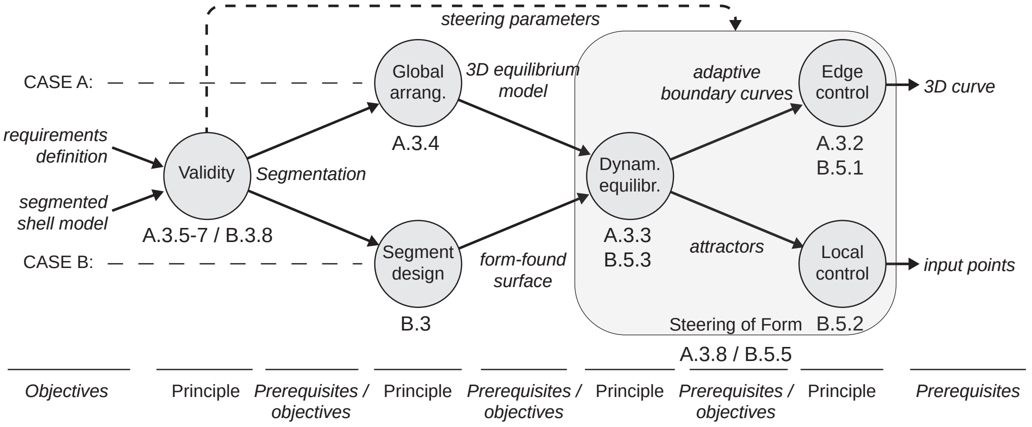


Figure 8.3: Chain of global design principles.

8.6.5 Structural design principles

Similarly generalizing from the case-specific structural design principles, the common objective in structural design / structural segmentation design is to achieve a globally bending bearing free-form segmented surface structure independent from load-optimized shapes based, e.g., on self-weight through a trivalent configuration of segments that leads to geometric locking between segments and therefore to global bending bearing capacity as described in the principle of Free-Form Shells (A.4.6, B.4.3-5).

Both cases pursue the structural design objective of more uniform stress distributions in the shell through homogenization strategies: in Case A Stress Reduction (A.4.5) is achieved by exclusively using convex plates, whereas in Case B the plate orientation is adapted to the principal stress directions through Force Alignment (B.4.5). This objective, together with the objective of improving the structural performance of a plate shell consisting of bending-weak joints through Stiffness Increase (A.4.4) in Case A or Added Bending Stiffness (B.4.4) in Case B lead to the objective of developing a detailed FE-model that can be used to iteratively optimize the segmentation through Automated FE-modeling (A.4.3, B.4.3). Both cases pursue such a generative approach, which allows for rapid iterations of simulation and analysis and is a pre-requisite for optimization.

The FE-model then incorporates the detailed joint model developed in structural joint design and its joint characteristics, which have undergone Structural Validation (A.4.2, B.2.4) through physical load-testing. Based on the given design brief, relevant load cases can then be applied to the FE-model for simulation and analysis, including those that are based on code requirements for permanent buildings.

At the basis of the structural design is the objective to develop an analytical structural model of bending stiffness in trivalent plate arrangements, a concept that has been derived from biological role models as part of the research leading up to the ICD/ITKE Research Pavilion 2011.

Table 8.20: Structural design principles.

Design objective	**Case Study I (Double-layer)**	**Case Study II (Single-layer)**
Globally bending bearing	Free-form shells (A.4.6, [N.N.])	
Uniform stress distribution	Stress reduction (A.4.5)	Force alignment (B.4.5)
Improved performance	Increased stiffness (A.4.4)	Added bending stiffness (B.4.4)
Detailed FE-model	Automated FE-Modelling (A.4.3, B.4.3)	
Structural Validation	Load-testing (A.4.2, B.2.4)	
Structural model	Trivalence (A.4.1, B.4.1)	

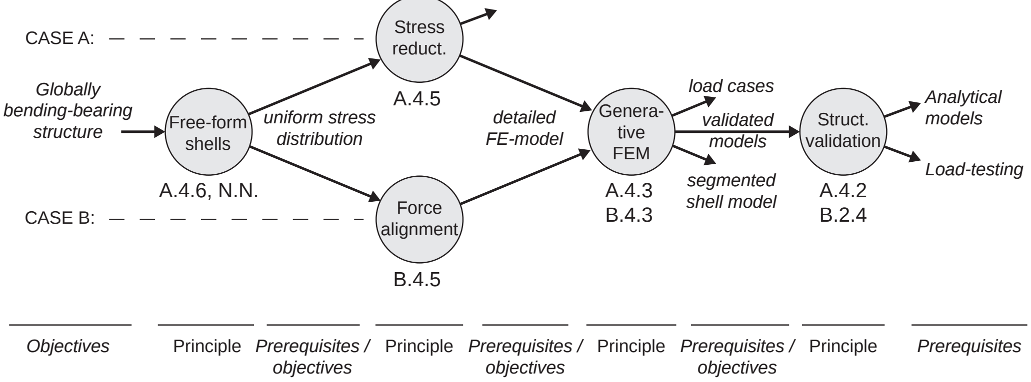


Figure 8.4: Chain of structural design principles.

Table 8.20 shows a summary of the mentioned design principles and figure 8.4 a high-level example of how selected principles can be arranged to form a chain of design principles.

8.6.6 Fabrication design principles

Generalizing design principles for fabrication design, the common highest level objectives in both cases are to generate Machine Code (A.5.8, B.6.4) that drives the machines used in fabrication as well as to describe the space of producible form as a precondition for fabrication-informed design through the high-level design principle of Machinic Morphospace (A.5.8, B.6.5).

In both cases, the subsequent design objective of fabrication validation is met via the corresponding principles of Process Validity (A.5.7, B.6.3), which are based on a kinematic model of the 7-Axis Robotic Fabrication machine setup (A.5.5, B.6.2), and Generative CAM (A.5.3, B.6.1). Both, fabrication modeling and machine code generation are achieved through a generative approach. While in Case A machine code generation was split into two consecutive steps of NC-Code Generation (A.5.4) and post-processing as part of Fabrication Validation (A.5.7), in Case B both steps were combined in a “production-immanent” design environment [27, p. 360] as part of Process Validation (B.6.3). Both cases pursue a parametric approach towards finger joint fabrication modeling, which, at the lowest level, is based on a case-specific fabrication sequence for Finger Jointing (A.1.1, B.1.1).

Table 8.21 shows a summary of the mentioned design principles and figure 8.5 a high-level example of how selected principles can be arranged to form a chain of design principles.

Table 8.21: Fabrication design principles.

Design objective	**Case Study I (Double-layer)**	**Case Study II (Single-layer)**
Producible form	Machinic morphospace (A.5.8, B.6.5)	
G-Code generation	Machine Code (B.6.4)	
Fabrication validation	Fabrication validation (A.5.7) / Process validity (B.6.3)	
Kinematic model	7-axis robotic fabrication (A.5.5, B.6.2)	
ISO code	NC-Code generation (A.5.4)	-
Machine tool paths	Automated CAM (A.5.3, B.6.1)	
Fabrication sequence	Finger jointing (A.1.1, B.1.1)	

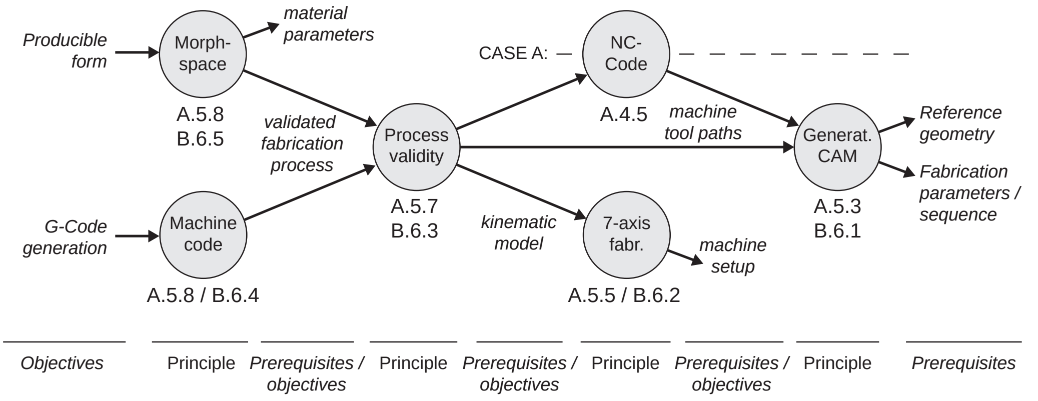


Figure 8.5: Chain of fabrication design principles.

8.6.7 Quality control principles

The abstraction and definition of general design principles in the domain of quality control is based on the single case study of the Landesgartenschau Exhibition Hall. It is therefore, from a methodological point of view, lacking generality. On the other hand, quality control is an established concept, such that the validity of the proposed design principles can be confirmed through expert knowledge.

The overarching objective of quality control is to vet the fabrication quality. Prerequisites for Quality Control (B.7.3) thus are the availability of samples that have been fabricated using the process to be vetted and tolerance specifications, which in Case B have been derived from structural and assembly requirements via the principle of Fabrication Tolerance (B.7.2). The inherent Fabrication Accuracy (B.7.1), on the other hand, is a function of the machine setup, the material used, and chosen fabrication parameters. Therefore, meeting given accuracy requirements can be challenging if, for example, the requirements cannot be met by adjusting fabrication parameters alone: changing material would have structural and architectural implications; changing the fabrication equipment can have repercussions on what can be produced and, conversely, what can be designed.

Importantly, after fabrication accuracy has been determined and confirmed to be within the tolerance specifications, the accuracy value (standard deviation) can be fed back into the tool path generation as part of Fabrication Design. As discussed in Case Study II, this has not been the case in the Landesgartenschau Exhibition Hall.

A further complication of quality control of timber products are possible dimensional changes due to changing environmental conditions, which can happen after quality control has taken place. While the corresponding functional principle (B.7.4) cannot meaningfully be turned into a design principle, the obvious objective is to minimize those changes via appropriate warehousing and transportation measures.

The functional principle can therefore be restated as the design principle of Quality Assurance (B.7.4).

Table 8.22 shows a summary of the mentioned design principles and figure 8.6 a high-level example of how those principles can be arranged to form a chain of design principles.

Table 8.22: Quality control principles.

Design objective	**Case Study I (Double-layer)**	**Case Study II (Single-layer)**
Minimise shape changes	-	Quality assurance (B.7.4)
Vetted quality	-	Quality control (B.7.3)
Tolerance specification	-	Fabrication tolerance (B.7.2)
Inherent accuracy	-	Fabrication Accuracy (B.7.1)

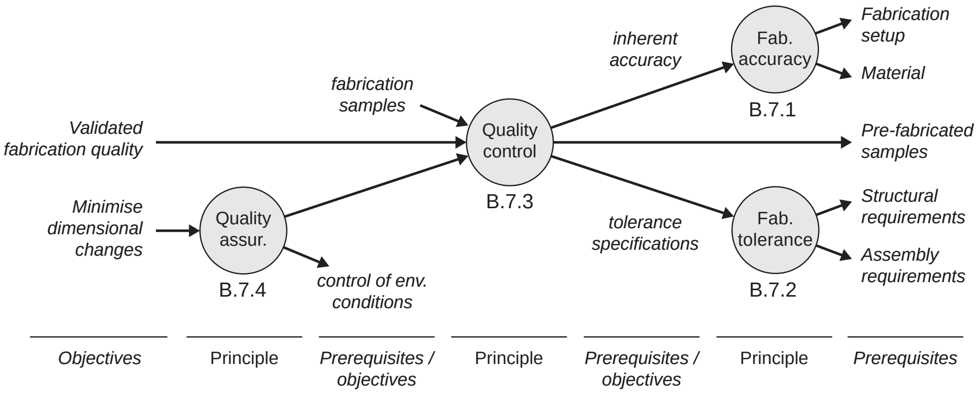


Figure 8.6: Chain of quality control principles.

8.6.8 Production design principles

Finally, generalizing design principles for production from the two case studies, the common high-level objective in both cases is the finished building. In Case B, this was achieved through the principle of Fit-Out and Finishing (B.9.4) as a step following On-site assembly (B.9.3), whereas in Case A the pavilion was completed by On-Site Assembly (A.6.4) (in this case, fit-out was limited to the installation of the interior lighting).

For the assembly of segmented shells, the objective and corresponding design principle to use minimal Support Structures (A.6.4, B.9.2) constitutes a de facto mandatory design principle, as this is one of the primary advantages of the segmentation of shells. In both cases, on-site assembly predicates the objectives to define the building location on site through Stake Out (not explicitly mentioned in Case Study I; B.9.1) and to install pre-assembled segments through Pre-Assembly

(A.6.3, B.8.3): in Case A of polygonal plate cells and in Case B of sub-assemblies, such as the insulation package. In Case B, due to the distance between the site of pre-fabrication and construction site, off-site fabrication also leads to the objective of on-site delivery via Transport and Delivery (B.8.4).

Both cases pursue the objective and implement the corresponding design principle of Assemble-ablity, albeit case-specifically: in Case B, this leads to Joint Gaps (B.2.5) and the need for variable edge orientations as part of Geometric Compatibility (B.1.4); whereas in Case A, assemble-ability was ensured on the plate-level through the elasticity of the material and on the segment-level through the convexity of the plate cells as the outcome of Cellular Morphology (A.2.1). Pre-assembly finally leads to the lowest level objective of pre-fabricated elements, which in both cases is achieved through case-specific robotic pre-fabrication of plates (A.6.1, B.8.1).

Table 8.23 shows a summary of the mentioned design principles including the additional requirements addressed in Case B, which mostly relate to the intended use and permanence of the building as additional design objectives and corresponding design principles: regarding the finishing of the building (including glass facade, lighting, and internal fit-out), support structures, and regarding transportation and just-in-time delivery on site. Figure 8.7 shows a high-level example of how selected principles can be arranged to form a chain of design principles.

Table 8.23: Production principles.

Design objective	**Case Study I (Double-layer)**	**Case Study II (Single-layer)**
Finished building	-	Fit-out and finishing (B.9.4)
Assembled shell	On-site assembly (A.6.4, B.9.3)	
Minimal formwork	On-site assembly (A.6.4)	Support structures (B.9.2)
Building location on site	Stake out (B.9.1)	
On-site delivery	-	Transport and Delivery (B.8.4)
Pre-assembled segments	Pre-assembly (A.6.3, B.8.3)	
Assemble-ability	Assemble-ability (A.6.2, B.8.2)	
Pre-fabricated plate	Robotic pre-fabrication (A.6.1, B.8.1)	

8.7 Design Patterns

As stated before, identifying patterns of individual design principles and groups of design principles that occur in both cases aims at generalization as part of an inductive research approach. Combined with the intended reuse of those patterns, this defines what is commonly referred to as ‘design patterns.’

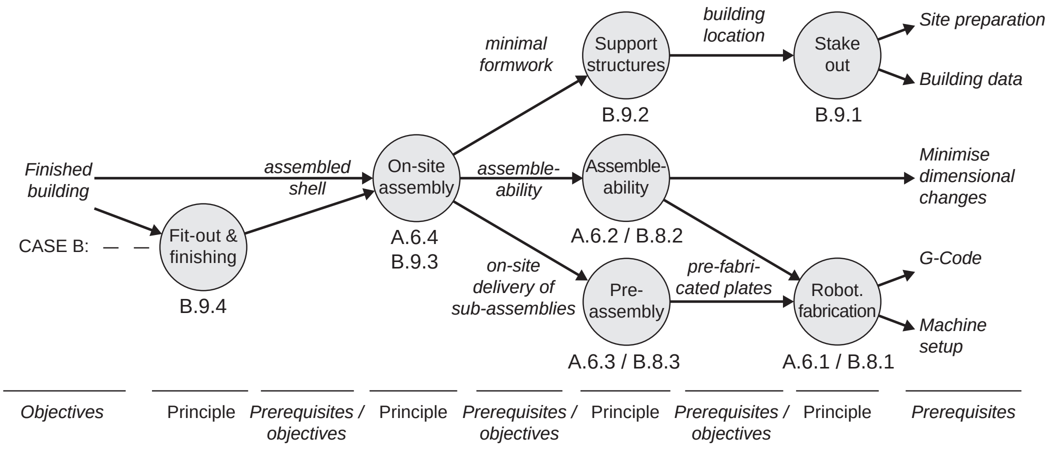


Figure 8.7: Chain of production principles.

8.7.1 Historical View

The term was first used by Christopher Alexander (*1936) in his 1964 publication “Notes on the synthesis of form,” but it is most commonly known from his seminal work “A Pattern Language: Towns, Buildings, Construction” from 1977. There, Alexander et al. [5, p. x] define it as:

> a “pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.”

Patterns are normative, stated as instructions, and describe what should be done ([5, p. x]; [212, p. 186]). Seen as a rulebook, however, “A Pattern Language” can hardly be considered new, as the definition of systematic rules for architectural design is a common thread throughout architectural history. The oldest known example in the Western hemisphere is Marcus Vitruvius Pollio’s “De architectura libri decem” (Ten books on architecture) from Roman times (ca. 20 BCE), which was rediscovered during the Renaissance and became a model for subsequent treatises, such as Leon Batista Alberti’s “De re aedificatoria” (1485), Palladio’s “Four Books of Architecture” (1742), and other authors such as Laugier (1753) and Ruskin (1844) ([212, p. 186]; [72, p. 155]). Still, “A Pattern Language” must be considered one of 20th century’s seminal treatises on architectural design and it is still resonating in the 21st, especially in the current context of computation in architecture. It provides the conceptual framework for the reusability of parts of a model and the connections between them and is thus directly transferable to parametric design, as demonstrated by Woodbury, who states that any “complex [parametric] model is composed of (mostly reusable) parts” [212, p. 185].

"A Pattern Language" must be seen in the context of computational thinking of the time: while Alexander worked on his PhD "The synthesis of form; some notes on a theory" (1963) at Harvard, he attended Herbert A. Simons' (the later Nobel laureate and Turing award winner) Symposia on System Theory and collaborated with neighboring MIT, where at the same time Ivan Sutherland developed Sketchpad, the first parametric CAD system [104, p. 28]. Furthermore, the definition of a language of patterns must be seen in the context of Noam Chomsky's 'generative grammars' in linguistics ([212, p. 186]; [104, p. 31]). Finally, "A Pattern Language" must also be seen in relation to Alexander's previous work "Notes on the synthesis of form." While both works are preoccupied with the decomposition of design problems—an idea based on Simon's work [104, p. 28]—in the earlier "Notes" the components were organized hierarchically, in a tree-like manner, whereas in the later "Language" they are organized in a loose, decentralized network. The realization that "A City is Not a Tree" (1965) expresses this change in the conceptualization of design problems, which appears to be influenced by emerging insights into the networked character of complex systems.

While it is fair to say that the teaching and application of Alexander's design patterns seem to have had limited relevance in architectural design education and practice—judged by its own ambitions it must be considered a failure [104, p. 27]—his book has had significant impact on other disciplines, most notably on computer science. In 1994, Gamma, Helm, Johnson and Vlissides published "Design Patterns: Elements of Reusable Object-Oriented Software", where they explicitly reference Alexander's definition of 'pattern' (see [60, p. 12]). In view of increasing complexity of software projects, "design patterns allow programmers to refer to networked problem solving patterns that have been proven in practical applications" [104, p. 27]. Patterns thus became a "tool to explain informal mid-level compositional ideas in computer programming" [212, p. 186]. Since the mid-1990s, thousands of articles and hundreds of books about software design patterns have been published referencing Alexander's work [104, p. 27]. Through object-oriented programming, the concept of design patterns has even influenced agent-based modeling (see, e.g., [86] and [69]).

On the one hand, Kühn [104, p. 27] considers Alexander's pattern language a failure, on the other hand, he recognizes the validity of the concept of design patterns, which has been proven in Computer Science (CS) and other fields. According to Kühn [104, p. 31], the problem of Alexander's pattern language was the universal scope of its ambition. In fact, this is how Gamma et al. [60, p. 393] distinguish their work from Alexander's: "Alexander claims his patterns will generate complete buildings. We do not claim that our patterns will generate complete programs."

A further critical difference between Alexander's original design patterns and the patterns defined in referencing work, including the design principles presented here, is that while some effort was made to give the "empirical background of the pattern, the evidence of its validity" [5, p. xi]) many of the statements seem to be tainted by a particular, some might say esoteric [104, p. 31], world view. In contrast, software engineers "grounded specific patterns in shared expertise with a group of authors and reviewers" [212, p. 186]. What made design patterns a success then, particularly in computer science, is that "Software engineers dropped all of the philosophy, leaving only the device itself" [212, p. 186].

8.7.2 Design patterns for computational design

These aspects of scope and validation also are what would differentiate design patterns for segmented shell design from the original Pattern Language: First, design patterns in segmented shell design do not aim at generating complete shells, let alone complete buildings. Second, as demonstrated in this case, functional principles are based on the analysis of built segmented shells by looking for repeating 'patterns' (in the original sense of the word) in the development process of those shells. These shells have been developed collaboratively (group of authors) according to building regulations and development processes have been peer-reviewed (reviewers).

Given that the concept of design patterns has been validated in CS and the segmented shells, from which the proposed design principles have been derived, have been validated through prototype buildings and independent review, the definition of design patterns for segmented timber shells appears to be a meaningful proposition. Furthermore, the affinity between software engineering and architecture (with regard to methods but also language, see, e.g., the term "software architect" and "software architecture"), not only in the heavily computer-based areas of agent-based segmented shell design, but also in more mainstream areas, such as building information modeling, suggests that some of the development approaches in CS might be applicable in architectural design.

Interestingly, the inverse point of view originally led to the adoption of design patterns in Computer Science: "Even though Alexander was talking about patterns in buildings and towns, what he says is true about object-oriented design patterns" [60, p. 12]. Tidwell [191, p. ix] proposing patterns for User Interface Design states that Alexander's work is "considered the gold standard for a pattern language because of its completeness, its rich interconnectedness, and its grounding in the human response to our built world." According to Gamma et al. [60, p. 24], "the hard part about object-oriented design is decomposing a system into objects." The relevance of Alexander's pattern-based approach thus is also due to the systematic approach to the decomposition of design problems.

The view that CS approaches are applicable in architectural design is supported by contemporary research on architectural design methods, where Christopher's and Gamma et al's work continues to resonate. An example being Robert Woodbury's 2010 book on "Elements of Parametric Design" [212], in which he dedicates one chapter to "Patterns for parametric design."

One further reason to define design patterns for segmented shells would be to "foster communication" about segmented shell design: "Rather than having to explain an idea from scratch, a group of designers can just mention a pattern by name. Everyone will, know at least roughly, what is meant" [212, p. 186]. According to Woodbury [212, p. 185], "patterns have become a common device in explaining systems and design situations." This view is seconded by Kühn [104, p. 27], who states that patterns can facilitate the discourse between participants when finding solutions in programming.

8.7.3 Distilling design patterns

In order to define design patterns from design principles, i.e., the intermediate layer "above nodes but below design" [212, p. 187], one would first look for design principles that span both case studies. Working from tables 8.17–8.23, this applies to the following design principles:

- in the area of **joint design**, to Bending-free edges (A.1.5, [N.N.]), Shear-load transfer (A.1.4, B.1.2), Joint production (A.1.2, B.1.1), Finger-jointing (A.1.1, B.1.1), Joint adaptation (A.1.3, B.1.5), and Load testing (A.4.2, B.2.4);
- in the area of **segment design** to Segment model (A.2.4, B.3.9);
- in the area of **global design** to Validation (A.3.5-7, B.3.8), Cell sizes (A.3.6, B.3.5), Steering of form (A.3.8, B.5.5), Dynamic equilibrium (A.3.3, B.5.3), Local controllers ([N.N], B.5.2), and Edge control (A.3.2, B.5.1);
- in the area of **structural design** to Free-form shells (A.4.6, [N.N.]), Automated FE-modeling (A.4.3, B.4.3), Load testing (A.4.2, B.2.4), and Trivalence (A.4.1, B.4.1);
- in the area of **fabrication design** to Machinic morphospace (A.5.8, B.6.5), Machine code (B.6.4), 7-axis robotic setup (A.5.5, B.6.2), Automated CAM (A.5.3, B.6.1), and Finger jointing (A.1.1, B.1.1);
- in the area of **quality control** the 'spanning' criterion doesn't apply, but Quality assurance (B.7.4), Quality control (B.7.3), Fabrication tolerance (B.7.2), and Fabrication accuracy (B.7.1), might still be considered general enough;
- in the area of **production** to On-site assembly (A.6.4, B.9.3), Stake out (B.9.1), Pre-assembly (A.6.3, B.8.3), Assemble-ability (A.6.2, B.8.2), and Robotic Fabrication (A.6.1, B.8.1).

From this list, the higher-level principles, meaning ones that combine multiple design principles, would be obvious candidates for design patterns as they group recurring design principles. This applies to:

- **Bending-free edges** being composed of Shear-load transfer and Trivalence;
- **Steering of Form** being composed of Edge control, Local controllers, and Dynamic Equilibrium;
- **Free-form shells** being composed of Automated FEM, Stiffness increase, and Stress reduction / Force Alignment;
- **Morphospace** being composed of Automated CAM, 7-axis robot setup, and Process validity;
- **Prefabrication** being composed of Robotic fabrication, Pre-assembly, and Fabrication accuracy.

In the next step, one way to define design patterns would be to follow Woodbury [212] in the definition of his 13 design patterns and adopt Tidwell's [191] style comprising *Title, What, Use When, Why, How*, and examples. As a proof-of-concept, this is shown for the group of principles that define "Bending-free edges." For the purpose of the demonstration, this involves a change in tone from descriptive to normative.

8.7.4 Example Pattern: Bending-free edges

What. Join plates along their edges in a bending-resisting configuration while using only hinged, i.e., bending-weak joints.

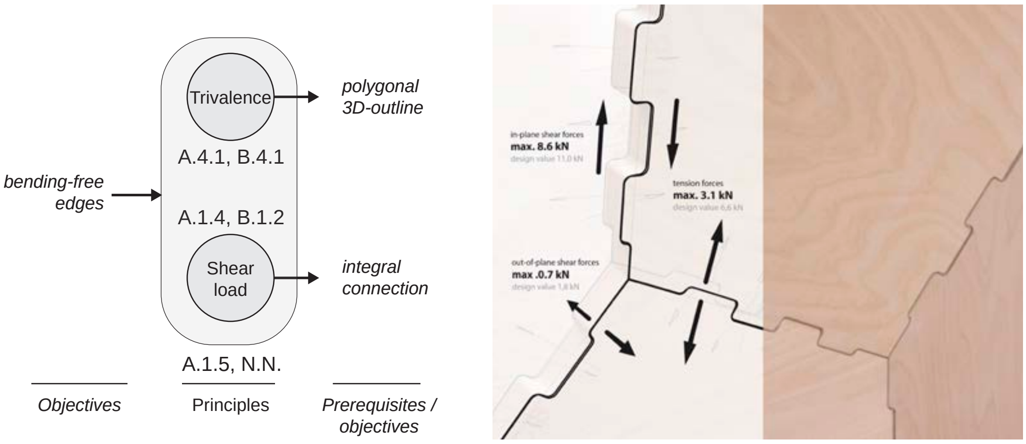


Figure 8.8: A pattern for the design of bending-free connections between segments in a segmented shell. Right image from [103] by Krieg (2014).

Use When. Segmenting shell surfaces interrupts the material continuity of the shell and thus disturbs the stiffness of the entire structure [114, p. 123]. Building shell surfaces from individual segments that are thin relative to their extents thus requires a connection that both transfers loads between segments and re-establishes shell action. Use this pattern when you have control over both the joint type and the arrangement of the plates, because both need to be considered together. Conversely, if you are, e.g., constrained to quadrilateral segments, then adding bending stiffness in the joint is mandatory, and this pattern is not applicable.

Why. Bending-weak joints do not add to the cross-section of the shell in the same way as bending-resistant joints typically do, but they need to be tightly coupled to the topology of the plates. Using a bending-weak joint like finger-joints can have various advantages, such as material reduction, reducing or even eliminating the use of metal fasteners, and ease of assembly, while providing high shear stiffness. Additionally, low-profile, integral joints allow for approximating the aesthetics of continuous surface shells (which, remember, was the original goal of segmentation).

How. The main requirement is that bending moments, arising, e.g., from imperfections in the shell, unsymmetrical external loads, such as wind or snow load, or non-load-optimized shell forms are transferred into shear stresses in the plane of the plate (membrane forces). Plate structures that are composed of only plates and hinges fulfill this requirement if always three plates meet in one point (the definition of a ‘pure plate’ structure).

The resultant stability is due to the geometric locking of the plates, where each plate is constrained by the other two. The plates are stabilized by resisting internal forces, which lie in the plate itself. Each face of a ‘pure’ plate structure carries plate forces only, such that plates do not need to rely on carrying bending moments and torsion for their own stability [109, p. 31]. Thus bending stiffness is generated even though the joints themselves are only hinged connections [114, p. 3]. However, bending stiffness in the shell also depends on the angle between consecutive edges: if one of the three angles of a vertex is close to 180 degrees, a trivalent plate structure will become locally kinematically unstable [115, p. 130]. In this case adding a small amount of bending stiffness, e.g., in the form of crossing screws, can improve the structural performance [115, p. 130]. This overall bending-bearing effect thus depends on a shear-load bearing joint type that prevents sliding along the edges of the plates.

Finger joints are such a joint type that is both bending-weak (a hinge) and shear-load bearing. The joint’s multiple interlocking straight-shaped fingers are arranged in an alternating pattern, which connect two plates along their shared edge in a force- and form-fitting manner [174, p. 158]. The connection functions similar to step joints in traditional truss works: the thrust force is taken by the contact surface between the fingers and transferred to the shear plane [115, p. 126]. Furthermore, the aesthetic qualities of this joint type are typically only achieved in furniture making.

From these considerations follow the subsequent requirements for a trivalent arrangement of polylines and the need for a modeling method for shear-load bearing joints.

Examples. Both case studies show example applications of this pattern: In the case of the ICD/ITKE Research Pavilion 2011, the arrangement of plates and joint type on the cell-level and the arrangement of segments (polyhedral cells) within the structure (two-level hierarchy); in the case of the Landesgartenschau Exhibition Hall, the arrangement of the segments and the joint type.

In the first example, it is interesting to see that effectively two different joint types were used: on the level of the plates, a finger joint type was used, whereas on the level of the segments bolted connections were used. The bolted connections were located along the flange surfaces of the cells and bolts inserted perpendicularly to the plate surface, effectively connecting adjacent cells surface-to-surface. This allowed for straightforward assembly and disassembly (while providing shear resistance), whereas the finger joint connections were designed to be permanent and enabled the edge-to-edge joining of plates that were only 6.5 mm thin.

Similarly, in the second example, it is important to note that crossing screws were added to the edge-to-edge connection in order to provide a minimal amount of bending stiffness. This was necessary as the bending bearing effect diminishes when angles between consecutive edges approach 180 °, i.e., become co-linear. If the segmentation is generated using the TPI method, then this is the case in areas surrounding the so-called ‘parabolic line’ where Gaussian curvature $K = 0$. Additionally, the crossing screws prevent out-of-plane shear forces, such as wind suction, from moving the plates along the reverse assembly direction.

Related patterns. Shear load transfer, Three-plate rule, Joint gaps, Added bending stiffness.

8.8 Mapping from Design Principle to ABM Paradigm

As the final step towards instrumentalizing the identified functional principles for the design of segmented shells, recommendations can now be made regarding: (1) which design principles can be abstracted and transferred to the ABMS approach and (2) how design principles and their mechanisms map to the constituent elements of agent-based models, such as agents, interactions, topology, environment, and behaviors.

The figures and tables in section 8.6 above provided an overview on design principles for segmented shells: generic principles that could be identified in both case studies and specific design principles that facilitate design objectives specific to a case. As stated above, these design principles are shown to be in an inverted causal relationship with each other.

In the implementation step, the nodes (principle mechanisms) in the corresponding network representations can be implemented, generally speaking, by software that process information or, more specifically, by classes in an object-oriented programming paradigm. The relationships between nodes can correspondingly be conceptualized as information that is exchanged between objects through respective methods that are associated with an object. A general way of representing this exchange of information in object-oriented programming is through UML (Universal Modeling Language) diagrams.

By synchronously updating the states of each individual object / node in the network, all interactions can be taken into account, thereby capturing the non-linearity of the development process. In the network of interacting software nodes, the flow of information thus is not limited to top-down relationships, which is the main difference to the uni-directional flow of information in the hierarchical structure of conventional feed-forward parametric models.

The particular potential of ABMS as a behavioral micro-level modeling approach can now be seen in its capacity to integrate bottom-up relationships into the design process that can reach across domains and disciplines. This would ultimately allow for the guided search of a solution space using multi-disciplinary rules, which in the terminology of an ABM are called *behaviors*.

It is worth noting, that in Case Study II the analysis of functional principles was intentionally separated from their implementation in the design of the Exhibition Hall. This allowed for the identification of functional principles independent from implementation and therefore now for making informed recommendations as to which design principles are particularly suited to be implemented within the ABMS paradigm.

The implementation of design principles for segmented shells within the ABMS paradigm is predicated on the mapping of mechanisms to the corresponding elements of the ABM, such as:

- building elements as locomotive agents,
- domain-specific evaluation routines as software agents,
- model space and design surfaces as environment,
- individual low-level rules as individual agent behaviors,
- connection between buildings elements as interaction topology,
- joints as type of interaction between agents,
- feedback through iterative update of the agent states.

The following table (8.24) provides an overview of selected design principles and recommendations as to how they can be mapped to corresponding constructs in the ABMS paradigm.

Table 8.24: Mapping design principles to ABMS paradigm.

Design principle	ABM construct
Agents as building elements	
Polygonal plate cell	Locomotive agent
Point in the 2D plane (Topological control)	Agent position attribute
Size of segment	Agent radius attribute
Joints between segments	Agent method
Two-level hierarchy	Agents (cells) composed of sub-agents agents (plates)
Agents as domain-specific software	
Machinic morphospace	Domain agent checking producibility based on machine setup / kinematic simulation
Structural evaluation	Domain agent performing FEA; also computes principal force directions in the shell
Available resources	Domain agent checking required resources
Environment and Interaction Topology	
Design surface	Environment agent
3D Cartesian space	Distance-based interaction
Dynamic equilibrium	Method of the environment
2D reference mesh	Interaction topology as basis for TPI and joint method
Behaviours	
Connecting Angles	Agent method, which ensures that the angles between plates stay in the producible range
Cell Sizes	Agent method, which controls the size of agents as a function of the distance to the system boundary
Anisotropy	Agent method, which changes segment aspect ratio w.r.t. principle force directions computed by the structural domain agent (agent radius is ellipsoid rather than circular)
Force alignment	Agent method, which locally changes a interaction topology w.r.t. principle force directions computed by the structural domain agent

Continued on next page

Design principle	ABM construct
Plate coverage	High-level agent system method, which changes number of agents and sizes of agents w.r.t to required resources computed by resource domain agent
Fabrication validation	Agent method controlling size and position of the agent w.r.t its producibility computed by Morphospace domain agent

8.9 Discussion

The two research buildings that have been studied in the above two case studies are physical demonstrators of the integrated and concurrent development of architectural and structural design, material system, fabrication, and production; and of the feedback between these development areas. Arguably, these buildings would be impossible to realize in the conventional, sequential, architectural design and development approach.

A methodical approach for the identification of functional principles and their role in the network, and for defining design principles has been introduced and described as **analysis**, **abstraction**, and **translation**. This not only allowed for the definition of domain-specific design principles for segmented shells and outlining possible relationships between design principles and domains, but is also a precondition for reproducibility of the findings that were originally made. The relevance of the presented research for the design of segmented shells therefore lies in the fact that, for the first time, the causal relationships between development steps and domains have been made explicit.

Three methodological aspects that have been instrumental in the case studies deserve special attention in the discussion: the dependency graph, design patterns, and the implementation recommendations.

Dependency graph. The abstraction step uses the tool of the dependency graph to form a unique representation for each shell and allows for comparison between shells. While every researcher might produce slightly different dependency graphs based on the same source material due to her or his unique research experience and point of view, the dependency graph can form the basis for a discussion about type and quality of relationships between development steps of a segmented shell. In fact, this was identified as one of the advantages of defining ‘design patterns.’ Effectively, the entire graph or parts of it can provide templates, i.e. design patterns, for the development of new segmented shells.

Based on the analysis of the dependency graphs in the two case studies, a distinction between 'top-down' and 'bottom-up' relationships in the architectural design and development process has been proposed in this chapter. One significant finding from the analysis of the dependency graph in Case B was, e.g., that the design and realization of segmented shells might well be implemented within the established stages of project delivery, which would significantly lower the threshold of adoption.

The study of the network of dependencies and circular causalities also confirmed empirical knowledge that principles in the structural development and fabrication development domains typically are in a 'bottom-up' relation with design development. This means that they are both influencing and influenced by design development. Missing links in the flow of information and the lack of bottom-up feedback in the established design process can thus be identified as one of the reasons for the challenges in architectural production, such as late design changes or conflicts between different trades, that ultimately lead to the problems that the construction industry is facing in terms of efficiency and resource effectiveness.

Design patterns. Following the translation step from functional principles to design principles, domain-specific catalogues of design principles have been synthesized from the two case studies and respective causal chains of design principles have been proposed in this chapter.

This has lead to the question regarding the applicability of design patterns for designing segmented shells. The potential of design patterns and their relevance is considered the same as stated by Alexander et al. [5], Gamma et al. [60], Woodbury [212], and others with regard to facilitating a conversation about "mid-level compositional ideas" [212, p. 186] in shell design that is independent from implementation. Design patterns thus add another level of abstraction to the process from analysis to implementation—the forth level of abstraction so to speak—after translation, but before implementation.

It is taken as a given that there will be additional ways on the lowest functional level to develop specific aspects of segmented shells and other ways to categorize the functional principles into development areas, which would translate into additional design principles. However, this assumption does not affect the validity and the functioning of the individual principles that have been identified so far. These observations thus point to two preliminary conclusions:

1. More case studies are necessary to expand on the catalogue of functional principles; e.g., the BUGA Wood pavilion (Fig. 1.3) would be an obvious candidate.
2. A comprehensive catalogue of design patterns for segmented shell should be developed.

While it has been shown that developing design patterns from the design principles that have been defined so far is possible, to remain within the scope of this dissertation, establishing a comprehensive list of design patterns that includes additional case studies must be left for future work.

Finally, one could argue that the principles and patterns identified in the previous chapters may help in designing the demonstrators from which they have been derived, but this would be a circular argument. Therefore, in order to avoid "the fallacy of using the same data to both form and verify theory" [212, p. 188], the applicability of the principles and patterns needs to be verified in novel work. This is what will be demonstrated in chapter 9.

Mapping recommendations. Finally, recommendations could be made regarding the mapping from design principles to the constructs of an ABM. The importance of the definition of design principles for any software implementation and the mapping from design principle to software construct, particularly in the context of ABMS, is that the identified design principles represent validated conceptual models that are being implemented. As discussed before, the validation of conceptual models and their mapping to software are critical steps in building models of any natural or man-made system and typically happen through data collection or expert knowledge and review. The subsequent validation of implementation would be the final step in building a fully validated model.

PART III

Further Development and Conclusion

The most powerful and challenging use of the computer [...] is in learning how to make a simple organization (the computer) model what is intrinsic about a more complex, infinitely entailed organization (the natural or real system).

—Sanford Kwinter [106, p. 91]

9

Proof-of-Concept

9.1 Introduction

The aim of the research is to develop a methodology for creating agent-based models of segmented shells. As previously mentioned, the methodology pursues an inductive approach through a sequence of steps involving analysis, abstraction, translation, and implementation. Part II of this dissertation covered the first three steps by establishing design principles for segmented shells from built work through case studies. This chapter introduces the final step in the proposed methodology as a proof-of-concept.

In summary, the **analysis** step identified functional relationships between development steps and development areas of specific “cases”. **Abstraction** then occurred on three levels: On the first level, *functional principles* were identified. On the second level, *inference chains* were constructed that relate the outputs of a development area to its inputs through a series of functional principles where each principle’s output becomes the input for subsequent principles. On the third level of abstraction, relationships between development areas were analyzed using a graph-based representation termed the *dependency graph*, which allows for identifying feedback loops and concurrent developments.

Finally in the **translation** step, *design principles* were defined by inverting the causality of the functional principles. It was shown that taking the output of a principle as a design objective links the objective to the minimal required inputs via the mechanism of the principle. It was also shown that design principles (individuals or groups) that occurred in both case studies can be considered *Patterns* for the design of segmented shells. Furthermore, it was shown how different parts of the design process of segmented shells can be mapped to constructs of an agent-based system, in other words, how the problem of shell design can be conceptualized as an agent-based problem.

In order to "avoid the fallacy of using the same data to develop and confirm theory" [212, p. 188], **implementation** is considered the necessary final step towards completing and validating the proposed methodology. The objective of this chapter thus is to implement selected design principles as a proof-of-concept in order to demonstrate how domain-knowledge, which was abstracted in a systematic way from built work, can be applied to the design of new segmented shells using an agent-based modeling approach.

This is considered relevant for three reasons:

1. As stated in previous work [72], the main challenge remains the translation of design objectives into agent behaviors. Implementation is considered to be the final step in this translation.
2. Furthermore, "reusability is of major importance for the viability of this approach in industry" (ibid.). Because of the time that is allocated for design in practice, agent-based models cannot be developed from scratch as part of the formal design process due the the limited resources allocated to it.
3. By defining design principles in an inductive approach, they don't need to be proved again when they are used, they are, in other words, axiomatic and can be referred to "by name." Conceptual models, which conceptualize the part of the world under investigation (in our case the design problem), that are built from these design principles can therefore be validated more easily.

The vehicle for the proof-of-concept is a problem-oriented case study:[1] In previous approaches of modeling segmented shells, where agents typically represent building elements, global form-finding and agent-based modeling were separate development steps and integration between global form and agent system has not been realized. The objective therefore is to develop an agent-based model for the design of a segmented shell structure that explicitly takes advantage of the plate-lattice dualism. This dualism is expressed as

1. for every vertex of a plate in a plate structure exists a triangular face in a triangular lattice and vice-versa;
2. if a triangular lattice, e.g. form-found or free-form, is in static equilibrium then its plate-dual will equally be in static equilibrium (see section 4.4.2).

The expected advantage of such an approach is that it obviates the need for a design surface and allows for tighter integration of global form and local plate behaviors. It also holds the prospect of realtime structural analysis based on the analysis of the plate dual as proposed by Wester [206] and La Magna et al. [108].

---

[1] Following the recommendations by UNSW Sydney for writing case study reports in engineering (https://student.unsw.edu.au/writing-case-study-report-engineering, accessed October 21, 2018).

9.2 Related Work in ABMS Methodology

Taking the discussion of the related work regarding plate-lattice dualism (see 4.4.2) and the catalogue of design principles (see 8.6) as given, the remaining two principal aspects of the related work are ABMS best practices and previous work regarding the development of the ABM Framework at the ICD.

Macal and North [124, p. 157] provide a series of "questions to ask" when developing ABMs. These questions involve the purpose of the model ("what specific problem should be solved by the model?"), the agent representation ("what should the agents be in the model?"), the behaviors ("what agent behaviors are of interest?"), the environment, the type of interaction, the provenance of data ("especially on agent behaviors"), validation, implementation, and communication of the model. Particularly, "The communication of a model, its design assumptions, and detailed elements is essential if models are to be understood and reused by others than their original developers" [124, p. 158]. These questions guide the development of the ABM's constructs in the case study.

According to Macal and North [124, p. 157], "bottom-up, highly iterative design methodologies seem to be the most effective for practical model development." This view is seconded by Metz [132, p. 22], who frames the development of an agent-based model as a linear iterative process, not unlike that of architectural design, that will be repeated as necessary and in which an originally ambiguous, informal version is developed towards an unambiguous formal version, which ultimately is implemented in code. The individual steps of this development process can be summarized as

1. preliminary research (e.g., case study research, abstracting "patterns or *stylized facts*"),
2. informal or conceptual modeling (i.e., how the ABM constructs map to the objects of interest),
3. formal modeling (description of agents and behaviors, e.g., using "pseudo-code, Unified-Modeling-Language (UML)"),
4. implementation, and
5. verification and validation (of the conceptual model and implementation).

9.2.1 Preliminary Research and Informal Modeling

The above sequence of steps aligns well with the proposed methodology for developing agent-based models of segmented shells (consisting of analysis, abstraction, transfer, and implementation), where analysis and abstraction lead to *patterns* and *stylized facts*, and transfer concerns the mapping from design principle to ABM construct.

It is worth noting that, interestingly, the term *pattern* in this case is not used in the same sense as in the above *design pattern*. According to Railsback and Grimm [153, p. 228], a pattern signifies “anything beyond random variation. We can think of patterns as regularities, signals, or, as they are sometimes called in economics, ‘stylized facts’.” “Stylized facts are broad, but not necessarily universal generalizations of empirical observations and describe essential characteristics of a phenomenon that calls for an explanation” [153, p. 228].

The choice of a modeling paradigm in general, and of the constructs in the informal modeling step in particular, already constitute design decisions. These design decisions thus are inherently part of the conceptualization of the system under investigation (see [17, p. 413:1]). In particular, the microscopic perspective forces a specification of thinking from the part of the modeler that avoids ‘glancing over’ potentially relevant aspects of the system as might be the case in alternative top-down modeling approaches [132, p. 35].

9.2.2 Formal Modeling

In this development step, the specification of the agents, their behaviors, and their interaction is of particular importance:

- *agents* represent the entities of the system under investigation at the right level of abstraction;
- *behaviors* represent the rules by which agent act and react to other agents and the environment in the system, capturing the rules of the observed system;
- *interaction* is defined by a topology that determines which agents interact (see Section 3.2.4).

9.2.2.1 Agent model design

Jennings [82] structures the design steps of the agent model into decomposition, abstraction, and organization steps using flexible manufacturing as an example for decomposition. Similarly, Metz [132, pp. 19-20] states that agents should represent those characteristic features of the entities in the system under investigation that are considered relevant for the observed phenomenon. This view is seconded by Jennings [82, p. 286], who states that “the most powerful abstractions are those that minimize the semantic distance between the units of analysis that are used to conceptualize the problem and the constructs present in the solution paradigm.”

According to Metz [132, p. 24], a common design philosophy in ABM is “keep it simple, stupid!” In a similar vein, Coates and Schmid [40, p. 653] stress “simple rules - complex outcomes” and guard against “baroque rulesets”, which are self-defeating, since they already specify the majority of the problem solution, and take too long to program. Especially in the context of design, “one is easily trapped by preconceived ideas about the desired outcome. The simulation rules should [thus] be reduced as far as possible" [165, p. 276].

In other words, over-specifying the model will lead to over-constraining of its solution space especially in the Generator role of the model (see 3.2.9.2). While the abstracted agents and behavior should be close representations of the entities and rules in the system that is being modeled, they should capture the relevant aspects at the right level of detail. This implies that ABM can also be used to investigate and identify those aspects of the system, which are important for generating the observed effect. Determining the right level of description is therefore a critical aspect of ABM as noted by Bonabeau [23, p. 7287].

A further important modeling paradigm are hierarchical models as described by Minar et al. [134]. They allow representing individual agents as subgroups of agents at a lower level of hierarchy by nesting systems of agents within agents. An example would be modeling living cells as a collection of organelles (nucleus, mitochondria, endoplasmic reticulum).

Furthermore, Macal and North [124] identify means and methods for modelers to define the attributes and behaviors of agents. Termed 'Agent specification services', these services range from general purpose languages to domain-specific languages used in specific ABMS toolkits to visual languages, such as flowcharts.

9.2.2.2 Modeling behaviors

As discussed, behaviors describe the response of the agent to internal or external stimuli based on internal rules. The ability of the agent to change behaviors based on different contexts and based on previous experience is what distinguishes 'interactive' from 'adaptive' agent systems (see Section 3.2.4).

The design of behaviors can be based on normative rules, explicitly stating what the agent should do, or on heuristic rules, that is, rules that achieve the desired outcome based on previous experience without a guarantee of being optimal. Often, the challenge with designing micro-level behaviors is that the emergent macro-level system behavior is significantly different or 'novel' (see [47]), such that there is usually no obvious way of designing micro-level rules starting from a given desired macro-level outcome.

9.2.2.3 Agent Architectures

Different 'architectures' have been proposed to described the internal organization of an agent. This includes the reactive approach as in perception-action (or stimulus-response) architecture, the cognitive approach as in the Belief-Desire-Intent and a hybrid approach, which combines the first two [133, p. 14].

In reactive architectures, an "agent does not have an explicit representation of the environment nor of other agents. [...] The most well-known architectures in this domain are the 'subsumption architecture' in which tasks in competition are arbitrated along predefined priorities [...], the competitive task architecture, where concurrent tasks have their weight modified through a reinforcement learning process [...], and connectionist architectures which are based on neural nets [133, pp. 14 f.]. This approach represents the 'anti-classical-AI' idea that a group of agents may be able to perform tasks without explicit representations of the environment and of the other agents and that planning may be replaced by reactivity, which is at the core of swarm-based intelligence [135, p. 3].

In cognitive architectures, agents explicitly represent their environment using a symbolic formalism—the classical AI approach. The most well-known architecture of this type is Belief-Desire-Intent (BDI), which postulates that an agent is characterized by its beliefs, its desires and intentions. In this framework 'Belief' describes what the agent knows, 'Desire' what the agent wants and 'Intention' what the agent is doing [82, p. 288]. "A BDI agent is assumed to possess a library of plans, each plan being designed like a recipe making it possible to achieve a particular goal. An intention is set when an agent makes a commitment about achieving a specific goal by using a particular plan. The management and update of beliefs, goals and intentions are carried out by the BDI engine which selects the plans and the actions to be undertaken" [133, pp. 15].

In order to enable adaptation in agent behaviors, more advanced algorithms are used including methods from the field of Machine Learning. This includes Artificial Neural Networks in order to relate stimulus to effective agent action drawing from the agent's past experience as in reactive architectures; population-based approaches, such as Genetic Algorithms, to allow adaptation of agents at the population level such that agents with beneficial behaviors can represent a larger share of the population; and Genetic Programming in order to directly evolve successful behaviors.

9.2.3 Implementation

Regarding the actual implementation, Metz [132, pp. 35-36] proposes a series of design guidelines (i.e., modeling best practices), which also guide the implementation of the formal model in the case study below. The guidelines can be summarized as follows:

1. Keep the model simple in the beginning of the modeling process, then incrementally add complexity.
2. Start from a "null-model" without useful behavior, then build up the desired system effect step-by-step; alternatively, if there is an accepted model that can serve as the basis, start with the replication of the model before adjusting it to

specific requirements.

3. Changes to the model should occur incrementally and should always be compared to the system behavior of the previous version. In this way, changes in system behavior can be attributed to specific model changes or else determined that certain model changes have no system effect.
4. Program code and model should be well documented during development in order to avoid a black-box effect when used by others.
5. The model should be able to solve established benchmark problems in order to test if the behavior is correct.
6. If the model has a stochastic element, the same seed of the random number generator should be used in order to be able to distinguish correct system behavior from system error.

Agent-based models can be implemented 'from scratch' using general purpose programming languages or using specially designed ABM toolkits and development frameworks. Agent models can be implemented in the small, on the desktop, or in the large, using high performance computing and large-scale clusters [124, p. 158]. Macal and North also provide a comparison of the advantages and disadvantages of relying on existing toolkits vs. developing own tools (see also [134] on this topic).

The characteristic feature of ABMs is their object-orientation, where an agent can be conceptualized as an object possessing its own goals. Object-Oriented Programming (OOP) thus is an obvious choice for developing agent models. As Jennings [82] demonstrated, this concept can be extended to wrapping legacy software packages such that they appear as agents within the agent system. The ability to wrap legacy systems means agents may also be used as an integration technology (ibid.).

9.2.3.1 ABM Toolkits

One advantage of toolkits over general purpose languages is that they allow for re-use and sharing of sub-system components (such as agent models, behaviors and interaction models) between models and between applications. Another advantage is that toolkits usually come equipped with tools that support the modeling and simulation process, such as *Batch mode*, *Model analysis* for visualizations, data mining, statistics and report generation, or *Check-pointing*.

Specifically, *batch mode* is the automated processing without human intervention and can be used, e.g. for *parameter optimization* in order to find combinations of parameters that minimize a given cost function and for *sensitivity analysis* where model parameters are modified in order to evaluate the impact that certain parameters have on the system outcome. This involves running many instances of the model either with fixed parameters but randomized starting distributions in order to find the overall distributions the model generates and the relevance of the starting distribution, or with varying parameters in order to analyze how they drive the model's outputs and behaviors [131].

Check-pointing allows saving the state of the system at a given point in time in order to be able to go back to this specific state of the system [124, p. 160]. This is particularly relevant if getting to that point has taken a considerable amount of time or when alternative future trajectories of model development (or in this case, design alternatives) need to be studied.

The consolidation of ABM into an accepted modeling and simulation approach (in addition to DES and SD), which happened in most fields in the late 1990s and 2000s, was not only facilitated by but also predicated upon the development of ABM toolkits. Instead of scientists developing idiosyncratic, one-off software and duplicating efforts, the goal of agent development toolkits, also known as computational modeling software frameworks, is to allow comparison and repeatability of results by separating the development of the framework core from the modeling of the experiment. With this mind set, one of the first of a number of publicly released agent development toolkits was Swarm in 1996 ([121, p. 146]; see also [134]).

Since then, a significant number of software frameworks have been developed, mostly in the academic context. Nicolai and Madey [142] provide a broad overview and snapshot of the state of development by comparing 53 different ABM toolkits in 2009. Railsback et al. [154] on the other hand provide an in-depth look focusing only on four toolkits that were predominant in 2006 (NetLogo, Mason, Repast, and Swarm). While a considerable part of the toolkits in Nicolai and Madey's investigation seem to have ceased development as of 2020, with the exception of the original Swarm package, which continues as 'forks', the other three toolkits investigated by Railsback et al. are still major platforms today.[2] This hints at a challenge that many domain-specific toolkits, especially from the academic context, seem to be facing in that they need to build and maintain a community of users and developers of a critical size in order to be able to maintain and develop the code base.

Of course, new agent development toolkits have emerged since 2009, such as Eclipse's AMP, GAMA, Mesa, and the commercial software package AnyLogic. A review of contemporary toolkits, both open-source and commercial, such as NetLogo, Repast, Mason, Mesa, and Anylogic, showed that existing frameworks allow only for limited application in the field of Computer Aided Geometric Design (CAGD) due to its need for description of polynomial surfaces and geometry manipulation beyond simple transformations.

---

[2] A continuously updated overview of active and inactive toolkits is provided by the OpenABM initiative and can be found under https://www.comses.net/resources/modeling-frameworks/ (Accessed February 20, 2020).

The current state of the art in the architecture field, as represented by Gerber et al. [64], Schwinn et al. [172], Snooks [183], and Vasey et al. [195] is therefore comparable to the mid-1990s in other fields. A consolidation of approaches and software frameworks, which would make ABM an established modeling approach, and an alternative or, at least, extension to parametric modeling, has yet to happen in the field of agent-based architectural design.

9.3 Agent-based Framework

Previous work at the ICD, notably the work by Baharlou [11], Schwinn et al. [172], and Vasey et al. [195], demonstrated the applicability of the ABMS-approach for fabrication-oriented design with applications ranging from self-organizing building systems to the use of ABM for generative design and cyber-physical fabrication control. At the same time, it also showed the need for developing a common framework: on the one hand the methods were very similar, but on the other the implementations were project-specific and incompatible. This made it difficult to generalize from the specific projects and limited reusability of code.

Consequently, a coordinated effort was started in spring of 2016 to consolidate previous developments and develop a software framework based on the requirements of CAGD, in which different agent-based applications could be implemented. In this way, not only would duplication of effort be minimized by providing the computational 'mechanics' for developing and running different agent systems, but also the results of experiments and findings (such as successful agent types and behaviors) would become comparable and transferable between research projects.

The software implementation and initial findings produced using the framework were published in 2017 by Groenewolt et al. [72]. The ensuing description of the framework is a summary and further elaboration.

9.3.1 Aim and scope of the framework

Similar to Minar et al. [134], the aim ICD's ABM framework is to provide a shared platform for experimentation with agents and behaviors – a "scientific apparatus" that facilitates development, comparison and replication of results.

The principal approach is to separate the development of the software from running experiments in an effort to make the results of experiments comparable and transferable between research projects. This way, development of ABMs moves away from the previously mentioned one-off implementations that are built from scratch and that are inherently conditioned by the programming skills of the researcher and by the complexity of the task. Additionally, building on top of a collaboratively developed code-base allows for agent-systems that implement higher-level concepts such as evolution and learning, or complex interfaces with analysis software packages, which are out of reach of one-off implementations, especially in the context of architectural practice.

The development of the framework is an ongoing interdisciplinary collaboration between architects, engineers and computer scientists, and is funded through research projects, in which agent-based modeling is part of the research approach. While each research project focuses on specific research questions and thus represents different domain-specific applications of ABMS, such as plate systems, adaptive buildings, design of pre-cast concrete elements, etc., they all share a core code library, which provides the basic mechanisms and constructs from which domain-specific models can be built and ensures compatibility.[3]

The main usage scenarios align with the classification of architectural agent systems presented in chapter 3.4. Main applications of the ABM Framework so far have been experiments where building parts are conceptualized as agents that pro-actively try to meet user-specified goals. In such agent systems, many agents interact according to a given interaction topology, and as a consequence change shape and location. In the model's converged state, the resultant configuration of the agent system and its associated geometric representation then represent a solution to a given design problem, e.g., a building design or the segmentation of a shell. Another scenario that is being intensively studied are systems where agents represent real builders, such as mobile robots in distributed fabrication systems. A third scenario, which up to this point has only cursorily addressed, are systems where agents represent *virtual* builders, which according to Reynolds' classification [158, p. 2] are real agents that only exist in the virtual world, and iteratively deposit virtual material that ultimately represents an architectural design. Further scenarios, e.g., where agents represent stakeholders in the design process have not been studied so far, but should be able to given the generic structure of the framework.

9.3.2 Structure of the framework

The framework is object-oriented in nature and, in order to allow for extensibility and domain-specific applications, is composed of a set of class libraries organized around a core agent library (Fig. 9.1).

As described by Groenewolt et al. [72], the core library provides the basic constructs of agent models: agent, agent system, environment, behavior, and solver class types. The latter takes care of scheduling updates of the system. The former four class types are defined as abstract classes, from which application-specific classes can be derived. The core library also contains a few pre-defined classes such

---

[3] At the time of writing, ICD researchers Long Nguyen, Abel Groenewolt, Mathias Maierhofer, and Tobias Schwinn have contributed to the Framework.

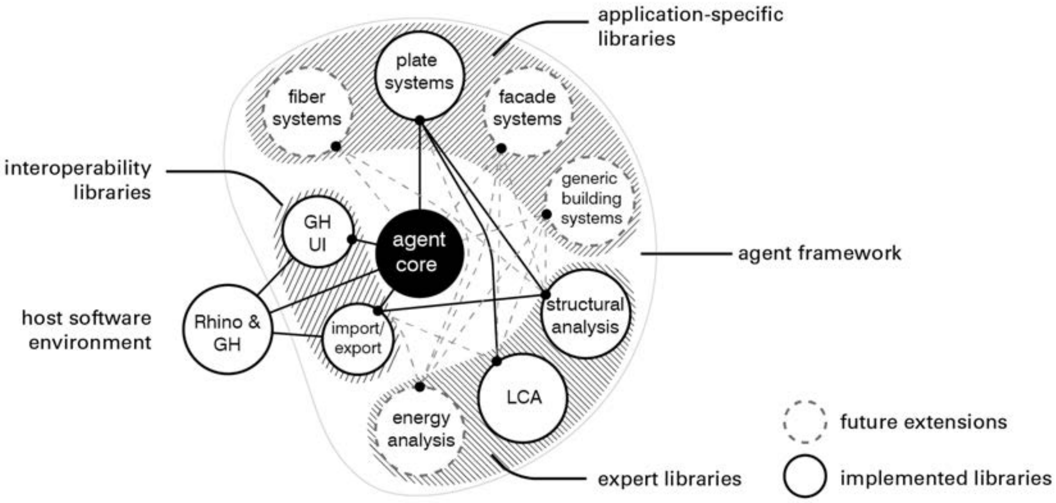


Figure 9.1: Structure of the agent framework. From [72] by Schwinn (2017).

as a "boid" agent class and a subset of the canonical flocking behaviors described by Reynolds [158], which allow for quick implementation of flocking models that can serve as baseline for further development.

The core library also pre-implements force-based locomotive and position-based Cartesian behaviors. The former is intended to be used in simulations of dynamic locomotive systems, e.g. schools of fish, while the latter can be used in applications of self-organization and optimization that are intended to converge to a local minimum. Specifically, the position-based approach avoids oscillations and overshooting that are common in force-based systems.

The core library is extended by libraries that fall into three categories: *application-specific libraries*, such as plate systems, etc., *expert libraries* that encapsulate expert software, such as structural analysis, as well as *support libraries*, which include UI-, import- and export functionality, and interfaces to functionality provided by the host software environment [72, p. 164].

An example for an application-specific library are the classes that were developed for conducting the experiments as part of this research, where agents represent building parts (shell segments) and connections between building parts, and behaviors represent rules by which building parts should be arranged; together, the agents form an agent system that represents a segmented shell design.

The Framework is implemented in C# as an add-on to Grasshopper, the node-based visual programming plug-in for Rhino, which was also used in the development of the cases studied in Part II (see 5.5.7), and developed within Microsoft's Visual Studio development environment. The integration of the framework into Rhino provides access to its CAGD capabilities, such as NURBS-based free-form curve and surface modeling, and to its 3D viewport. Furthermore, the integration into Grasshopper provides access to other relevant Grasshopper add-ons.

9.4 Case Study Setup

The aim of the case study is to investigate a higher integration between global form finding and local plate morphology by taking advantage of the so-called ‘plate-lattice dualism.’ As mentioned above, in previous agent-based models of segmented shells global form finding was separated from the agent model that generated the segmentation: first, a NURBS surface was designed, e.g. using dynamic equilibrium methods, acting as a reference surface, which was then populated by plate agents thereby ‘approximating’ the underlying reference surface through segmentation. This approach resulted in a unidirectional workflow, which limited feedback between local plate morphology driven by agent behaviors and global form. Examples for this approach are the agent models of the Landesgartenschau Exhibition Hall 2014 (Case Study II), the Rosenstein Timber Pavilion (2017), as well as, most recently, the BUGA Wood Pavilion in Heilbronn (2019). The ICD/ITKE Research Pavilion 2011 (Case Study I) used a hybrid approach, in which global form generation was influenced by local segment topology, i.e. to which other plate cells a given cells is connected, but the actual plate shapes were generated after form finding. As stated in Case Study I, this approach resulted in a comparatively large number of plates and also didn’t allow for bottom-up form generation.

The expected value added of the proposed integration therefore is to steer global form generation in a bottom-up way through local plate behaviors while resulting in a moderate number of plates that is comparable to the plate shells developed since 2014.

The case study is structured as a series of small experiments with increasing complexity in order to be able to understand the consequences of each modeling step and starts from a *null model* without behaviors [132, p. 35].

9.4.1 Software Used

Throughout the experiments four main software tools are used:[4]

1. The previously mentioned Rhinoceros (Rhino) for 3-dimensional geometry representation and manipulation;
2. Grasshopper for modeling the design’s principal logic using visual programming;
3. ICD’s Agent-based Modeling Framework; and
4. Microsoft’s Visual Studio (Visual Studio) as an Integrated Development Environment (IDE), through which the ABM Framework can be extended.

Grasshopper and Rhino provide the user interface for the ABM Framework

---

[4] For detailed software specifications see Appendix B.

and the model view, both for the 3D representation (output) and for the model's dependency graph (the model's internal logic). Visual Studio provides code editing, debugging and build automation functionality and is a commonly used platform for software development on Microsoft Windows operating systems.

The design principles for segmented shells, which have been postulated as the outcome of the research described in Part II, can be implemented as a combination of pre-defined Grasshopper components, custom code blocks (using C#, Visual Basic or Python programming languages), and custom-defined components that are part of an application-specific extension library to the ABM framework on the Grasshopper canvas (see 9.3.2).

It is important to note that the data flow structure of an ABM is to a large extent static, regardless of the application of ABMS and the often no-linear nature of agent-based systems, and it can essentially remain unchanged, even when may major changes to agent behaviors, types and interaction topology are implemented (Fig. 9.2). In comparison to the feed-forward graph of a conventional parametric model produced by visual data flow modeling, the feed-forward graph of an ABM is radically simplified (Fig. 8.1). This is a relevant aspect, as Davis [46, p. 48] reports that parametric models "often become so brittle that starting over is easier than making a change." ABMs on the other hand, even when implemented in visual data flow environment such as Grasshopper, allow for more robust model structures and consequently can help reducing or even avoiding the down-time arising from changes to the model logic that is common in parametric modeling.

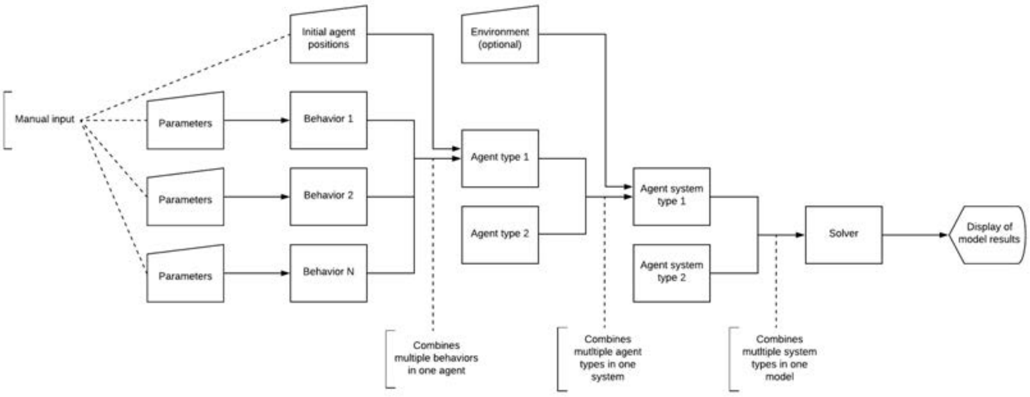


Figure 9.2: Generic structure of an agent-based model defined in the ABM Framework.

9.4.2 Global Design Decisions

As part of the formal modeling step, a few global design decision need to be made that relate to the purpose of the model. In this case, the purpose is to find an equilibrium state where multiple objectives are met, effectively corresponding to a local minimum of an unknown multi-objective function. Objectives include the double curvature of the shell as a structural prerequisite for shell structures, the validity of the plates in terms of their planarity and lack of self-intersections, the control the plate orientation at the supports, the reduction of variance in edge lengths, and a desired variation of plate sizes, e.g. in relation to curvature and/or to distance from the system edge. Examples of additional objectives are meeting the range of valid plate sizes (if sizes of stock materials and workspace of fabrication equipment are known) and a minimum edge length to accommodate joints.

9.4.2.1 Force-based vs. Position-based Approach

One of the first global design decisions of the case study concerns the question of position-based vs. force-based approach. This is akin to asking, ought the design task to be conceptualized from the point of view of optimization as a form of "self-organization," where the focus is on the minimization of a cost function, or is the focus more on simulating and exploring the dynamics of a complex system that is potentially constantly in flux, shows characteristic oscillations, or other consistent patterns? For the purpose of this case study, a position-based approach is chosen thus aiming at self-organization and convergence of the agent system to a state of equilibrium. The inherent difference of the two approaches is that in the force-based approach a vector representing *force* is acting on the velocity of the agent resulting in its acceleration. In this approach, the change of position between time steps is the result of manipulating the second derivative of the agent's position w.r.t time; whereas in the position-based approach, a vector representing *translation* is acting directly on the position of the agent providing a more immediate access to its position and a more reliable convergence behavior.

9.4.2.2 Agent representation

Agents are conceptualized as building elements, in this case plates, that constitute a plate structure. The state of an agent is defined by geometric information such as the closed polyline (i.e. a list of 3d points with straight-line segments in between with coincident start and end points) representing the plate outline, its size and diameter, and by its position and orientation represented as a *base* coordinate system (agent frame) located in the above mentioned Cartesian World coordinate system. Additional non-geometric parameters such as material, weight, or temperature can also be used to define the state, but have not been considered in this case.

From a formal modeling point of view, the `DualAgent` class is derived from the Framework's `AgentBase` class. In addition to the fields present in the CartesianAgent class, it implements the agent's `Frame` and `PlatePolyline` fields and overrides the `PostExecute` method to update the agent's orientation, which together with the agent's `PreExecute` and `Execute` methods is called in each iteration. Whereas the `PreExecute` method holds code that is to be executed at the start of each iteration for each agent and is mainly used for "house-keeping", the `Execute` method holds the main code calling the behaviors stored in the `Behaviors` list. After the `Execute` method has been called for each agent, the `PostExecute` method updates each agent's position, orientation, and plate polyline. This sequence of methods ensures that the agents in the system will update synchronously, i.e. that all behaviors have access to the same system state at the same time step and that the effects of the behaviors are considered simultaneously. On the other hand, this setup doesn't preclude asynchronous or threshold-based update mechanisms, if they are required. Finally, the `Neighbors` property calls the private method `getTopologicalNeigbors`, which queries the DualAgent system's topology object, and returns the list of neighboring agents.

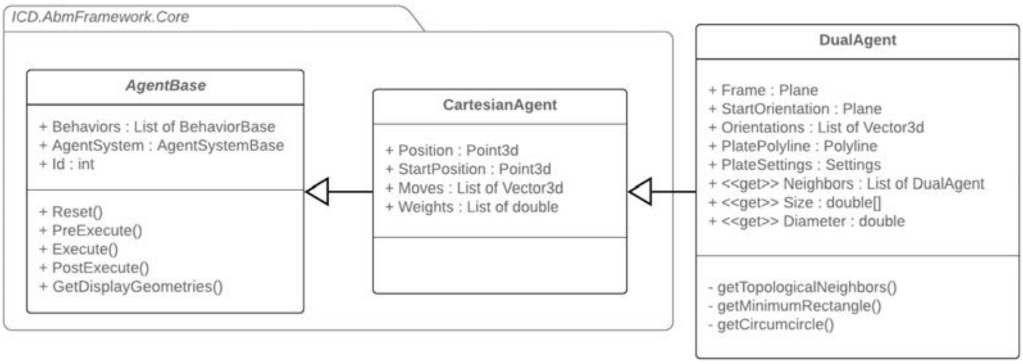


Figure 9.3: UML class diagram of the DualAgent class. The DualAgent class is derived from the abstract class AgentBase.

9.4.2.3 Behaviors

As per the definition of behaviors (see 3.2.4), behaviors are the internally coordinated response to internal (the agent's state) and external (other agents and environment) stimuli thereby affecting its own state (position, orientation, and plate polyline), other agents, and the environment (in this case the interaction topology).

From a formal modeling point of view, all behaviors described in this case study are derived from the Framework's abstract `BehaviorBase` class providing the basic fields and abstract `Execute` method, which needs to be overridden in each concrete behavior and which is called by the agent's own `Execute` method. Figure 9.4 shows an example of three behaviors used in the following experiments that are derived from the abstract behavior class.

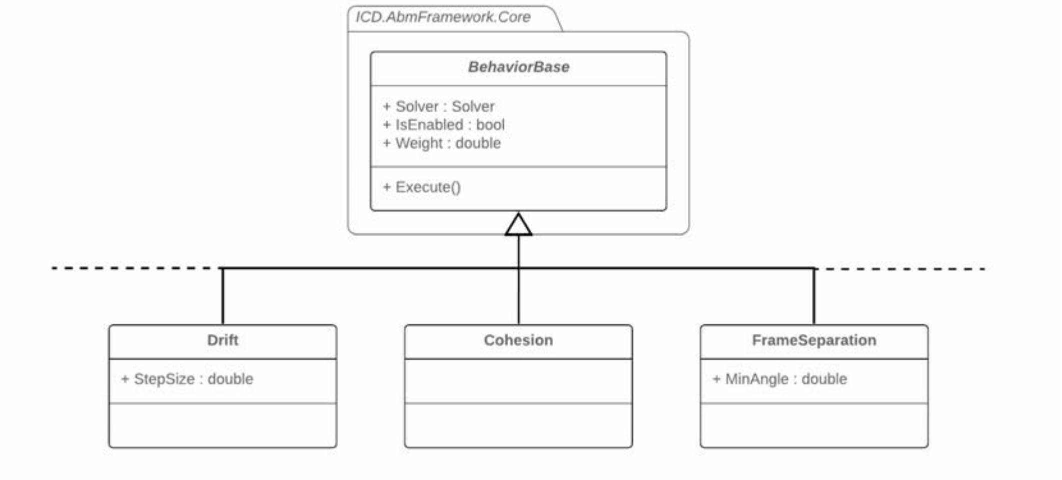


Figure 9.4: UML class diagram of the behavior classes. The figure shows an example of three behaviors, which are used in the experiments, that are derived from the abstract class BehaviorBase.

9.4.2.4 Environment and Interaction Topology

The environment is the "substrate", in which the agents are embedded and interact. In this case, the environment is a continuous, 3-dimensional Euclidean space. Situated in the environment, agents are conceptualized as active nodes in a network of nodes and edges, which defines the interaction topology.

In the planar system state at iteration 0, the interaction network is initialized by Euclidean distance, i.e. the ordinary straight-line distance between two agents in space, as a *Delaunay* triangulation in the World XY Cartesian plane. This means that agent positions are triangulated in such a way that no circle defined by a triangle's nodes contains another node. In this way, nodes are reliably connected to their nearest neighbors without causing self-intersections of edges and at the same time agent 3-tuples are defined that are required by the chosen plate generation method (see experiment 3). An agent's neighborhood, i.e. the set of agents that it is interacting with, is then defined by the network's topology, that is by the edges in the network that connect it to other agents. It is thus named the topological neighborhood. The interaction topology remains constant throughout the update cycles unless a behavior is acting on it (see experiment 7). Other definitions of the agent neighborhood are possible, such as the Euclidean neighborhood, which is defined as the set of agents within a given Euclidean distance, the von Neumann-neighborhood or the Moore-neighborhood in discrete environments. The way in which the neighborhood is defined has an effect on how the agents self-organize in 3d-space and thus on the resultant shell form.

It is interesting to note that in previous instances of agent-based segmentation design that were mentioned above, the environment was conceptualized as the NURBS-surface, forming a middle-layer between global design and local plate morphology, on which the agents were situated. With the absence of a NURBS-surface, not only can the environment be radically simplified, but global form can be conceptualized as the unmediated bottom-up result of local interaction between the agents.

9.4.2.5 Agent System

Conceptually speaking, an agent system is defined by a group of agents, their interaction topology and behaviors, and the environment in which the agents are situated. Certainly, multiple agent systems can interact within the same environment.

More formally, the `DualAgent` system class is derived from the Framework's `AgentSystemBase` class and, in addition to the fields provided by the `CartesianAgent` system class, implements fields that store a mesh-based interaction topology, the topological status of each agent (i.e., if it is surrounded by other agents or not), a list indicating if an edge needs to be swapped, and a system-specific threshold value that provides a stop criterion for the system solver. The `SystemMesh` object uses the Mesh type provided by Rhino to store agent connectivity instead of the Connectivity type used in the CartesianAgent system class as it stores additional information such as faces, normals, and naked edge point status and it provides methods such as SwapEdge that can be used to modify the interaction topology. `IsNaked` is a boolean array of the size of number of agents indicating if an agent is located at the perimeter of the system (i.e., is naked) or not, and `Switches` is a boolean array of the size of the number of edges in the system indicating if an edge is to be flipped at the end of the update cycle. The system's `PostExecute` method calls the agents' corresponding method, which updates the agents' position and orientation, then updates the connections in the system's interaction topology, and finally the agents' plate polylines using updated positions, orientations, and connectivity. This concludes one update cycle.

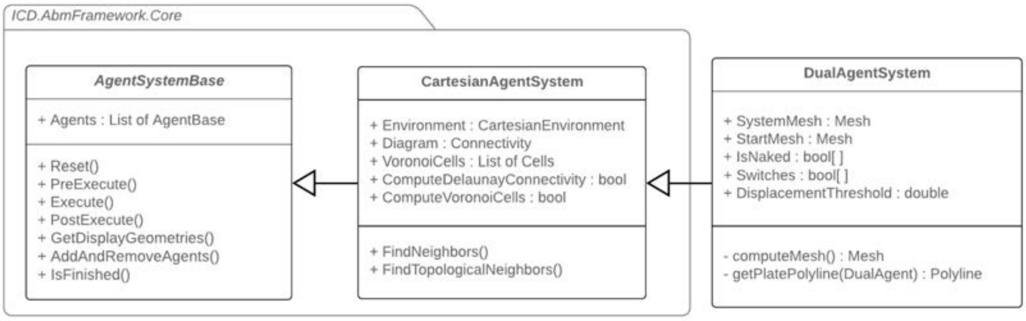


Figure 9.5: UML-Diagram of the DualAgent system class, which is derived from the abstract class AgentSystemBase.

9.4.2.6 Starting Configuration / Null Model

The starting configuration of the experiments, the so-called “null model”, consists of 61 agents of type `DualAgent` without any behaviors that are initialized at the vertices of a given triangulated Rhino Mesh object.

Circular in layout with a diameter of 5000 units, the design of the initializing mesh aims to be minimal yet universal enough to be able to represent both synclastic and anticlastic shell designs—the two predominant scenarios of surface curvature in shells, while at the same time providing a number of agents significant enough to be able to generate a discernible system effect. The mesh object, which defines interaction topology as described above, together with the list of agents initialize the DualAgent system. The anticlastic example applies the design principle of Edge Control and uses a 3d boundary curve as a guide for the boundary agents (see 8.6). In all subsequent experiments, the agents at the perimeter of the system represent the shell’s support and are defined as anchors (i.e. lacking behaviors that modify their positions), but, depending on the experiment, allow for re-orientation.

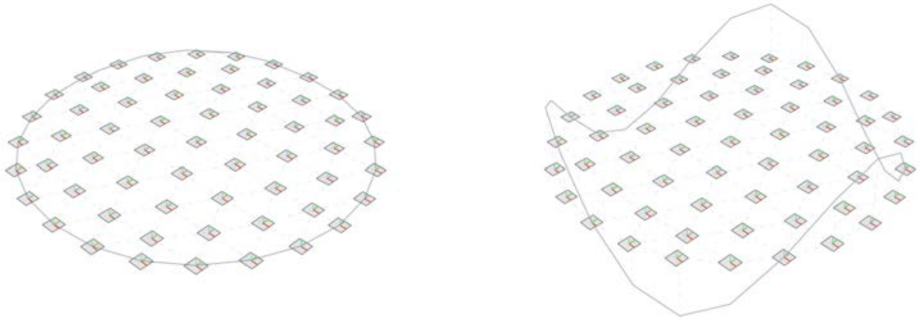


Figure 9.6: Starting configuration or ”null model” of the experiments described below.

9.4.2.7 Evaluation Metric

In order to be able to evaluate and compare different plate configurations, a metric is introduced focusing on the variation of edge lengths in the system: the average of the standard deviation of edge lengths per plate, subsequently called SD_{el}. The standard deviation of edge lengths $SD_{el,j}$ for plate j indicates how spread out the edge lengths in the given plate are, with a low $SD_{el,j}$ indicating similar edge lengths and a high value indicating a greater variance of edge lengths in a plate. For example, a regular hexagon where each edge has the same length has an $SD_{el,j}$ of 0. The average of all $SD_{el,j}$ then indicates how similarly shaped the edges of the plates in the system are. Importantly, this metric is independent of the absolute size of the plates. The metric can also be used to formulate an optimization objective, e.g. in terms of minimizing SD_{el} (see experiment 9).

For illustration, an average SD_{el} of 100 mm indicates that 68% of all edges in the system lie within an average of 100 mm of each plate's mean edge length (assuming a normal distribution of edge lengths). Correspondingly, 95% would lie within 200 mm of each plate's mean.

9.5 Experiments

9.5.1 Experiment 1: Creating Double Curvature

The aim of the first experiment is to define an agent system where agents form a double-curved configuration in space, which is a requirement for shell design and for the chosen plate generation method. This configuration is to be achieved exclusively based on local rules and without a priori definition of a reference surface.

9.5.1.1 Approach

As described in the previous case studies, double-curved synclastic shell designs can be achieved by simulating inverse gravity as part of a dynamic equilibrium method such as Particle-Spring. In the absence of a force-based approach in this case study, however, gravity is not conceptualized as a force acting on a particle resulting in its acceleration, but as a 'desire' of the agents to move in the positive z-direction by a given amount in each iteration, a behavior termed "Drift", thereby emulating a negative gravitational pull. The behavior thus computes a translation vector $\vec{t_d}$ parallel to the world's Z-axis:

$$\vec{t_d} = D(s) = \begin{pmatrix} 0 \\ 0 \\ s \end{pmatrix}, s \in \mathbb{R}^+ \tag{9.1}$$

where s is the step size. In order to avoid that the agents just "fly off" indefinitely as a result of this behavior, a second behavior termed "Cohesion" has been defined, which ensures that agents cohere. The behavior computes a translation vector $\vec{t_c}$ as a function of the neighbor positions that moves the agent to the average of its neighbors' position (similar to Laplacian smoothing):

$$\vec{t_c} = C(P_N) = \alpha * \frac{1}{N} * \sum_{j=1}^{N} \overrightarrow{P_a P_j}, \quad P, \vec{t} \in \mathbb{R}^3; \alpha \in \mathbb{R}^+; N \in \mathbb{N}^+ \tag{9.2}$$

where α is a scale factor, N the set of neighbor agents defined by the interaction network, and P_j the position of the j-th neighbor. The agent position is updated at each iteration as the weighted average of the two behaviors 9.1 and 9.2.

$$\begin{aligned} P_{i+1} &\leftarrow P_i + \frac{a * D(s) + b * C(P_N)}{a+b}, \quad && a, b \in \mathbb{R}_0^+ \\ \text{subject to} & && a + b > 0 \end{aligned} \tag{9.3}$$

where P_i is the position of an agent at iteration i, and a and b are weight factors.

In the limit, the translation vector computed by the “Cohesion” behavior will point in the opposite direction of the one computed by the “Drift” behavior thereby eventually cancelling each other out. At this point the system will come to a standstill and is considered to have converged. The cut-off criterion is a total displacement, computed as the sum of squared distances, of less than 0.0001 units. Consequently, the step size s of the drift behavior controls the height the dome in the synclastic example. In the anticlastic example, the weight factor a of the drift behavior is set to 0 and agents thus are updated using only the Cohesion behavior.

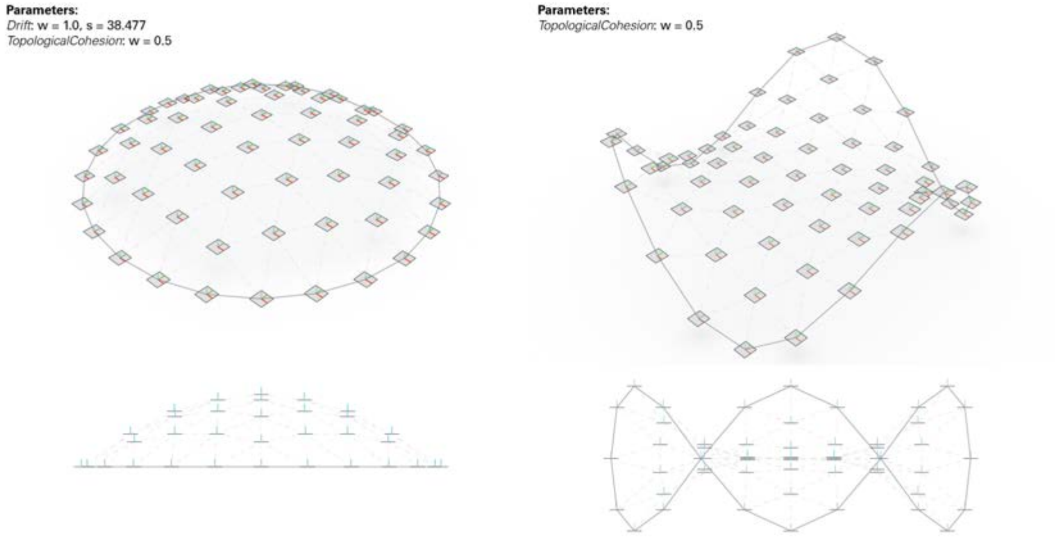


Figure 9.7: Double-curved agent system configurations. Left: synclastic example using the two behaviors “Drift” and “Cohesion”. Right: anticlastic example only using “Cohesion” behavior. The agents at the perimeter are considered anchors, i.e. they don’t change position.

9.5.1.2 Results

In the synclastic example, with the two behaviors combined and the support agents fixed, the system converges into a synclastic configuration in 261 iterations resulting in a funicular arrangement of agents in space with a height of 1000 units with s = 38.77 units resulting in an aspect ratio of 1:5 (shell height over span).

In the anticlastic example, which omits the drift behavior, the model converges after 24 iterations resulting in a minimal surface-type configuration that spans membrane-like between the boundary curve (9.7).

9.5.2 Experiment 2: Orientation Behaviors

The aim of the second experiment is to determine the orientation of agent frames in relation to their neighboring agents in addition to the agents' positions, which was described in the previous experiment.

9.5.2.1 Approach

Eight different approaches to define agent orientation have been tested, which fall into three principal categories. The first category uses the system mesh object as a reference, the second uses explicit rotational separation, and the third variations of interpolation.

1. Orientation based on system mesh:
   (a) by aligning agent orientation with the vertex normals of system mesh, a behavior termed "VertexNormals",
   (b) by aligning agent orientation with the average of the adjacent face normals ("FaceNormals").
2. Orientation based on rotational separation:
   (a) by rotating each agent frame away from its neighbors to achieve a minimum rotational separation between agents ("MinFrameSeparation"),
   (b) by adding an additional constraint to approach 2.a) to limit the maximum rotational separation ("MinMaxFrameSeparation"),
   (c) by explicitly setting the rotational separation to a desired value ("TargetFrameSeparation").
3. Orientation based on interpolation:
   (a) by interpolating neighbor orientations ("FrameInterpolation"),
   (b) by weighted interpolation of agent orientations taking into account the distance to the neighbors ("WeightedFrameInterpolation"),
   (c) a combination of rotational separation and interpolation, by projecting neighbor frame normals onto a vertical plane aligned with an underlying vector field, averaging projections, and rotating a set amount in the direction of the local vector ("VectorField").

In principle, all approaches can produce valid results for the agent frame orientation, however not all are useful in every case with respect to generating plates as will be shown in experiment 3. Additionally, the interpolation behaviors require that the boundary agents have a pre-set constant orientation or else all orientations remain flat or will eventually become flat.

The examples below show the rotational separation behavior “TargetFrameSeparation.” The orientation of the frame is a function of the neighbor orientations. The behavior calculates a new normal vector $\vec{n}$ for the agent frame as the average of individually rotated normals based on the angular difference between the minimum desired rotational separation and the current separation with a given neighbor.

$$\begin{aligned} \vec{n} = O(O_N) &= \frac{1}{N} * \sum_{j=0}^{N} R_{\hat{n}}(\varphi) * \vec{n_a} \\ \text{with} \qquad \hat{n} &= n_a \times n_j \\ \text{and} \qquad \varphi &= \theta - \vec{n_a} \cdot \vec{n_j} \end{aligned} \tag{9.4}$$

where $\mathbb{R}_{\hat{n}}(\varphi)$ is the rotation matrix in $\mathbb{R}^3$ around the unit vector $\hat{n}$ defined as the cross product of the normal $\vec{n_a}$ of the agent and the normal n_j of neighbor j; and θ is the target angle. Agent orientation is therefore updated as a function of the orientation of its neighbors at iteration i:

$$O_{i+1} \leftarrow O(O_{N,i}) \tag{9.5}$$

Synclastic and anticlastic examples use the same “FrameSeparation” behavior in addition to the behaviors described in experiment 1.

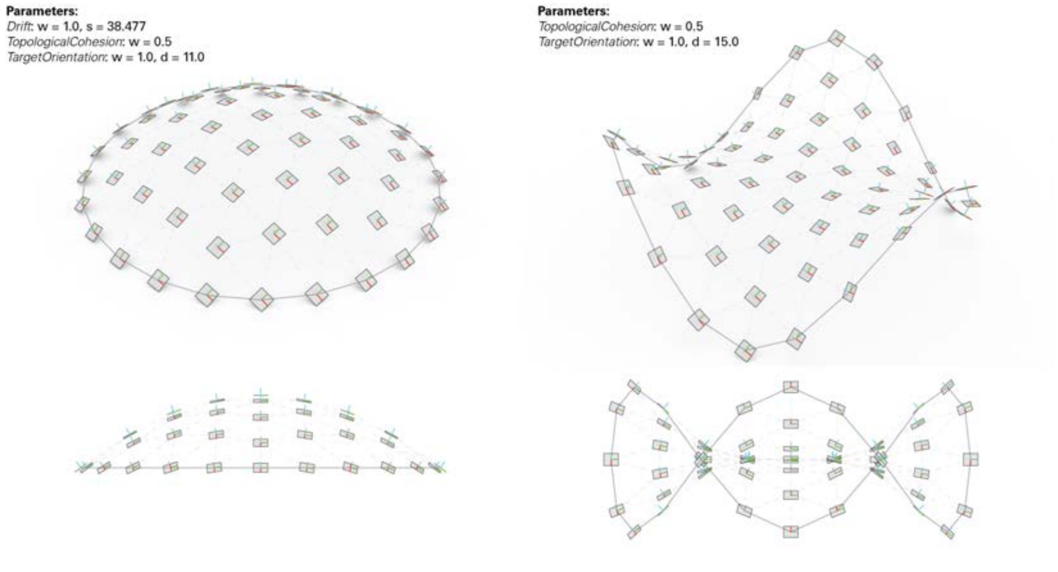


Figure 9.8: Orientation behaviors. Left: synclastic example using the orientation behavior “FrameSeparation” in addition to “Drift” and “Cohesion”. Right: anticlastic example only using “FrameSeparation” in addition to “Cohesion” behavior.

9.5.2.2 Results

In the synclastic example, which uses the three previously described and equally weighted behaviors “Drift”, “Cohesion”, and “FrameSeparation”, the system converges in 261 iterations into the funicular arrangement while meeting a target rotational separation of 10 degrees.

In the anticlastic example, which uses the previously described and equally weighted behaviors “Cohesion” and “FrameSeparation”, the system converges in 24 iterations into the membrane-like configuration while meeting the target rotational separation of ±15 degrees.

9.5.3 Experiment 3: Plate generation

The aim of third experiment is to generate a “watertight”, i.e. gapless, arrangement of plates in space as a function of the position and orientation of the agents in the system.

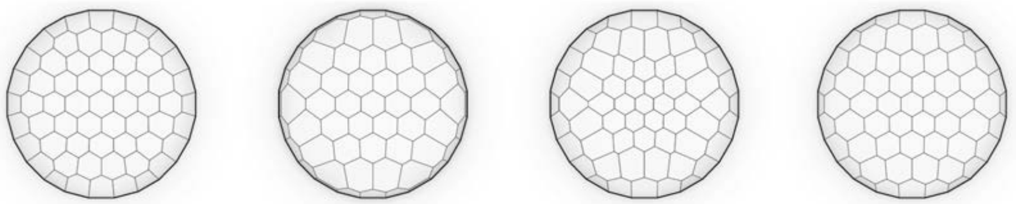
9.5.3.1 Approach

In line with the plate-lattice dualism, every edge in the interaction network can be conceived of as a shared edge between two plates in the shell and every node, i.e. agent, in the network as a plate. The method that is implemented here is a modification of the TPI method described in CS II, which similarly results in planar segments and is a dual representation of the interaction network. Similar to TPI, in this approach a plate vertex is defined by the intersection of 3 planes located at the vertices of a triangle (3-tuple of agents). A plate thus is defined by the “faces” (3-tuples) that surround a given node in the network. In contrast to the TPI method, where an agent’s frame is oriented tangentially to the given design surface at the agent’s location, which limits the agent’s degrees of freedom (DOF) to two, in the proposed variation an agent is free to re-position and re-orient using all 6 DOF of Euclidean space $\mathbb{E}^3$. The method is subsequently referred to as the *3-Plane Intersection* method (3PI) to indicate that the method is free from the tangency constraint.

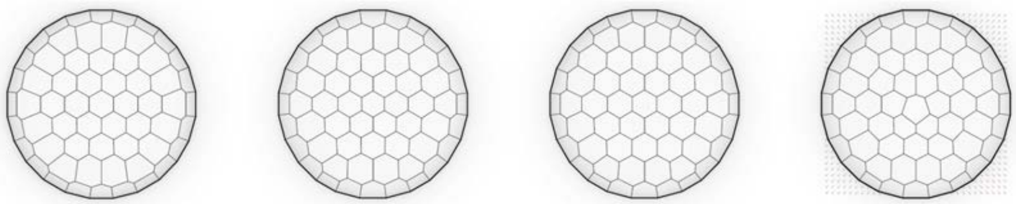
Conceptually speaking, plate generation can be conceived from the perspective of each agent as one of its behaviors, i.e. as a method of the agent object. In the formal modeling step, however, the plate generation method has been delegated to the `DualAgentSystem` object as it is dependent on the system’s interaction topology. In other words, an agent cannot have a plate without the presence of an interaction topology. Similarly, the ABM Framework is set up in such a way that an agent cannot exist without the system it belongs to, the agent class thus being a child of the system class. [5].

---

[5] Formally speaking, this corresponds to a *Composition* relationship between agent and system



VertexNormal SD_{el} = 76.07 mm MinFrame SD_{el} = 156.48 mm MinMaxFrame SD_{el} = 130.53 mm TargetFrame SD_{el} = 99.67 mm



FaceNormal SD_{el} = 107.52 mm FrameInterpolation SD_{el} = 96.62 mm WeightedInterpolation SD_{el} = 89.40 mm FlowField SD_{el} = 107.81 mm

Figure 9.9: Comparison of plate configurations based on different orientation behaviors. The orientations of the boundary plates of systems that use Interpolation behaviors including “FlowField” are defined explicitly.

Different orientation behaviors result in different plate shapes while all systems have the same network topology and global shape. Figure 9.9 shows different plate patterns for the synclastic scenario resulting from the different orientation approaches described in experiment 2. As mentioned previously, orientations of the boundary plates of systems that use Interpolation behaviors including “FlowField” need to be set explicitly. The detailed example (Fig. 9.10) shows the plate pattern resulting from the “TargetFrameSeparation” behavior described in experiment 2 for both synclastic and anticlastic scenarios.

Similar to experiment 2, the synclastic example, which uses the three previously described and equally weighted behaviors “Drift”, “TopologicalCohesion”, and “TargetFrameSeparation”, converges in 261 iterations into the funicular arrangement while aiming for a target angle between plates of 11 degrees. The SD_{el} of this plate configuration is 99.67 mm.

The anticlastic example, which uses the previously described and equally weighted behaviors “Cohesion” and “TargetFrameSeparation”, shows the characteristic bow-tie-shaped plate outlines and converges in 24 iterations into a saddle-shaped configuration while aiming to enforce the target angle between plates of ±18 degrees. The SD_{el} of this plate configuration is 228.59 mm.

9.5.4 Experiment 4: Equalize plate outlines (Move-to-Centroid)

The aim of the forth experiment is to control the shape of the plates through variations of a “Move-to-Centroid” behavior, a heuristic that has been inconclusively studied

---

(denoted in UML class notation with a closed diamond)

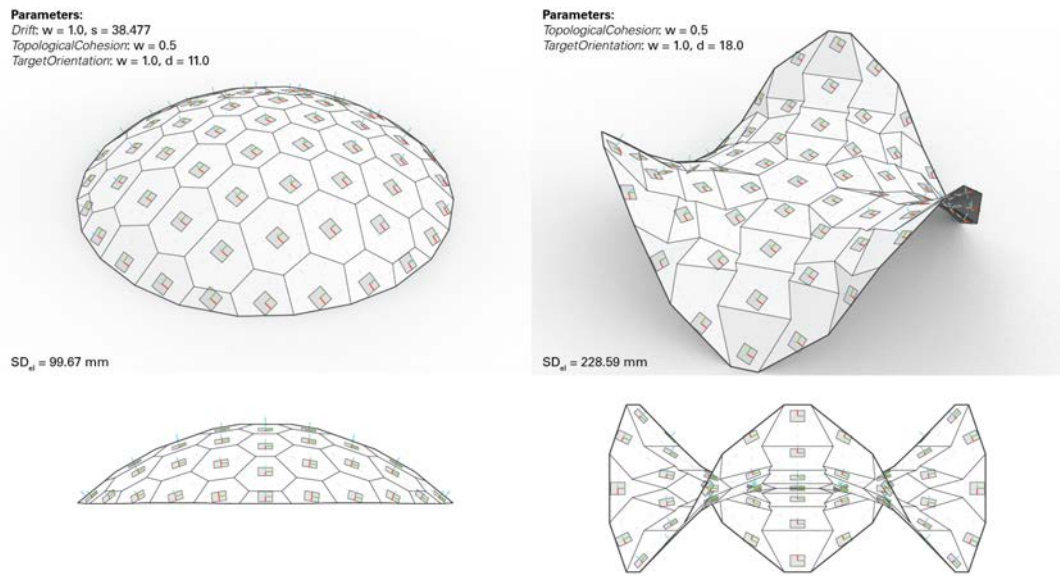


Figure 9.10: Plate Generation. Left: synclastic example using “TargetFrameSeparation”, “Drift”, and “TopologicalCohesion” behaviors. Right: anticlastic example only using “TargetFrameSeparation” and “TopologicalCohesion” behaviors.

in previous research.

9.5.4.1 Approach

Previous work has shown that in synclastic surface areas a behavior that iteratively moves an agent to the centroid of its plate reduces the variation of its edge lengths, in other words it minimizes its SD_{el}. A similar approach is used in the well established “Lloyds algorithm”, which confirms that this is also the case in two and three dimensions of Euclidean space [162].

The centroid of a polyline can be calculated from its vertices as their average position, from its edges as a weighted average, or from its area as the center of gravity. The “CentroidByVertexAverage” behavior computes a translation vector $\vec{t_v}$ as a function of the plate polyline’s vertices pointing towards their average position:

$$\vec{t_c} = M(P_{pl}) = \alpha * \frac{1}{N} * \sum_{j=1}^{N} P_a \vec{P}_{pl,j}, \quad P, \vec{t} \in \mathbb{R}^3; \alpha \in \mathbb{R}^+; N \in \mathbb{N}^+ \tag{9.6}$$

where α is a scale factor, P_{pl} the set of vertices of the plate polyline, and $P_{pl,j}$ the position of the j-th vertex. The position of the agent is updated as the weighted average of all position-based behaviors:

$$\begin{aligned} &P_{i+1} \leftarrow P_i + \frac{a * D(s) + b * C(P_N) + c * M(P_{pl})}{a + b + c}, && a, b, c \in \mathbb{R}_0^+ \\ &\text{subject to} && a + b + c > 0 \end{aligned} \tag{9.7}$$

where P_i is the position of an agent at iteration i, and a, b, c are weight factors. As expected from experiments in previous work, this should have an equalizing effect on the plate's edges.

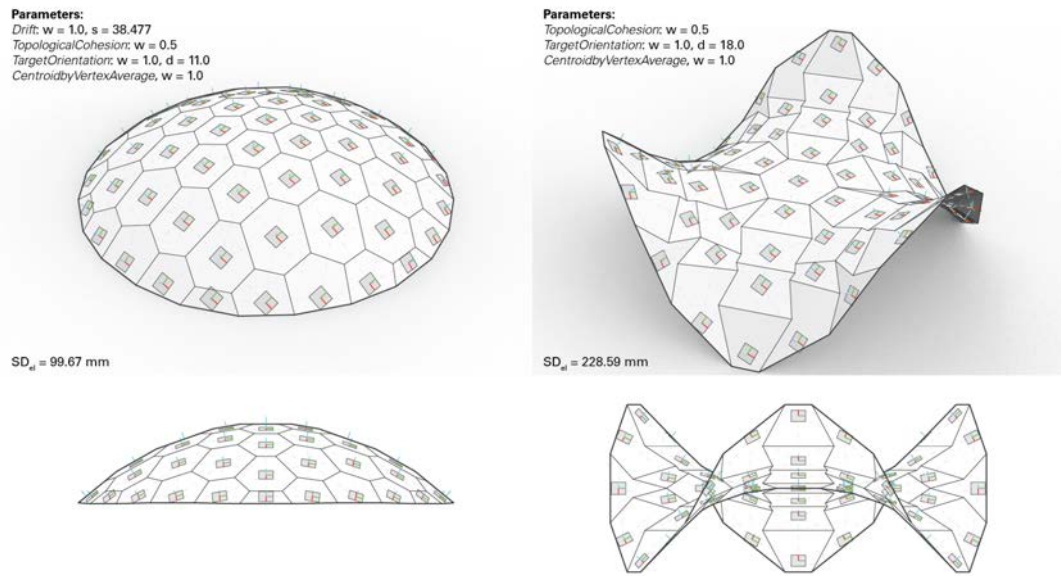


Figure 9.11: Move-to-Centroid behavior. Left: synclastic example using the "CentroidByVertexAverage" behavior in addition to the previously assigned behaviors. Right: anticlastic example using the same Move-to-Centroid behavior as the synclastic example in addition to previously defined "TargetOrientationSeparation" and "TopologicalCohesion" behaviors.

9.5.4.2 Result

It is at first surprising to observe that the move-to-centroid behaviors do not seem to affect the plate outlines at all: the plate configuration of experiment 4 is the same as in experiment 3 even though the agents are in different locations. At second thought, however, it become clear that an agent can be positioned anywhere on a plane that is coplanar with its frame without affecting the shape of its plate, since the points of the plate polylines are computed from the intersection of planes and planes have infinite extent. In other words, movements of the agent in the plane of its frame as, e.g., caused by the Move-to-Centroid behaviors, don't affect the plate polyline.

The important finding from this experiment therefore is that there is an infinite amount of agent positions (parameterized by u, v) that an agent can have in the co-planar plane. Therefore, of the six DOF only 4 variables are responsible for the generation of plates: the frame's normal direction as parameterized by x, y, z and its distance from the World origin. In other words, there is an infinite amount of possibilities to generate the same possibly idealized plate configuration. This significantly increases the probability that a suitable configuration can be found by a search algorithm.

Another important finding is that in order to achieve a change in the plate configuration, the Move-to-Centroid behaviors would need to be coupled with orientation behaviors that correlate the orientation of the frame with the relative position to its neighbors, such as “WeightedFrameInterpolation” behavior. In the case of TPI the underlying reference surface correlates the orientation of the agent to its position. The crucial question then, however, is how, in the absence of a designed surface this correlation should occur, specifically, how the distance to the neighbors should be weighted, i.e. linearly, quadratically, etc., or inversely. It seems that another approach would be preferable, which finds the right correlation between agent position and orientation that doesn’t rely on an explicitly defined (surface) function.

9.5.5 Experiment 5: Constrain supports

The aim of the fifth experiment is to constrain the orientation of agents at the supports in order to explicitly control how, i.e. in what angle, the segmented shell meets the ground. The need for such control was shown in the Landesgartenschau Exhibition Hall where only for a small variation of plate angles relative to the ground plane were permissible due to the connection type of the plates to the edge beam.

9.5.5.1 Approach

The orientation of a support agent is controlled by pre-orienting the frame that is used as part of the constructor of the agent object. If, additionally, no behaviors are assigned to the support agent that affect its orientation, then the agent will maintain its a priori set orientation. Combined with the absence of behaviors that modify position, such an agent can be considered a rotational and positional “anchor.” Figure 8.10 shows the result of this approach both for synclastic and anticlastic examples.

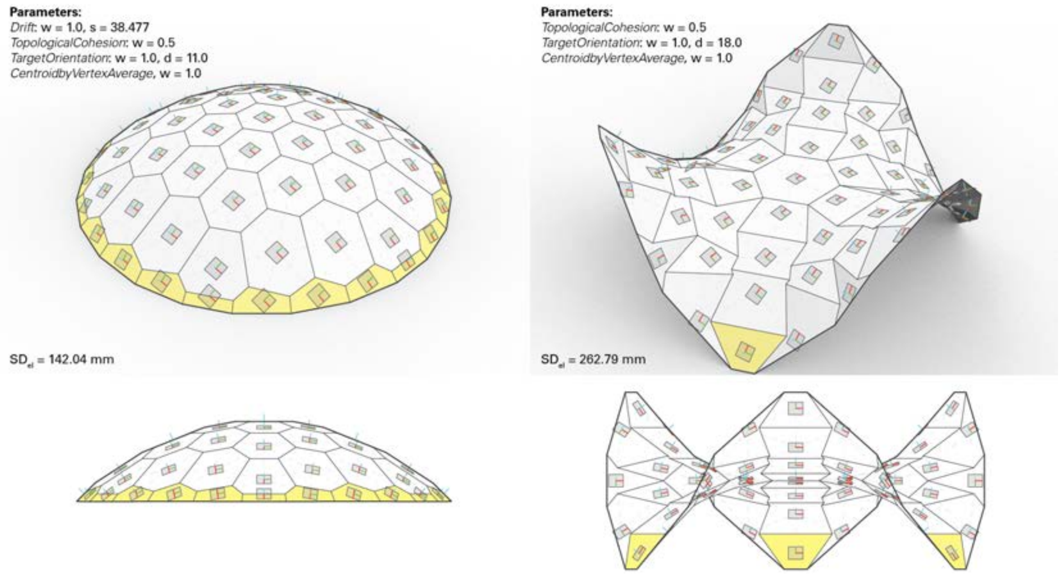


Figure 9.12: Support constraint. The position and orientation of the highlighted agents have been set a priori. Left: synclastic example using the previously assigned behaviors. Right: anticlastic example using the previously assigned behaviors but constraining only those agent that meet the ground plane.

9.5.5.2 Result

While all behavior parameters remained unmodified, the explicit definition of the support agents' orientations constitutes a change of the boundary conditions of the system. As expected, this is reflected in the changed SD_{el} of both curvature scenarios.

This experiment is an example for the advantages that the 3PI approach, which exploits the duality between interaction topology and plate shape, offers in terms of providing additional degrees of geometric freedom compared to the TPI method. These ultimately result in additional degrees of design freedom.

9.5.6 Experiment 6: Constrain intersections

The aim of the sixth experiment is to constrain the plate polyline to a maximum diameter by limiting the displacement of so-called "outlier points."

9.5.6.1 Approach

As previously described, each vertex of a plate polyline is the intersection point P_I of three agent frames (3-tuple) that are associated with a triangle in the neighborhood of the corresponding plate agent. In this setup, if at least two of the planes are close to being co-planar, then P_I might be far from the agent's position relative to the other vertices of the polyline. In fact, if agent frames are co-planar then the 3PI method would fail (similar to the TPI method) because parallel planes have no intersection, in which case the triangle center would be used instead to define the polyline's vertex. In both cases this results in a non-planer plate polyline. In order to determine if P_I is to be considered far away, the following metric is used: if the distance of P_I to the corresponding triangle center P_C is more than the radius r_c of the circle C_T defined by the triangle's vertices, then the position of P_I is updated to P'_I using a behavior that scales the vector $\vec{P_C P_I}$ by a scale factor α.

$$P'_I \leftarrow \alpha * r_c * \frac{\vec{P_C P_I}}{|\vec{P_C P_I}|} + P_C \tag{9.8}$$

The scale factor α is computed as a function of the distance $|\vec{P_C P_I}|$ and in the limit is asymptotic with 2.0, i.e. twice the radius of C_T. Specifically, the following equation is used to compute α:

$$\begin{aligned} \alpha = f(x) \quad &= \frac{4}{\pi} * atan(\frac{x}{r_c}) \\ \text{where } x \quad &= |\vec{P_C P_I}| \in \mathbb{R}^+ \end{aligned} \tag{9.9}$$

such that $f(1.0) = 1.0$ and $lim_{x \to \infty} f(x) = 2.0$. Figure 9.13 shows the graph of the scale factor equation for illustration. This way, all the intersection points that constitute the plate polyline lie at a maximum distance of $2 * r_c$ from the center of the triangle.

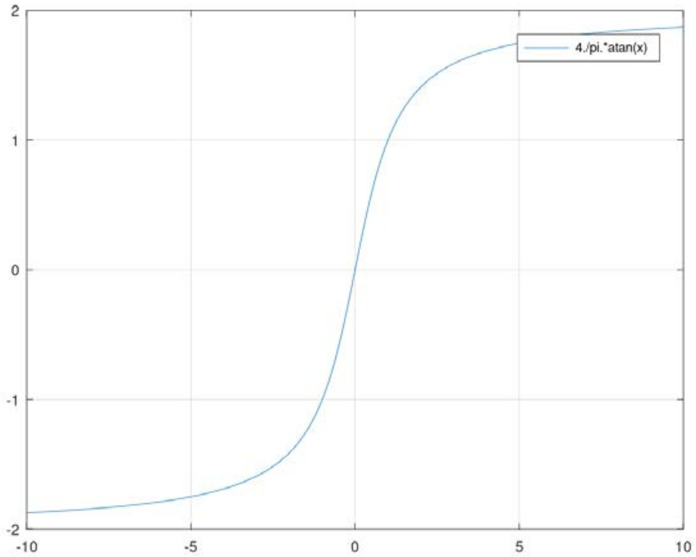


Figure 9.13: Graph of the equation defining the scale factor α. The graph is centered on 0 and asymptoting at ±2, however only $\mathbb{R}^+$ is relevant for the rule which scales the vector $\vec{P_C P_I}$, where $lim_{x \to \infty} f(x) = 2.0$.

The result of this behavior is a plate polyline that is non-planar but watertight nonetheless, because adjacent agents use the same rule to compute the shared polyline vertex from the given interaction triangle. This raises the question of how to ensure planarity, i.e. the plate's validity, as this is one of the stated objectives of the experiments.

In the proposed approach, the orientation of the agent's frame is updated such that the sum of squared distances between the agent frame and the polyline vertices is minimized. The corresponding behavior termed "FitFrame" uses RhinoCommon's built-in method `FitPlaneToPoints`, which is part of the `Geometry.Plane` class, to fit the agent frame to the non-planar intersection points. Applied iteratively, the behavior results in planar plates that meet the constraint defined above, where each vertex of a plate polyline is located within the circle defined by the vertices of the corresponding interaction triangle.

9.5.6.2 Result

It turns out that, in the synclastic case, the application of scaling constraint and "FitFrame" behavior results in some self-intersecting plate polylines as shown in figure 9.15. To remedy the self-intersections constitutes the aim of experiment 7.

The focus of this experiment is therefore on the anticlastic example, where the application of the scaling constraint makes some plate polylines non-planar, as ex-

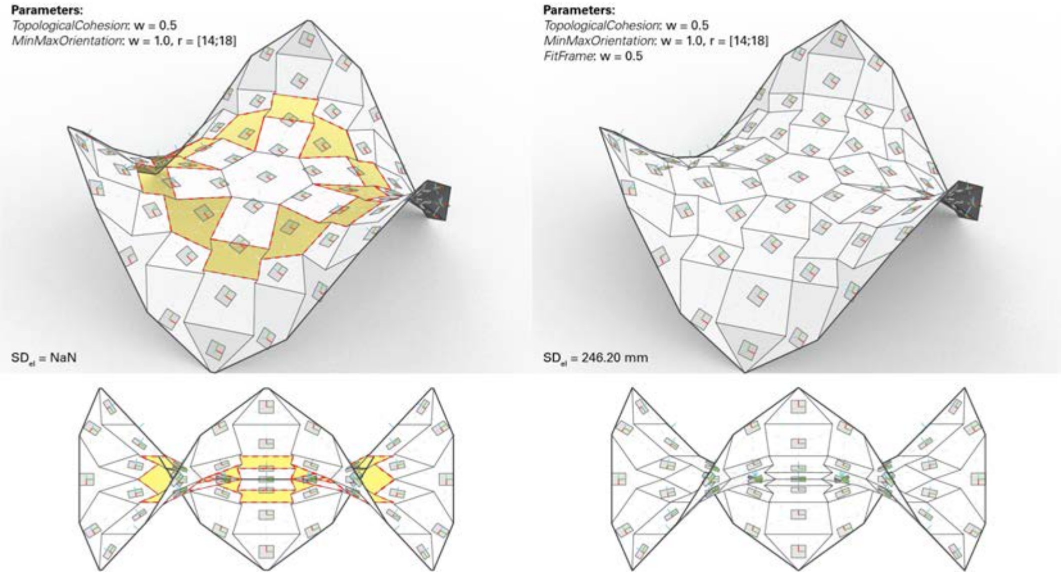


Figure 9.14: Scaling constraint and "FitFrame" behavior. Left: The application of the constraint results in some non-planar plate polylines (highlighted). Right: The application of the "FitFrame" behavior results in valid planar plates.

pected, which are highlighted in the diagram in Fig. 9.14 (left). It is important to note that this experiment uses the frame orientation behavior "MinMaxFrameOrientation", which produces a plate configuration that exceeds the "Point-in-circle-of-triangle" constraint, because the "TargetFrameOrientation" behavior would not. The "FitFrame" behavior then achieves planarity of the plates after a total of 63 iterations Fig. 9.14 (right). The application of the "FitFrame" behavior therefore ensures validity of the plates, while the scaling constraint associates the extents of the plate polyline with the geometry of the underlying interaction network.

The correlation of plates sizes with the resolution of the interaction network could be the basis for a system level behavior that programmatically adds agents to the system relative to the resulting plate sizes (as opposed to, e.g., the distance between agents), which will be explored in future work.

9.5.7 Experiment 7: Edge swapping

The aim of the seventh experiment is to define a rule by which edges in the mesh-based interaction topology can be swapped in order to solve possible self-intersections of plate polylines.

9.5.7.1 Approach

It has been hypothesized in previous work that the following edge swapping rule would result in plates free of self-intersection: "For every pair of edge-connected triangles, if the distance between the opposing triangle vertices is smaller than the length of the shared edge, then the edge connection should be swapped." During work on this experiment, however, it turned out that this is not the case. Therefore, instead of comparing the length of edges, this experiment proposes to observe the plates themselves if they show self-intersections and then act on the underlying interaction topology.

Self-intersections of plate polylines are possible, as shown in figure 9.15, given that

1. the order of the polyline vertices corresponds to the anti-clockwise order of the “faces” of the interaction network that surrounds a given agent,
2. the corresponding intersection points meet the intersection point constraint mentioned above.

Furthermore, the duality between interaction network and plate configuration states that the polyline segments correspond to the edges in the network that connect an agent to its neighbors. If a self-intersection is present, it can therefore be resolved by swapping the network edge that is the dual of the edge that is surrounded by the edges that self-intersect (Fig. 9.15, left).

The proposed approach to resolve the intersection first determines if a self-intersection is present in the plate polyline using RhinoCommon’s `Intersection.CurveSelf` method. If an intersection is present, then the edge indices of the segments that intersect are determined in the second step. Given the duality, the index of the segment that is enclosed by the intersecting segments corresponds to the edge index in the network that needs to be flipped. This rule also applies in case multiple intersections occur in a plate. However, it does not apply in case more than one segment is enclosed by the intersecting edges. Finally, if it is determined that an edge needs swapping, this is indicated in the DualAgent’s Boolean array `Switches` (see Fig. 9.6) by setting the array index that corresponds to the edge index of the interaction network to `True`, so that it can be swapped at the end of the iteration.

9.5.7.2 Result

In the experiment, the edge swapping behavior adaptively changes the interaction topology during runtime and the system converges in 411 iterations. The experiment focused on the synclastic case since the previous experiment’s synclastic scenario produced the starting point for this experiment. The rule, however, is similarly applicable to the anticlastic scenario should a self-intersection arise.

The most relevant finding of this experiment is that the interaction topology is not fixed but can be acted upon and modified locally by individual agent behaviors. The decision who is interacting with whom, or which plates share edges, can therefore be negotiated and renegotiated at the local level. Ultimately, this is what should be expected given the individual focused modeling perspective of the ABM approach.

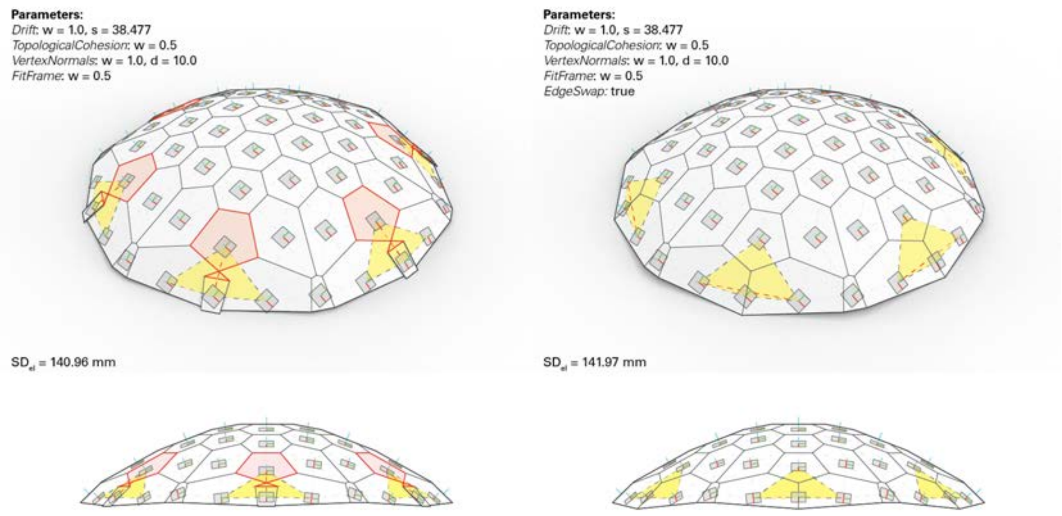


Figure 9.15: "Edge Swap" behavior. Left: Self-intersections of plate polylines are present. The edge that needs to be flipped to solve the self-intersection is the shared edge between the two highlighted triangles. Right: Applying the "EdgeSwap" behavior solves the self-intersection and results in valid plates. The figure highlights the flipped edge and corresponding plate configuration.

Furthermore, different rules by which agents modify the existing interaction topology could be explored. For example, the interaction topology could be made to respond to stresses in the shell's plates and modified in such a way that resulting joints between plates are aligned with the principal stress directions, as long as the validity of the plates is maintained (see design principle of Anisotropy). Such a behavior would result in a anisotropic plate configuration.

9.5.8 Experiment 8: Plate size gradient

The aim of the eighth experiment is to control the size of each plate in the system such that gradients can emerge, e.g. based on the distance of an agent to the system edge. The example being the pavilion studied in CS I, where one stated objective was morphological adaptation, which in this case meant that the plate cells ought to be smaller at the perimeter of the system than in the center, a rule that was derived from the biological role model (see also Design Principle Cell Sizes).

The hypothesis in this experiment is that plate sizes are a function of the distances between agents, therefore distances should be smaller close to the perimeter and larger, away from it.

9.5.8.1 Approach

In order to test this hypothesis, a behavior termed "VariableTopologicalSeparation" has been created that computes a specific target separation value s_a for each agent a in the system as a function of its distance $d_{a,b}$ to the system border:

$$s_a = f(d_{a,b}) = B_{min} + (d_{a,b} - A_{min}) * \frac{(B_{max} - B_{min})}{(A_{max} - A_{min})} \tag{9.10}$$

where A and B are intervals representing the range of distances to the system edge and the range of desired separation values respectively, resulting in a linear "remapping" of distance values to separation values.

Consequently, an agent's position is updated to move towards a neighboring agent if the actual distance to the neighbor is more than the separation value calculated at the current position; and away from the neighboring agent if the current distance is less than the desired separation.

$$P_{a,i+1} \leftarrow P_{a,i} + \frac{1}{N} * \sum_{j=1}^{N} (|\overrightarrow{P_{a,i}P_{j,i}}| - s_a) * \overrightarrow{P_{a,i}P_{j,i}} \tag{9.11}$$

where N is the set of neighbors, $P_{a,i}$ the position of the agent a at iteration i, and $P_{j,i}$ that of the j-th neighbor. Applied iteratively, the agent system converges to an agent distribution where agent spacing is denser at the perimeter and less dense at the center.

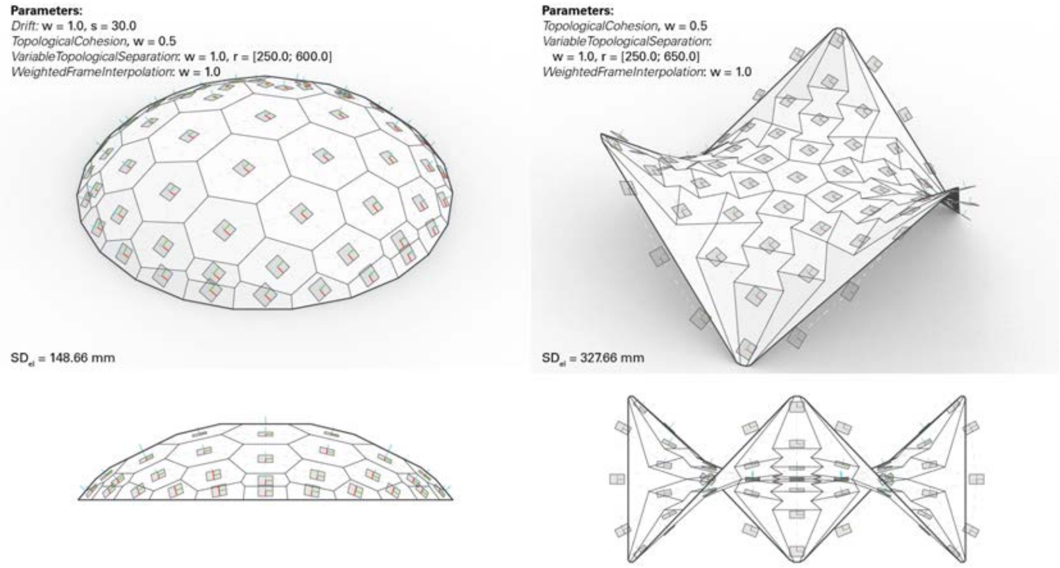


Figure 9.16: Gradient plate size behavior. Left: Synclastic example. Right: Anticlastic example.

9.5.8.2 Results

In the synclastic example, using the "VariableTopologicalSeparation" behavior in combination with the previously used behaviors "Drift", "Cohesion", and "WeightedFrameInterpolation", which correlates orientation to the distance of an agent to its neighbors, the system converges in 61 iterations to the a plate configuration where plates are smaller at the perimeter and larger at the center. It is important to note that in order to achieve this result the "Drift" behavior had to be significantly modified. Through trial-and-error, it was found that a distribution of agent positions that results in less “curvature” at the apex of the shell, while increasing curvature close to the supports would result in the desired plate configuration (Fig. 9.16, left). The behavior thus needed to be adjusted correspondingly.

In the absence of a “Drift” behavior , this strategy did not produce the desired outcome in the anticlastic example even though the same “VariableTopologicalSeparation”, “TopologicalCohesion”, and “WeightedFrameInterpolation” behaviors were used (Fig. 9.16, right). While the resulting plate configuration is interesting per se, it is unclear what the right combination of position and orientation behaviors would need to be in order to achieve the desired system result. A further experiment could thus explore the effect of increasing the number of agents at the perimeter in order to produce smaller plates, and removing agents from the center.

The experiment showed that the size of the plates is not only a function of the position of the agents but also of their orientations. While it is possible to create a point distribution where agents are closer to each other at the perimeter and further spaced at the center, it is not clear in the anticlastic example how exactly position and orientation would need to be correlated in order to achieve a targeted result. Furthermore, it is unclear how using such an approach, higher level goals, such as a specific target plate size per agent, could be achieved.

9.5.9 Experiment 9: Optimize plate shape

The aim of the ninth experiment is to optimize system parameters and to find optimal parameter combinations that reduce the SD_{el} in order to produce the most regular plate configuration possible under the given boundary conditions.

9.5.9.1 Approach

With the number of agents considered a constant, the number of parameters in the synclastic example, which uses “Drift”, “TopologicalCohesion”, and “TargetOrientation” behaviors, is 5 (including their weights). This means, in principle, that 5 parameters need to be individually adjusted to search for a desired plate configuration, which corresponds to a 5-dimensional solution space, a.k.a. fitness landscape. While it is still possible to manually ‘tweak’ these parameters, given their relatively low number, e.g., in order to get an intuition of what the optimal combination might be, it is fairly unpractical to find an optimum considering that parameters values might need to be adjusted up to the third decimal point.

The objective of this experiment is therefore to tune these system parameters automatically in order to minimize SD_{el}. As described above, SD_{el} is a function of the previously described behaviors “Drift” (itself a function of step size s parameterized by weight w_d), “Cohesion” (a function of neighbor positions P_N

parameterized by weight w_c), and "Orientation" (a function of neighbor orientation O_N parameterized by desired separation θ and weight w_o):

$$
\begin{aligned}
&\min_{s, w_d, w_c, \theta, w_o} \quad SD_{el} = f(D(s|w_d), C(P_N|w_c), O(O_N|\theta, w_o)) \\
&\text{subject to:} \\
&\qquad s, \theta \in \mathbb{R}^+ \\
&\qquad w_d, w_c, w_o \in [0, 1] \\
&\qquad P_N, O_N \in \mathbb{R}^3
\end{aligned}
\tag{9.12}
$$

Given that the actual function is unknown, a derivative-free, model-based "black-box" optimization approach is chosen. Model-based methods construct a "surrogate model" (i.e., an approximation of a fitness landscape) that is refined during the optimization process. Wortmann [214] has shown that such an approach is the more efficient and robust compared to genetic algorithms, which are quite popular in the architectural design optimization community despite their significantly weaker performance.

The algorithm that is used is RBFOpt (Radial-Basis-Function Optimization) developed by Costa and Nannicini [44] and implemented as part of the "Opossum" (Version 2.1) add-on to Grasshopper by Wortmann.

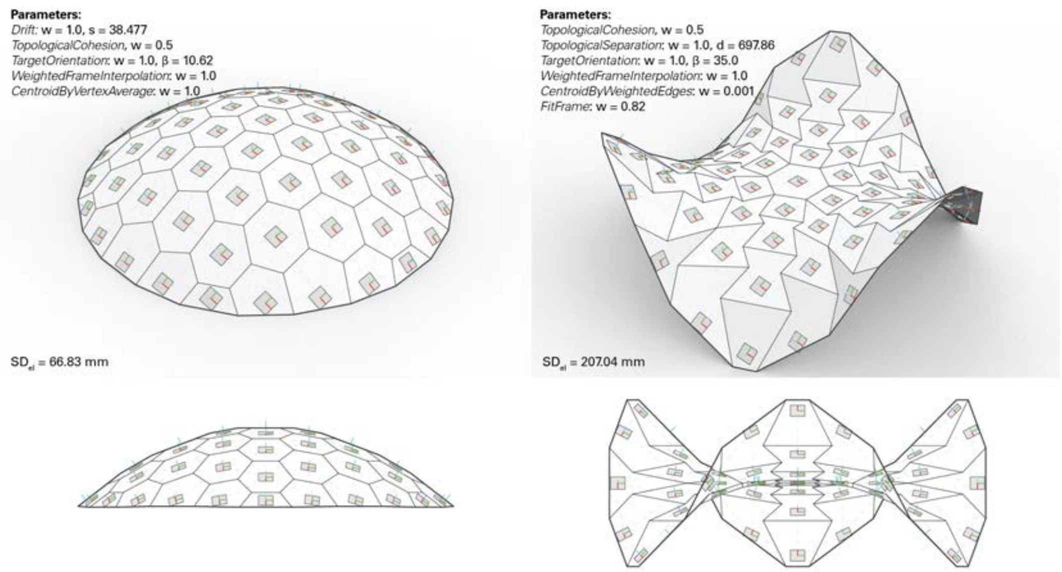


Figure 9.17: Results of parameter optimization. Left: Synclastic example resulting in an SD_{el} of 66.83 mm. Right: Anticlastic example resulting in an SD_{el} of 297.15 mm.

9.5.9.2 Result

In the synclastic example, the modeling process consists of three episodes: in the first episode, the step size s of the "Drift" behavior is optimized to approximate a target shell height of 1000 mm, by minimizing the distance to the target height. In fact, the resulting value of 38.477 mm has been used throughout the earlier experiments. In the second episode, with the “Drift” and “TopologicalCohesion” parameters a given, the desired separation angle parameter of the “TargetOrientation” behavior is optimized. Using the resultant value of 10.62 degrees as desired separation angle, the system converges in 261 iterations to an SD_{el} of 95.21 mm. The third episode uses the result of the second episode as a starting point and combines the “ByVertexAverage” move-to-centroid behavior with “WeightedFrameInterpolation” while constraining the orientation of the support agents. The “WeightedFrameInterpolation” behavior computes the orientation of an agent frame not only as function of the orientation of its neighbors but also of their distances to the agent. The system converges in 55 iterations and further reduces the SD_{el} to 66.83 mm. This is the lowest value that has been observed in all experiments.

In the anticlastic example, the modeling process consists of two episodes: in the first episode, with the parameters of “TopologicalCohesion” and “TargetFrameSeparation” a given, the optimizer searches for the desired distance parameter of the “TopologicalSeparation” behavior that minimizes the SD_{el} of the system. Using the resultant value of 697.86 mm as distance parameter, the system converges in 2 (!) iterations to an SD_{el} of 231.01 mm. The second episode uses the result of the first episode as a starting point and combines the “ByWeightedEdges” move-to-centroid behavior with “WeightedFrameInterpolation” orientation and “FitFrame” behaviors while constraining the orientation of the support agents. The system converges in 1260 iterations and further reduces the SD_{el} to 207.04 mm.

The main finding of the experiment is that the model-based black-box optimization approach is in fact extremely powerful while being efficient given “a limited evaluation budget” [214, p. 414]. While in both examples a maximum of two parameters have been optimized at the same time focusing solely on the single-objective of minimizing the SD_{el} of the plate system, the approach could easily be extended to include many more parameters and objectives. On the other hand, the experiment also shows that the choice of the designer, in terms of which behavior to optimize, is crucial, while the selection process itself might be less than optimal.

Therefore it is suggested to not only externalize the fine-tuning of parameter values, but also the trial-and-error approach on which the finding of right behavior combinations is based. In fact, it is proposed to go one step further and also externalize the definition of the behaviors themselves using an approach called Reinforcement Learning.

9.5.10 Experiment 10: Learning behaviors

The aim of the tenth experiment is to investigate the potential of Reinforcement Learning (RL), more specifically Q-Learning, for the development of goal-oriented behaviors that can be deployed as agent behaviors in the ABM Framework.

9.5.10.1 A Brief Introduction to Q-Learning

Reinforcement Learning (RL) is an established branch of Machine Learning that differs from its two other main branches Supervised and Unsupervised Learning in that the learning process does not rely on a given training set, labeled or unlabeled. Instead, the RL learning process itself generates the necessary data, in which an agent receives rewards based on the consequences of its actions in an environment. RL is conceived from the perspective of an individual agent that through trial-and-error explores the environment and tries to maximize its rewards. The underlying reward hypothesis states that "all goals can be described by the maximization of expected cumulative reward" [180].

In Q-learning, the environment is conceived of as being discrete, i.e. having a finite number of possible states. During the learning process the agent learns to correlate states of the environment and its actions that will lead it to the given target, which triggers a positive reward. Being in the environment is costly for the agent (negative reward), it thus tries to find the target as quickly as possible. While the choice of actions at the beginning of the learning process might be random (exploration phase), the agent learns which actions lead to a reward later based on the current state of the environment. This allows the agent to increasingly exploit that knowledge as the learning progresses (exploitation phase).

The expected reward for a given action at a given state is called the Q-value, hence the name Q-learning. At each iteration of a learning episode, the Q-value for a specific action at a specific state is calculated and stored in a table in the form of a $n \times m$ matrix, where n is the number of states and m is the number of possible actions. The learning process consists of multiple episodes, during the course of which the agent learns to maximize the reward. In this way, the agent learns a *policy*, that indicates which actions to take given a specific state of the environment. This is formalized in the following update rule for the Q-value $Q(s,a)$ for a given state-action pair:

$$Q(s,a) \leftarrow (1-\alpha)*Q(s,a)+\alpha*(reward+\gamma*\underset{a}{argmax}Q(s_{i+1}, all\ actions)) \tag{9.13}$$

where α is the learning factor, $reward$ is an arbitrarily chosen value, γ is the discount factor, and $argmax$ means choosing the action a that maximizes the Q-value at the next state out of all possible actions.

Figure 9.18 shows a diagrammatic view of the described Q-Learning process.

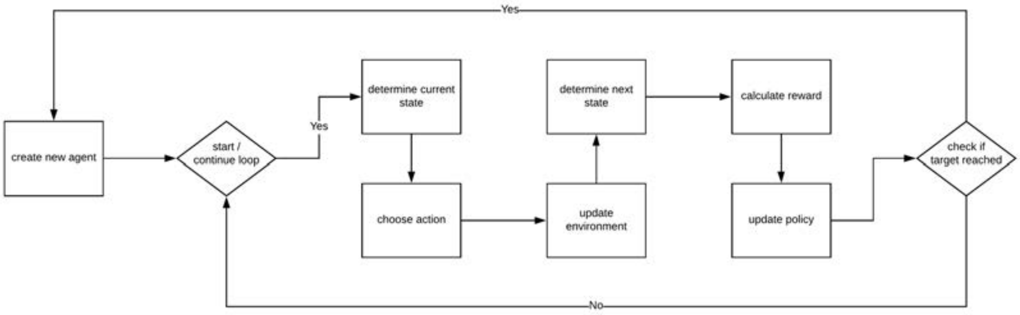


Figure 9.18: Diagrammatic representation of the learning process.

9.5.10.2 Approach

For the given objective of finding position and orientation of a plate agent that minimize its $SD_{el,j}$, the first task is to identify features of the agent model and of the environment that are assumed to be relevant for the learning of the agent. As the Q-Learning method assumes a discretized environment, this means finding *discrete* representations of *continuous* features (i.e. represented by floating-point values) such as positions of other agents or lengths of edges. For comparison, a discrete environment of 20x20 “pixels” where the agent can be located on exactly one pixel has 400 possible states. With the addition of one other agent, e.g. a predator, the number of possible states rises to 160.000 (assuming they can be on the same cell, in which case the predator would have caught the agent). This essentially means that an important task in defining the Q-Learning process is to represent the state of the environment in a way that limits the amount of possible permutations of features. This limit is essentially defined by the available computational resources.

One strategy for limiting the number of possible states is to allow only for a local perspective of the agent, i.e to intentionally limit its view of the world, which is the approach that is taken here. Importantly, this approach is consistent with the established definition of agents in Agent-based Systems (ABSs) as having a limited view of the world and acting on locally available information. RL therefore appears to be a ‘natural’ approach for learning in ABSs.

As a learning setup, one plate agent and its neighbors are chosen randomly from the set of agents in the synclastic plate system produced in experiment 3 for each episode (Fig. 9.9, left). The chosen features for the experiment are

1. the midpoint of the shortest edge,
2. the plate’s centroid,
3. the polyline vertex at corner of the largest internal plate angle,

4. the average of the neighbors' positions, and
5. the minimum bounding rectangle of the agent's plate (Fig. 9.19, right).

all defined in the local frame of the agent. In order to avoid that an agent perceives the same state as different states due to different orientations of its frame relative to the plate polyline, the agent frame is X-axis aligned with the polyline. The result of choosing two different alignment rules are compared: in the first, the direction to the midpoint of the shortest edge is chosen to align the agent frame; in the second, the frame is aligned with the direction of the plate edge that defines the orientation of the bounding rectangle.

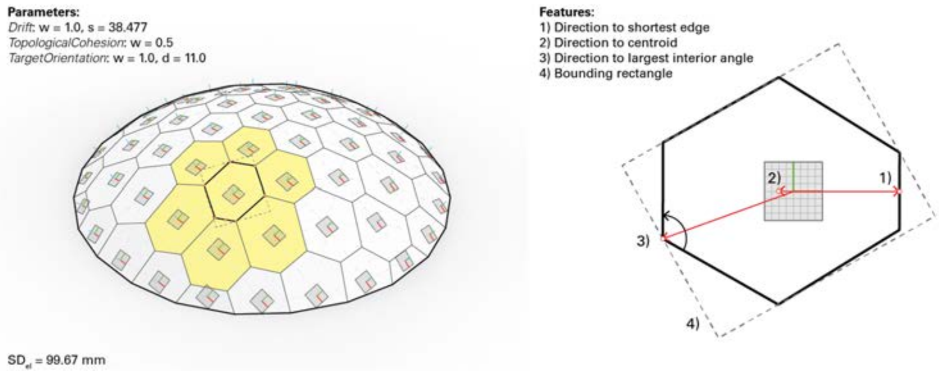


Figure 9.19: Learning setup. Left: One agent and its neighbors are randomly chosen at each learning episode. Right: the frame of the agent is aligned with the plate polyline, in this case the midpoint of the shortest edge.

The three other reference points can thus be either in the N (0,1,0), NE (1,1,0), E (1,0,0), SE (1,-1,0), S (0,-1,0), SW (-1,-1,0), W (-1,0,0), or NW (-1,1,0) direction from the point of view of the agent or the agent can be located on them (0,0,0). In the case of the average neighbor position, it can additionally be either above (0,0,1), below (0,0,-1) or on the agent frame. This very limited view of the environment that an agent has still results in a state space S of 2187 discrete states (the size of S is 3^7, or -1, 0, or 1 for 7 of the 9 variables).

At each iteration, the agent has the choice of changing its position by moving either N, S, E, or W in its aligned coordinate system or of changing its orientation by tilting its frame either in the N, S, E, or W direction. Any action would thus result in a change of the plate outline and thus in a change of the environment. However, the agent has also the option of not changing position and orientation to allow for the case in which the $SD_{el,j}$ can not be improved by any action (as per the definition of behavior, no action is also a possible reaction). This corresponds to a total of 9 possible actions and a Q-table of the size of 2187 x 9.

Furthermore, given that the exact target is unknown in this learning problem, i.e. the exact position and orientation that an agent should be in in order to minimize $SD_{el,j}$, a positive reward is given for every action that reduces $SD_{el,j}$ or else a small negative reward is given. In this way, the agent learns to move and orient in the direction that minimizes $SD_{el,j}$ for the current state, which is defined by its plate and its neighbors' positions.

In the implementation, the learning setup uses the Q-learning algorithm implemented as part of the Accord.NET ML Framework [185], which has been wrapped by the "Owl" (version 2.1) add-on to Grasshopper by Zwierzycki (2019). For the purpose of the experiment, the number of learning iterations is limited to 50.000. A learning episode is considered finished when an agent does not improve its $SD_{el,j}$ score for 20 iterations. At this point a new learning episode starts by randomly choosing a new agent from the set of plate agents and by initializing it at a random location on the original agent frame in the [-200;200] domain of its origin. This ensures new plate polylines as starting configuration even if the same agent is chosen multiple times during the learning process. In this way the agent will encounter many different polyline shapes and be able to better generalize its actions from the learning experience.

After learning has completed, the Q-table, which summarizes the learning experience of the agent, provides the policy by which the agent behaves, i.e. chooses its actions, and can be deployed in the ABM framework as part of a new "StateAction" behavior for plate agents. The behavior first evaluates, which state an agent is in by using the criteria mentioned above and then maps the state to an action to take using the Q-table. Applied iteratively, the expected result is that the agent system converges to a configuration where the SD_{el} can not be reduced any further.

9.5.10.3 Results

The results of the experiment show that a given plate configuration can be improved by autonomously learned agent behaviors (Fig. 9.20). Specifically, the plate system converges after 17 iterations and reduces the SD_{el} of the plate system in experiment 3 from 99.67 mm to 98.09 mm.

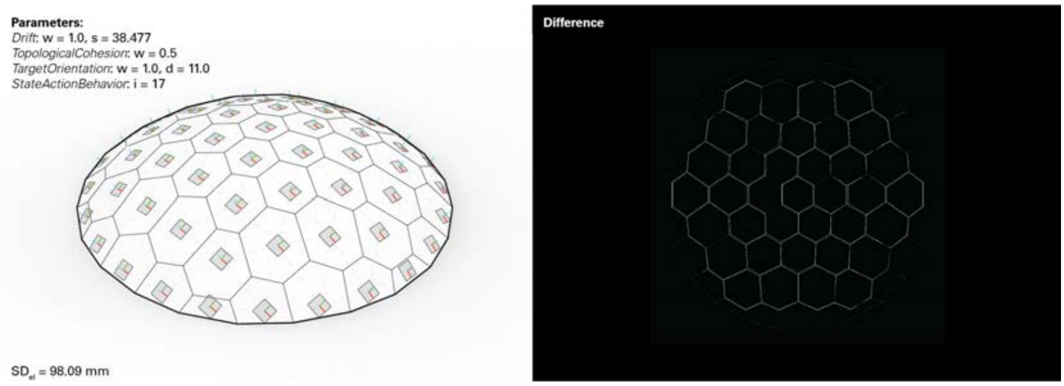


Figure 9.20: Result of the learning experiment. Left: Improved plate configuration. Right: the pixel-by-pixel difference to the starting configuration.

This improvement is obviously not enormous and the resultant value is considerably more than the result achieved using parameter optimization of “hand-designed” behaviors described in experiment 9 (66.83 mm).

This raises the question which point of the learning process needs to be modified to enable a more significant improvement. One obvious point is “feature engineering”, i.e. defining the number and selection of “features” of the environment that the agent perceives. Preliminary experiments show that selection of features affects the policy and thus the amount of improvement, which means that some features are more relevant than others to finding the optimal agent position and orientation. For example, using the midpoint of the shortest edge to orient the agent frame during the learning process resulted in the SD_{el} of 98.09 mm when deployed. Whereas using the bounding rectangle to orient the agent frame during the learning process did not result in an improvement of the SD_{el} compared to the starting configuration (Fig. 9.21).

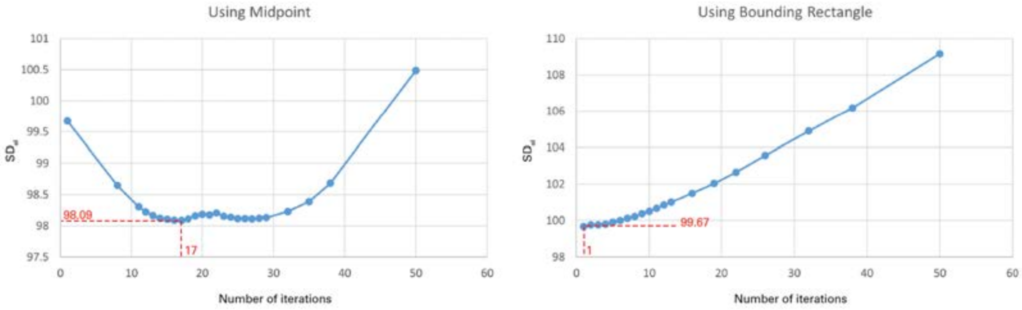


Figure 9.21: Comparison of the SD_{el} improvement. Left: Using the midpoint of the shortest edge as a feature during the learning process resulted in an effective improvement of the SD_{el}. Right: Using the bounding rectangle as a feature during the learning process did not improve the score.

Finding the right features thus requires domain knowledge and experimentation. The fact that illustrates that other features are possibly more relevant than the ones introduced above, is that the state space S is only partially explored (Fig. 9.22).

In other words, the learning agent never experienced a significant part of the possible system states. Therefore, features that add a higher “resolution” to the part of the state space that has been explored seems like a good strategy, without increasing the total number of possible states.

Another point is “reward engineering”, i.e. defining the positive or negative reward that an agent receives for a specific action, which also requires domain knowledge and experimentation.

Still, this experiment introduced the basic mechanisms that are necessary for autonomous learning of goal-oriented behaviors and demonstrated one way to implement such a learning process. More experiments and possibly alternative implementations will be investigated as part of future work. This includes

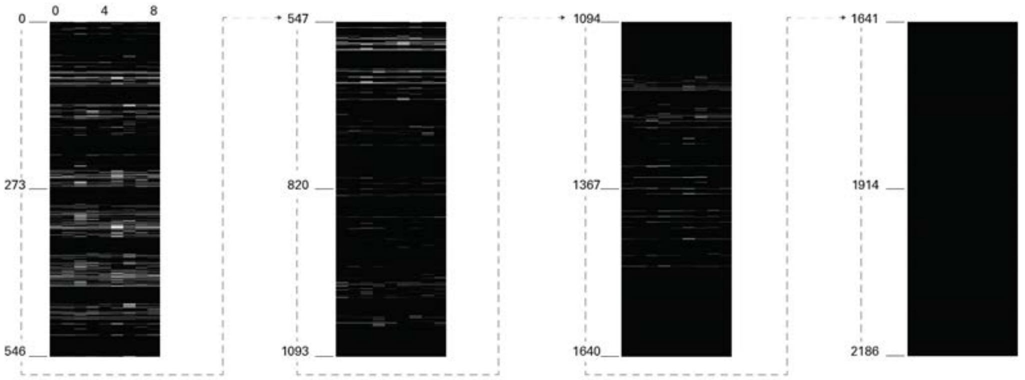


Figure 9.22: Plot of the Q-table.

1. extending the learning setup to include anticlastic surface scenarios,
2. exploring the “engineering” of features and rewards,
3. exploring other objective functions in addition to minimizing SD_{el}, such as minimizing the difference between the current plate size and a targeted plate size (see experiment 8),
4. adding the learning capability to the framework itself, and also
5. exploring a Deep RL approach, which would allow for the representation of continuous, i.e. floating point, features (while still requiring a curated selection of features).

9.6 Results and Discussion

This chapter introduced the last step in the proposed methodology for developing agent-based models of plate shells ranging from analysis and abstraction to transfer and implementation, with the objective to implement selected design principles as a proof-of-concept. The vehicle of this proof-of-concept was a problem-oriented case study focussing on a crucial aspect of existing approaches towards agent-based shell design: the need to first define a surface which hosts the agents that then segment the surface into planar plates. The aim of this case study thus was to obviate the need for such a designed surface and, in turn, to allow for a direct feedback between global form, plate configuration, and local rules. The chosen approach was to focus on the established duality between a triangular network representation and polygonal plate representation, where nodes and edges in the former represent plates and joints between plates in the latter.

Starting from a null model, each modeling step, termed “experiment”, introduced an additional aspect of modeling a plate system. Conceptualizing each modeling step as an experiment allowed making the aims and methods of each modeling step explicit. This needs to be seen in contrast to often implicit or tacit decisions in model development, not only in ABMS but in modeling in general. It thus allowed for comparing the results of each modeling step to previous steps and often provided the starting point and baseline for subsequent experiments.

The case study thereby not only demonstrated the incremental modeling process that is recommended for ABMs, and thereby the implementation of previously defined design principles, but it also presents an important contribution to the catalog of design methods available for plate shell design. This approach should thus be differentiated from the previously mentioned, which is reflected in the proposed terms “plate shell” vs. “segmented shell”, which does not imply a segmentation of an a priori defined surface and “3PI” vs. “TPI” in terms of the plate generation method.

The demonstration of a plate shell design method that obviates the need for a top-down designed surface, and rather lets global form emerge from local behaviors in a bottom-up way, can be considered one of the most important findings of this case study. While this has been demonstrated in experiment 3, the subsequent experiments resulted in additional important contributions to plate shell design, which include

1. the ability to explicitly define in which angle plates meet the ground,
2. to limit the extents of a plate to a user-defined maximum diameter that is based on the interaction network’s topology
3. edge-swapping to remedy possible self-intersections of plates,
4. parameter optimization in order to optimize certain global aspects of a plate shell; and finally
5. learning goal-oriented behaviors using the Q-Learning approach.

Particularly the fourth and fifth contributions are considered noteworthy. Parameter optimization allows finding optimal parameter combinations for global aspects, such as SD_{el}, that would be difficult or impossible to determine manually. The field of optimization provides many powerful algorithms that are able to handle complex and even unknown objective functions (black-box optimization) and even creates the peculiar situation where agent systems optimize agent systems such as in the example of using PSO for parameter optimization.

Learning behaviors, the last contribution, opens up a fundamentally different approach for designing agent behaviors as an alternative, and possibly more powerful, approach to hand-designing behaviors. Its application and implication not only reach beyond its demonstrated application in plate shell design, but it also addresses one of the main challenges that agent-based design is facing: how to design local behaviors that achieve a desired global effect, where even predicting the global effect is a challenge (see [205, p. 754]). While the results achieved in the corresponding experiments cannot be considered more than a proof-of-concept, this approach opens up a series of interesting and exciting questions for plate shell design, which will be pursued in further research.

10

Summary and Conclusion

10.1 Summary of Findings and Contributions

The main findings of the dissertation naturally fall into its two main research areas: 1) the investigation of a systematic approach for building conceptual models of segmented shells by 'distilling' design principles from real-world examples through case studies in part II; and 2) the implementation of design principles using ABMS as a proof-of-concept of the proposed methodology in part III. Additional novel insights were presented in part I as part of the review of related work and state-of-the-art in shell design and ABMS. This first section summarizes the findings of each of the three main parts of this dissertation, while discussing their contributions to the field of agent-based architectural design of segmented shells.

10.1.1 Part I: Shell Design and ABMS

The first part of the dissertation started with introducing the general problem, its relevance, and the research objectives (chapters 1 and 2). It then summarized the related work and state-of-the art in the fields of shell design and ABMS and situated the topic within the historical context of the two fields. In the field of ABMS (chapter 3), the work is part of research into a methodology for agent-based architectural design. The topic currently experiences renewed interest and relevance not only in the context of artificial intelligence research, but also in the context of developing tools for architectural design that allow addressing the challenges that the building industry is facing, for example, regarding embodied energy as part of the 'production challenge.' In shell design (chapter 4), it is part of ongoing research in discrete surface structures as an alternative to and further development of research in continuous surface structures, which peaked in the third quarter of the 20th century. A focus was placed on the question of segmentation and the relevance of the established concept of the so-called *plate-lattice dualism* for segmented shell design was introduced. In this context, ABMS is seen as a method that is able to simultaneously integrate multiple objectives that are subject to geometrical, structural, and fabricational constraints.

Given the objectives of this dissertation, the first finding (1), which is drawn from the review of the state of the art in shell design, is an unexpected one. It is the realization that 'form-optimized', discretized shells made of dry-stone have already been built in neolithic times purely based on empirical knowledge and material understanding and therefore long before descriptive geometry allowed designing and building the geometric shells known from Roman times, which typically are considered the ancestors of modern shells. These shells, which are stabilized by their dead-load, are in fact more akin to 20th century shells: from a methodological point of view to the empirically form-found shells of Heinz Ilser and from a geometric point of view to shells designed using numerical methods. Numerical methods are still the basis of contemporary form-optimization in shell design. Identifying these 'ancestors' of contemporary shells gains new relevance in view of recent work that explores robotic assembly of unbound aggregates, ranging in sizes from gravel to rocks, into architectural structures (see, for example, the work by Clifford and McGee[1] or Rusenova[2]).

The second contribution (2) in the first part of this dissertation is in the area of agent-based modeling and simulation. It is the proposed classification of agent systems based on the entity that agents in the model represent, be it *real* or *virtual* agents, in relation to the *real* world. This classification is not only compatible with previous classifications by Reynolds (1999), Snooks (2014), and Baharlou (2017), but it is also extensible, for example, in the direction of virtual agents of real world building systems, such as in the case of adaptive structures and building envelopes.

10.1.2 Part II: Building Valid Conceptual Models

The second part of the dissertation addressed the first explicitely stated research goal of developing a systematic approach for creating valid conceptual models of segmented shells based on pre-validated principles (see 2.3). This part started off with an introduction to the case study-based, inductive research approach, which aims at a generalization of case-specific findings. It first provided a definition of the terms 'functional principle' and 'design principle' and presented the shared context and background of the cases (chapter 5), followed by the case studies themselves.

---

[1] see Brandon Clifford and Wes McGee. Cyclopean Cannibalism: A Method for Recycling Rubble. In Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture, pages 404-413. 2018.

[2] see Gergana Rusenova. Material- and Fabrication-informed Design of Structurally-sound Jammed Architectural Structures. ETH Zürich, 2019

The case studies (chapters 6 and 7) constitute the core of this dissertation. Through analysis and increasing levels of abstraction functional principles were identified in the case data that represent the methodical steps and their connections (the chain of causality) in the design, fabrication and construction of the segmented timber shells under investigation (corresponding to research objective [i]). The studies revealed previously implicit networks of interactions and information flow through the networks, summarized in dependency graphs, which then allowed for identifying and distinguishing between feedback loops and concurrent developments between domains (corresponding to research objective [ii]). The studies also identified high-level functional principles that point beyond the specific cases and are relevant for the design of segmented shells in general.

Chapter 8 then presented the synthesis of the two case studies in terms of its transfer from functional to design principles, its definition of a catalog of design principles, its contextualization with regards to design patterns, and, ultimately, its proposed mapping from design principle to agent-based modeling construct (research objective [iii]). In particular, the last aspect showed how the problem of shell design can be conceptualized as an agent-based design problem.

The importance of the definition of design principles for any software implementation and the mapping from design principle to software construct, particularly in the context of ABMS, is that the identified design principles constitute validated conceptual models given that the cases from which these design principles were abstracted are validated plate shells. The design principles are thus understood to be axiomatic (the third category of principles next to methodic and systematic principles, see 5.3), meaning that they do not need to be proven again each time that they are intended to be used. Rather, they can be referred to “by name”, as it is customary in the case of *design patterns*, and be scrutinized by expert opinion.

Part II of this dissertation thus contributes to the field of shell design methodology in terms of (3) proposing a roadmap and toolset for abstracting, and transferring validated general design principles from the analysis of specific real-world cases. The proposed approach is intended to be applicable to other examples of segmented shells and can be used to expand the proposed catalog of design principles. Part II not only demonstrated the viability of the approach, but it also constitutes the basis for the proof-of-concept in part III. For the first time in the context of the design of segmented shells, a systematic approach is proposed and demonstrated for transferring functional principles, which are specific to a project, into general design principles. Such an approach is paramount for generalizing design knowledge from specific segmented shells: as part of an inductive approach towards developing design methods in particular, and for knowledge generation in general.

10.1.3 Part III: Implementing Conceptual Models

While the catalog of design principles per se is independent of a specific implementation, the third part focused on the *implementation* of selected design principles using ABMS as the final step in the proposed methodology for creating agent-based models of segmented shells (chapter 9). This part thus addressed the second stated research goal of defining, implementing, and testing a computational framework, in which the previously defined principles can be implemented while taking discipline-specific requirements into account (see 2.3).

The chapter first introduced the state-of-the-art in ABMS methodology and then summarized aim, scope and structure of ICD's ABM Framework, which provided the platform for implementing a proof-of-concept. The development of the framework, which was done collaboratively, corresponds to research objective [iv]. Objective [v] is met through the modular structure of the framework, which ensures extensibility through application-specific libraries. This is demonstrated by the different application scenarios in which the framework is currently used, which include the design of CLT structures, adaptive structures, design of formwork for modular concrete construction, and plate structures (research objective [vi]). In these application scenarios, the framework is embedded in interdisciplinary contexts that require specific interfaces between architectural design and other disciplines. The category of so-called 'expert libraries' ensures that those interfaces can be systematically addressed and extended through additional domain-specific interfaces to, for example, LCA, structural analysis, or redundancy matrices as in the case of adaptive structures.

The vehicle for the proof-of-concept (research objective [vii]) was a problem-oriented case study focussing on the solution of a specific problem: the agent-based modeling of a plate shell that is not a segmentation of a previously designed reference surface, as was the case in previous work. Instead, the model is the result of the 'free' positioning and orienting of plate agents in Euclidean space $\mathbb{E}^3$ utilizing the previously introduced concept of *plate-lattice dualism*. The study was structured as a series of experiments with increasing complexity that not only addressed previously identified challenges in agent-based modeling of segmented shells, such as the need for a 'hand-designed' proxy surface, but also presented new findings with respect to parameter optimization of ABMs of segmented shells and autonomous learning of agent behaviors using Reinforcement Learning. The incremental modeling approach of the case study, together with the transparent description of the individual experiments was driven by research objective [viii].

Part III thus contributes to the field of architectural shell design (4) a novel approach for plate shell design that obviates the need for a reference surface, which allows for direct feedback between global form and local rules. This is considered one of the most important findings of this case study. The fifth and sixth contributions are to the field of agent-based architectural design: by (5) demonstrating parameter optimization as a means to meet specific design objectives that would be difficult or impossible to meet manually; and by (6) presenting a functional prototype of learning goal-oriented behaviors for plate agents. This last finding opens up a fundamentally different approach for designing agent behaviors as an alternative, and potentially more powerful, approach to conventional design of behaviors, be it normative or based on heuristics. Its application and implication point beyond its demonstrated application in plate shell design towards all computational modeling that seeks to harness behavioral intelligence for achieving specific goals, where the exact rules or their combination are, however, unknown.

An further important finding is that the proposed methodology for agent-based architectural design of segmented shells is compatible with the methodology that is generally recommended in literature for developing ABMs: 1) *preliminary research*, 2) *informal modeling*, 3) *formal modeling*, 4) *implementation*, and 5) *validation* [132, pp. 22-27]. Specifically, analysis and abstraction correspond to preliminary research and informal modeling, and transfer and implementation correspond to formal modeling and implementation. The point of *validation* is considered at two stages in the development process: first, with respect to *conceptual modeling* through the definition of design principles from real-world segmented shells and the ability to subject these principles to expert review. Second, validation is considered with respect to the *implementation* of ABMs through the use of an open-source ABM framework and through platform-independent documentation of ABMs. This last aspect hints at further research.

10.1.4 Summary of Contributions

In summary, the presented research makes the following contributions to the fields of shell design history, ABMS, shell design methodology, architectural shell design, and agent-based architectural design:

1. The insight that ‘form-optimized’, discretized shells made of dry-stone have already been built in neolithic times long before the geometric shells from Roman times, which typically are considered the ancestors of modern shells.
2. The proposed classification of agent systems based on the entity that agents in the model represent, which synthesizes previous classifications and is extensible towards novel conceptualizations of agent systems.
3. A systematic approach for abstracting, and transferring validated functional principles from specific real-world cases to general design principles for seg-

mented shells.

4. A novel approach for plate shell design that obviates the need for a reference surface, which allows for direct feedback between global form and local rules.
5. The demonstration of parameter optimization as a means to meet specific design objectives that would be difficult or impossible to meet by manually tuning parameters.
6. The demonstration of a functional prototype for learning goal-oriented behaviors for plate agents using Reinforcement Learning.

10.2 Further Work and Research

Next to presenting concrete findings, the research in parts II and III of this dissertation also points towards the need for further research, which is outlined in this section.

10.2.0.1 Agent-based Architectural Design Methodology

In the area of agent-based architectural design methodology, more research should go into architecture-specific documentation of ABMs and interoperability. While toolkits facilitate the ambition to make ABMS results comparable, another important goal is to standardize documentation of ABMs in order to transparently communicate findings produced by ABMS. In this regard, Grimm et al.'s Overview, Design Concepts, and Details (ODD) protocol [70], which is used in many fields and very recently has been updated [71], could be a template for a standardized way of model documentation. Further work will therefore go into documenting the presented agent-based design model for plate shells using or adapting an established protocol such as ODD in order to provide a platform-independent description of the ABM. In this way, the model can not only be replicated but also form the basis for further research in the area of plate shell design—even for research pursuing different priorities.

A promising area of further research in agent-based systems is Reinforcement Learning, particularly in the area of agents as 'real' builders simulating physical machinery, which is being investigated at the time of writing in the Cluster of Excellence IntCDC.

Another aspect in this area would be a study of interdisciplinary collaboration and the embedment of the ABM Framework in an interdisciplinary planning context where high-level agents would represent stakeholders in the planning process. This would require the development of an *ontology* for shell design from the 'instances' of built shells, and specifically from the case-specific knowledge representations in the form of functional principles and dependency graphs. As part of a hierarchical model, where agents would both wrap lower-level agent systems and other computational models, ABMS would constitute the integration technology (as originally proposed by Jennings [82, p. 288]).

10.2.0.2 Shell Design Methodology

In the area of shell design methodology, a need for further research is seen in expanding and consolidating the proposed catalog of design principles through the study of additional cases of segmented shells, such as BUGA Wood Pavilion, using the proposed roadmap and toolset. Similarly, an important aspect in this area is a standard way for documenting design principles and a continued discussion of the relevance and mechanisms of specific principles.

Furthermore, the concept of *plate-lattice dualism*, especially the ability to evaluate plate shells by analyzing their dual lattice representation, which is exponentially faster, opens up the possibility for investigating near realtime preliminary structural analysis as part of an agent-based design approach. In this way, the structural performance of a shell would not anymore have to be solely conceived of as the global outcome of the interactions at the local level and evaluated at the global level after convergence, but the actions at the local level could also be structurally evaluated at runtime and, consequently, structural behaviors could be developed that are normative instead of based on heuristics. This also hints at a the possibility of learning structurally informed behaviors using the learning mechanism outlined in chapter 9.

Furthermore, the agent-based implementation of design principles and, in particular, the novel approach towards plate shell design presented in Chapter 9 naturally asks for a physical demonstration in the form real-world plate shell.

10.2.0.3 Teaching ABMS

Finally, the formalization of the agent-based design approach in ICD's ABM Framework has already shown to facilitate the systematic teaching of agent-based modeling in architecture and, by extension, the integration of learning mechanisms in agent models. Noteworthy preliminary results have been achieved as part of ITECH courses with respect to modeling distributed robotic assembly as part of the project "Agent-based Robotic Builders" by Kalousdian and Locknicki (2019) (see C.1); with respect to informing agent behaviors with structural rules as part of the project "Discrete Morphogenesis" by Locatelli and Chen (2020) (see C.2); with respect to evolutionary learning of behaviors using dynamic evolution of behavior parameters as part of the project "Drone Builders" and direct neural evolution of behaviors as part of the project "Neuroevolution for Robotics in Architecture" both by Sahin, Skoury, and Treml (2020) (see C.3 and C.4). Systematic teaching of ABMS is expected to increase competence among graduates, through which ABMS will ultimately trickle into architectural design practice; and with it the competence to support decision making processes where it has the highest impact with regard to resoure effectiveness: in the early design stages.

10.3 Final Considerations

10.3.1 On Agent-based Architectural Design

As previously stated, two aspects of ABMS are considered to be particularly relevant in architectural design: on the one hand, its capacity to discover novelty by harnessing emergence of complex systems; on the other, its capacity to converge to 'optimal' dynamic or static equilibria, by harnessing Swarm Intelligence, i.e., the ability of many to solve problems that are beyond the scope and capacity of the individuals in the swarm.

At one end of the spectrum, agent models might therefore be built as part of an explorative approach in order to explore the dynamics of a system of interacting agents or to test hypotheses about the micro-level behaviors that lead to macro-level system behaviors. In this case models are generators. At the other end of the spectrum, agent systems are used for goal-oriented applications that focus on convergence to an optimal end state that minimizes an objective function by using, for example, reinforcement learning, parameter optimization or a particle swarm approach. In this last approach solutions to the optimization problem are particles in a fitness landscape that is parsed by utilizing flocking behaviors (see Section 3.2.8). Such an application would be an example of second order agent systems where one abstract agent system optimizes parameters of a second agent system that is the representation of the design.

In practice, the focus will very often be on a hybrid approach merging exploration and exploitation, where the dynamics of agent systems are exploited in order to converge to different steady states, not primarily focusing on optimization but rather on the ability of complex systems to adapt and self-organize, taking into account changing 'environmental' conditions and user input. Agent-based architectural design thus embodies the divergent-convergent design approach that is characteristic for architectural design, which oscillates between increasing variation, as part of an explorative search process, and selection, as part of homing in on the design solution (see 2.1).

Agent-based architectural design thus still resembles a 'steering' process, however not anymore only in the sense described by Kilian [89], where a single agent, the designer, steers a parametric system to a desired outcome through "steering principles" and by manipulating system parameters; but now also in sense described by Kelly [88, p. 23], where the designer is more aking to a "shepard" who drives a "herd" of design agents "by applying force at crucial leverage points."

10.3.2 On Evaluating Agent-based Modeling

One question that might be arising from the discussion of the importance of both validated conceptual models and the validation of their implementation, is if ABMs designed and implemented in the proposed way really are 'better' than ABMs that are produced in an ad-hoc, 'black-box' way as it is currently the case across the architectural design community. It would seem that models that are built from pre-validated design principles perform better than those that are constructed from a single designer's intuition. While such a hypothesis could be confirmed by drawing from multiple examples in literature and might therefore be generally considered to be true, it cannot be proven, because exceptions might be possible: that is, cases where ad-hoc purpose-built, possibly black-box, implementations outperform validated implementations and produce valid results with regards to the system that they represent.

This problem is related to the open question to what extent an ABM actually is an explanation of the observed phenomenon. According to Metz [132, p. 18], these considerations are known as *generative explanations* in the field of CAS. The general agreement is that models that produce an expected outcome are sufficient as an explanation, in that they not only show that the system produces the outcome, but also how it is achieved. A sufficient condition, however, does not exclude other conditions, such that every generative explanation is considered to be one among potentially many [132, p. 18]. In other words, there are multiple models and modeling paradigms that theoretically can produce the same result including implementations that converge faster than one that is conceptually validated and implemented in an open ABMS software framework.

Therefore, instead of trying to prove the above hypothesis, a more relevant question would be to determine circumstances, in which such an exception might be true. For this purpose, defining evaluation criteria would be of particular importance, since the dis-/confirmation of the hypothesis might be subject to weighting evaluation criteria differently, such as convergence speed, development time, or the necessity of transparency and compatibility, based on the application goal. An objective of further research might therefore be to define and discuss a list of evaluation criteria, which, when weighted on a case-by-case basis, allow a designer to determine at an early stage, which modeling strategy to pursue.

10.4 Conclusion

From a methodological point of view, the relevance of the presented research lies in identifying and abstracting functional principles, and translating them to design principles that can be implemented in software and applied to other designs of segmented shells. This allows activating the identified design principles as part of exploring novel designs for new segmented shells. As opposed to starting each new design from scratch and questioning the validity of the assumptions underlying a design model, a library of design principles can form the basis of a new development.

This is not a far-fetched proposition, since every designer already uses implicit or tacit design principles, rules-of-thumbs, and heuristics besides the rules established in building regulations. Making implicit principles explicit allows for reproducibility, review, transfer, and dissemination in the form of shareable knowledge, which can be built inductively from case studies. Since the validity of the design principles can be confirmed through realization and evaluation in the form of research pavilions and demonstrators, the expansion of the library of design principles is therefore predicated on further case studies and the realization of further research buildings.

The prospect of this research and its proposed methodology for developing ABMs of segmented shells, starting from the analysis of built examples, through abstraction, transfer, and implementation is ultimately not limited to the research area of segmented shells, but can be applied to developing conceptual models for other architectural design tasks such as designing building envelopes, load-bearing systems, or urban design: whenever multiple entities interact in a rule-based way, including systems where entities adaptively change the rules (behaviors).

The dissertation confirmed the cross-disciplinary applicability of concepts and approaches that are established in the modeling of, for example, ecosystems, social animals, or robot swarms to the field of architectural design of segmented shells. It thereby also highlights the need for agent-based architectural design to look outside its own domain in order to avoid challenges, such as transparency and reproducibility, that have effectively been solved already. At the same time, the work also shows how agent-based architectural design can contribute to the field of ABMS by extending the concept of agents to building elements that are inherently rule-based considering, e.g., rules defined by producibility, structure, or building code. While having no intention in the ‘real’ world (which would otherwise be one of the essential requirements in the definition of the term agent), building elements can be conceptualized as having intention in the ‘virtual’ world.

Finally, the characterization of building elements as software-defined objects with embedded, possibly adaptive, rules opens up promising synergies with another established modeling approach in architectural design: building information modeling (BIM). In BIM building elements are similarly defined as heterogeneous software objects. From the point of view of this dissertation, imbuing these objects with (adaptive) rules in order to harness self-organization appears to be the next logical step in BIM. Agent-based modeling would be the candidate method.

Backmatter

Glossary

Adaptation – The process by which organisms (agents) change their behavior or by which populations of agents change their collective behaviors with respect to their environment [120].

Agent-based modelling – A modeling and simulation approach that can be applied to model complex, or complex adaptive system, in which the model is comprised of a large number of interacting elements (agents) (ibid.).

Agent – Computational entity that is self-contained and identifiable, autonomous and goal-directed; it has a state that is defined by static and dynamic attributes; it has behaviors that map sensed information to decisions and actions; and it has dynamic interactions with other agents and the environment.

Artificial Life – A discipline that studies 'natural' life by attempting to recreate biological phenomena from scratch within computers and other 'artificial' media. Rather than studying biological phenomena by taking apart living organisms, the aim is to assemble systems that behave like living organisms [110].

Behavior – The internally coordinated responses (actions or inactions) of whole entities (individuals or groups) to internal and/or external stimuli [113].

Catenary – The shape of a hanging chain or rope under self-weight (constant load per unit arclength) is a particular case of a funicular curve. For comparison, the funicular curve for constant load per horizontal length is a parabola, as in the case of a suspension bridge [210, p. 28].

Cellular Automata (CA) – A mathematical construct and technique that models a system in discrete time and discrete space, in which the state of a cell depends on transition rules and the states of neighboring cells [120].

Complex Adaptive System (CAS) – A system comprised of a large number of strongly interacting components (agents) that adapt at the individual (agent) level and/or collectively at the population level (ibid.).

Dead load – The permanent load that the shell has to resist, which usually is self-weight.

Downward causation – The process by which agents change their behavior at the micro-level in response to the emergent system behavior at the macro-level (ibid.) (see also [47]).

Emergence – The process by which coherent 'emergents', that is, properties, behaviors, structures, or patterns, dynamically arise at the macro-level of a system from the interactions between the parts at the micro-level [47]. Alternatively, "the process by which order is produced in nature" [120].

Feedback – The outputs of a system at time t affect the inputs of that system at time $t + 1$ [38, p. 10].

Funicular shape – The shape of a hanging chain or string that is loaded with any number or weights and corresponds to an admissible equilibrium state of a planar arch for the same loads [2], [21].

Force density – The ratio of force over (stressed) length in a bar or cable segment. It is also known as the 'tension coefficient' [117, p. 68].

Form-active – Describes structures that adjust their form based on changing load conditions including tensile surface structures such as tents and cable nets. The shape of the structure mirrors the tensile forces acting in it [140, p. 15].

Form finding – The forward process in which parameters are explicitly/directly controlled to find an 'optimal' geometry of a structure, which is in static equilibrium with a design loading [4, p. 2].

Form-passive – Describes structures that are rigid and do not significantly adjust shape to external loading [210, p. 22], which includes surface structures acting mostly in compression. Distinguishes shell structures from tensile surface structures [140, p. 15].

Graphic statics – A graphical method for finding the funicular load for a given shape of cable or funicular shape for a given load [210, p. 28].

Heterogeneity – Heterogeneity refers to a "differentiation in terms of the agent specifications themselves; that is, a difference in their rules and/or their set of attributes" as opposed to merely a differentiation in attribute values [37, p. 22].

Homomorphism – The abstraction of the part of the world that is being represented in the model also called the "many-to-one mapping from states of the world to states of the model (the equivalence classes)" [57, p. 60].

Locomotive agent – An agent capable of locomotion, the type of agent predominantly used in this investigation.

Membrane stress – Tensile and compressive stress in the plane of the shell surface; corresponds to the axial stress in an arch [210, p. 23].

Nonlinear - The property of a system where the change of system output is not proportional to the change of an input variable [19, Sec. 1.2.2].

Self-organization – The dynamical and adaptive process where systems acquire and maintain structures themselves, without external control [47].

Structural optimization – The inverse process to form-finding, in which parameters are implicitly/indirectly optimized to find the geometry of a structure such that an objective function or fitness criterion is minimized [4, p. 3].

Surface shells – the category of shells where the load-bearing layer is the spatial enclosure which is predominantly closed, as, for example, in the case of a colander or a cereal bowl [210, p. 25].

Swarm intelligence – Collective intelligence emerging from the actions of interacting agents behaving according to a set of prescribed simple rules [120, p. 115].

Update rule – A rule or transformation directive for changing or updating the state of an agent in an agent-based model (ibid.).

A

Credit lists

A.1 ICD/ITKE Research Pavilion 2011

Project Team

Institute for Computational Design and Construction:
Prof. Achim Menges, Steffen Reichert, Tobias Schwinn
Institute of Building Structures and Structural Design:
Prof. Jan Knippers, Markus Gabler, Riccardo La Magna, Frédéric Waimer
Competence Network Biomimetics Baden-Württemberg

Concept and project development

Oliver David Krieg, Boyan Mihaylov

Planning and realisation

Peter Brachat, Benjamin Busch, Solmaz Fahimian, Christin Gegenheimer, Nicola Haberbosch, Elias Kästle, Oliver David Krieg, Yong Sung Kwon, Boyan Mihaylov, Hongmei Zhai.

Project Support

Main supporters

KUKA Roboter GmbH, Ochs GmbH.

Additional supporters

KST2 Systemtechnik GmbH, Landesbetrieb Forst Baden-Württemberg (ForstBW), Stiftungen LBBW, Leitz GmbH & Co. KG, MüllerBlaustein Holzbau GmbH, Hermann Rothfuss Bauunternehmung GmbH & Co., Ullrich & Schön GmbH, Holzhandlung Wider GmbH & Co. KG.

A.2 Landesgartenschau Exhibition Hall

Project Team

Institute for Computational Design:
Prof. Achim Menges, Tobias Schwinn, Oliver David Krieg
Institute of Building Structures and Structural Design:
Prof. Jan Knippers, Jian-Min Li
Institute Engineering Geodesy:
Prof. Prof. Volker Schwieger, Annette Schmitt
Müllerblaustein Holzbau GmbH:
Reinhold Müller, Benjamin Eisele
KUKA Roboter GmbH:
Alois Buchstab, Frank Zimmermann
Landesbetrieb Forst Baden-Württemberg:
Sebastian Schreiber, Frauke Brieger
Landesgartenschau Schwäbisch Gmünd 2014 GmbH:
Karl-Eugen Ebertshäuser, Sabine Rieger

Scientific development

University of Stuttgart:
Oliver D. Krieg, Jian-Min Li, Annette Schmitt, and Tobias Schwinn.
Müllerblaustein: Benjamin Eisele

Project Support

Project Funding

Clusterinitiative Forst und Holz Baden Württemberg, EFRE European Union, KUKA Roboter GmbH, Landesbetrieb Forst Baden-Württemberg, Landesgartenschau Schwäbisch Gmünd 2014 GmbH, müllerblaustein Bauwerke GmbH

Additional supporters

Adler Deutschland GmbH, Autodesk GmbH, Carlisle Construction Materials GmbH, Fagus Stiftung, Gutex H. Henselmann GmbH & Co. KG, Hess & Co. AG, MPA – Materials Testing Institute (University of Stuttgart), Leitz GmbH & Co. KG, Spax International GmbH & Co. KG

B

Software Specifications

B.1 Computer-Aided Geometric Design

Rhinoceros 3d

NURBS-based geometric design.

- Developer: Robert McNeel & Associates
- Version number: Version 6 SR29, 2020-08-25
- Available at: https://www.rhino3d.com/

Grasshopper

Visual parametric programming.

- Developer: McNeel Associates
- Version number: Build 1.0.0007, 2020-08-25
- Included with Rhino 6

Opossum

Optimization add-on to Grasshopper.

- Developer: Thomas Wortmann
- Version number: 2.1.0, 2020-05-25
- Available at: https://www.food4rhino.com/app/opossum-optimization-solver-surrogate-models

Owl

Machine Learning add-on to Grasshopper.

- Developer: Mateusz Zwierzycki
- Version number: 1.0.0.0, 2019-01-31
- Available at: https://www.food4rhino.com/app/owl

ICD ABM Framework

Agent-based Modeling and Simulation add-on to Grasshopper

- Developer: Institute for Computational Design and Construction, University of Stuttgart
- Version number: 1.0.3.1, 2020-07-01
- Available at: https://github.tik.uni-stuttgart.de/icd/ICD.AbmFramework/releases

B.2 Integrated Development Environment

Visual Studio

- Developer: Microsoft Corporation
- Version number: 16.6.3
- Available at: https://visualstudio.microsoft.com/

C

Student Projects

C.1 Agent-based Robotic Builders

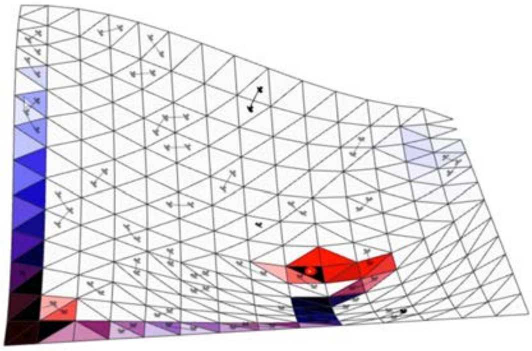


Figure C.1: Agent-based Robotic Builders (Kalousdian and Lochnicki, 2019).

Project credits – *Students*: Grzegorz Lochnicki and Nicolas Kubail Kalousdian | *Tutors*: Tobias Schwinn, Long Nguyen, Katja Rinderspacher, Maria Yablonina, Prof. Achim Menges | *Project type*: Seminar project | *Seminar*: Computational Design and Simulation “Behavioral Fabrication” | *Institution*: Institute for Computational Design and Construction, University of Stuttgart, Germany | *Year*: 2019

Project description[1] – The project is an agent-based simulation of a system for distributed robotic autonomous assembly. The project is inspired by, first, ants and their pheromone logic, with which ants mark their routes from nest to any resource they find. This behavior helps them get back to the nest and informs other ants about found resource. Second, the PhD Proposal by Benjamin Jenett called Relative Robotic Assembly of Discrete Cellular Structures. In this paper Jennet gives an

---

[1] The original description by the students has been edited for brevity and clarity

overview of relevant systems, how they work and what constraints exist.

The system includes BuilderAgent, CustomAgentBehaviors, BuilderMeshEnvironment, and BuilderAgentSystem classes, Solver, and custom visualization components. As a foundation, the provided ICD ABM Framework was used. All the custom classes inherit from the base classes in the framework. The code is encapsulated in a C# plugin for Grasshopper as an add-on the the Framework's core. Each BuilderAgent has a very limited perception of the overall system but performs complex tasks. There is no precomputed assembly sequence, and agents make decisions based on their closest surroundings.

The result of this assignment is working robotic assembly simulation. The pheromone distribution is tracked in real-time as a color on the mesh structure. The system performs better with a higher amount of agents (longer trails, more frequent interaction). Sometimes agents can stuck in their own trails in a manner observed in behavior of fire ants called the circle of death, but pheromone decaying allows to unlock this situation in time.

C.2 Discrete Morphogenesis

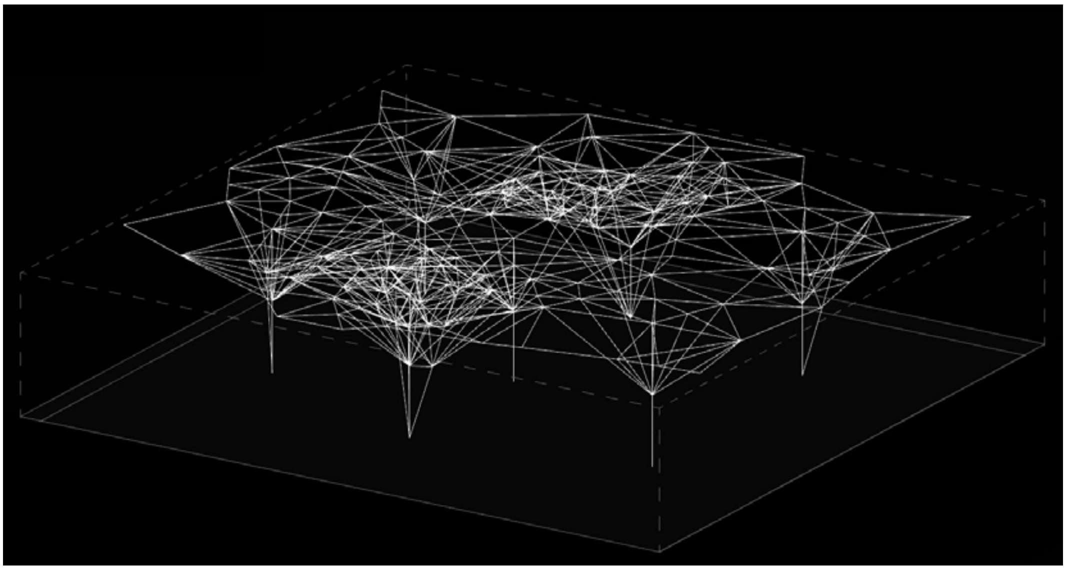


Figure C.2: Discrete Morphogenesis. Autonomous arrangement of beams based on structural rules (Locatelli and Chen, 2020).

Project credits – *Students*: Daniel Locatelli and Tzu-Ying Chen | *Tutors*: Tobias Schwinn, Oliver Bucklin, Samuel Leder, Mathias Maierhofer, Long Nguyen, Yasaman Tahouni, Prof. Achim Menges | *Project type*: Seminar project | *Seminar*: Computational Design and Digital Fabrication | *Institution*: Institute for Computational Design and Construction, University of Stuttgart, Germany | *Year*: 2020

Project description[2] – The aim was to take advantage of the open-ended nature of agent-based systems to create a computational morphogenetic method that integrates the discrete and anisotropic nature of timber assisted iteratively by finite element analysis.

To create the agent-based model, we used the ICD ABM Framework. The Cartesian Agent System was chosen for the straightforward static positioning as well as adding and removing of the agents. The structure is defined by connecting the agents through a Delaunay triangulation. In order to keep the elements within a realistic size, those that are larger than a defined threshold are filtered out. Karamba 3D was embedded in the ABM to produce the finite element model.

The structure stabilizes itself through iterative progress, which terminates when the maximum bending moment happening within the beams falls under a threshold. Benchmarking this outcome with a regular truss system, it would take three times as much the column thickness to reach the same performance as the generated result.

C.3 Drone Builders

Figure C.3: Structure built by Drone Builders (Sahin, Skoury, and Treml, 2020).

Project credits – *Students*: Ekin Sahin, Lior Skoury, and Simon Treml | *Tutors*: Tobias Schwinn, Oliver Bucklin, Samuel Leder, Mathias Maierhofer, Long Nguyen, Yasaman Tahouni, Prof. Achim Menges | P*roject type*: Seminar project | *Seminar*: Computational Design and Digital Fabrication | *Institution*: Institute for

---

[2] The original description by the students has been edited for brevity and clarity

Computational Design and Construction, University of Stuttgart | *Year*: 2020

Project description[3] – The chief drivers of the conventional construction industry are the labor of workers and the power of huge machines. While constructions by multiple collaborative agents are observed in nature, those principles and their potential in building environments still remain to be explored. The Drone Builders project aims to extend the limited understanding of the construction with humans and machines, to a computationally-driven multi-agent construction by utilizing the principles of collaboration and the genetic algorithms. One of the inspirations of the project was the Autonomous Grasping and Transport project developed by the researchers from the University of Pennsylvania's GRASP Lab. By integrating microsystems, the researchers were able to construct drones which are sensitive to payloads and thereby, can construct collaboratively.

The project was initiated by constructing a new agent system using Visual Studio. Setting up the basic rules for drones such as lift, wander, seek, containment, flee (avoid buildings, avoid hives), separation and build, enabled the agents to carry the building materials (picks) in the environment and form new structures without colliding with the existing obstacles and each other. The initial approach was then improved by developing a genetic algorithm which improves the performance of the drones similar to the tenets behind the biological evolution. This approach improves not only the agents by considering their current onditions and properties, but also the system in efficiency.

C.4 Neuroevolution for Robotics in Architecture

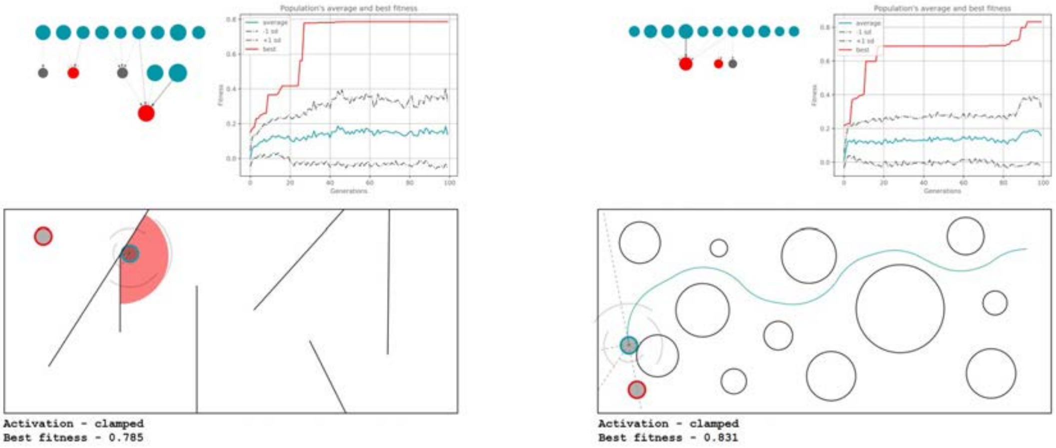


Figure C.4: Neuroevolution of Behaviors (Sahin, Skoury, and Treml, 2020).

Project credits – *Students*: Ekin Sahin, Lior Skoury, and Simon Treml | *Tutors*: Tobias Schwinn, Oliver Bucklin, Samuel Leder, Mathias Maierhofer, Long

---

[3] The original description by the students has been edited for brevity and clarity

Nguyen, Yasaman Tahouni, Prof. Achim Menges | *Project type*: Seminar project | *Seminar*: Computational Design and Digital Fabrication | *Institution*: Institute for Computational Design and Construction, University of Stuttgart | *Year*: 2020

Project description[4] – During the mid 1990s, complex reinforcement learning problems started to be solved by the artificial evolution of neural networks using genetic algorithms. With the aim of specifying the desired control behavior, the developed method called neuroevolution finds the right topology and connection weights.

The algorithm that was chosen in the project is NeuroEvolution of Augmenting Topologies (NEAT), an evolutionary algorithm which creates artificial neural networks. The NEAT algorithm starts with a minimal amount of nodes and connections and gradually evolves the complexity if necessary. NEAT uses speciation by which the algorithm clusters the population into several species depending on the similarity of connections and topology. Utilizing speciation and minimal structure proves the NEAT algorithm to be efficient and high-performing in solving challenging benchmark reinforcement learning tasks.

The autonomous steering experiment was benchmarked using two cases. In the first case, the autonomous robot was trained in the same environment using different activation functions named tanh, sigmoid, gauss and clamped and their fitness values were compared. After the initial training, it was tested on an unseen environment. While the autonomous robot with the clamped activation function showcased a promising result.

---

[4] The original description by the students has been edited for brevity and clarity

Bibliography

1. Abel, C. Urban Chaos or Self Organization. *Architectural Design* **39** (1969).
2. Addis, B. *Physical modelling and form finding* in *Shell Structures for Architecture: Form Finding and Optimization* (eds Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C.) 33–44 (Routledge, London and New York, 2014). ISBN: 9781315849270.
3. Adriaenssens, S., Barnes, M., Harris, R. & Williams, C. *Dynamic Relaxation: Design of a strained timber gridshell* in *Shell Structures for Architecture: Form Finding and Optimization* (eds Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C.) 89–102 (Routledge, London and New York, 2014). ISBN: 9781315849270.
4. Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C. *Introduction* in *Shell Structures for Architecture: Form Finding and Optimization* (eds Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C.) 1–4 (Routledge, London and New York, 2014). ISBN: 9781315849270.
5. Alexander, C., Ishikawa, S. & Silverstein, M. *A Pattern Language* in *A Pattern Language : Towns, Buildings, Construction* ix–xvii (Oxford University Press, New York, 1977). ISBN: 978-0-19-501919-3.
6. Alpa, G. General Principles of Law. *Annual Survey of International & Comparative Law* **1,** 1–37 (1994).
7. Alvarez, M. E. *et al. The buga wood pavilion* in *Proceedings of 39th Annual Conference of ACADIA 2019* (Austin, TX, 2019), 2–11.
8. Anumba, C., Ugwu, O., Newnham, L. & Thorpe, A. Collaborative design of structures using intelligent agents. *Automation in Construction* **11,** 89–103. ISSN: 0926-5805. https://doi.org/10.1016/S0926-5805(01)00055-3 (January 2002).
9. Bagger, A. *Plate shell structures of glass Studies leading to guidelines for structural design* Dissertation (Technical University of Denmark, 2010). ISBN: 9788778773005.
10. Baharlou, E. *Generative agent-based architectural design computation: behavioral strategies for integrating material, fabrication and construction characteristics in design processes* English. OCLC: 1032691839. PhD thesis (University of Stuttgart, Stuttgart, 2017). ISBN: 978-3-9819457-0-6. http://dx.doi.org/10.18419/opus-9752 (2019).
11. Baharlou, E. & Menges, A. *Behavioural prototyping: an approach to agent-based computational design driven by fabrication characteristics and material constraints* in *Rethinking Prototyping, Proceedings of the Design Modelling Symposium Berlin 2013* (eds Gengnagel, C., Kilian, A., Nembrini, J. & Scheurer, F.) (Verlag der Universität der Künste Berlin, Berlin, 2013), 291–303. ISBN: 9783844268454.

12. Ball, P. *The March of Reason* in *Critical Mass: How One Thing Leads to Another* 145–192 (2004).

13. Balz, M. *Umfassende, mutige Bautätigkeiten* tech. rep. (Aichtal, 2004), 31–32.

14. Bechert, S. *et al. Textile Fabrication Techniques for Timber Shells* in *Advances in Architectural Geometry 2016* (eds Adriaenssens, S., Gramazio, F., Kohler, M., Menges, A. & Pauly, M.) 154–169 (vdf Hochschulverlag AG an der ETH Zürich, Zurich, 2016). ISBN: 978-3-7281-3778-4.

15. Bechthold, M. The Return of the Future: A Second Go at Robotic Construction. *Architectural Design* **80,** 116–121. ISSN: 00038504. https://doi.org/10.1002/ad.1115 (July 2010).

16. Beetz, J., Van Leeuwen, J. & De Vries, B. *Towards a Multi Agent System for the Support of Collaborative Design - Assembling a toolbox for the creation of a proof of concept* in *Developments in Design & Decision Support Systems in Architecture and Urban Planning* (eds Van Leeuwen, J. & Timmermanns) (Eindhoven University of Technology, NL, 2004), 269–280.

17. Behdani, B. *Evaluation of Paradigms for Modeling Supply Chains As Complex Socio-technical Systems* in *Proceedings of the 2012 Winter Simulation Conference* (Winter Simulation Conference, Berlin, Germany, 2012), 413:1–413:15.

18. Bhooshan, S., Veenendaal, D. & Block, P. *Particle-spring systems: Design of a cantilevering concrete shell* in *Shell Structures for Architecture: Form Finding and Optimization* (eds Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C.) 103–113 (Routledge, London and New York, 2014). ISBN: 9781315849270.

19. Bishop, R. *Chaos* in *The Stanford Encyclopedia of Philosophy* (ed Zalta, E. N.) Spring 2017 (Metaphysics Research Lab, Stanford University, 2017). https://plato.stanford.edu/archives/spr2017/entries/chaos/.

20. Bletzinger, K.-U. & Ramm, E. *Computational form finding and optimization* in *Shell Structures for Architecture: Form Finding and Optimization* (eds Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C.) 45–55 (Routledge, London and New York, 2014). ISBN: 9781315849270.

21. Block, P., Lachauer, L. & Rippmann, M. *Thrust Network Analysis: Design of a cut-stone masonry vault* in *Shell Structures for Architecture: Form Finding and Optimization* (eds Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C.) 71–87 (Routledge, London and New York, 2014). ISBN: 9781315849270.

22. Bogenstätter, U. Prediction and optimization of life-cycle costs in early design. *Building Research & Information* **28,** 376–386. ISSN: 0961-3218. https://doi.org/10.1080/096132100418528 (September 2000).

23. Bonabeau, E. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences* **99,** 7280–7287. https://doi.org/10.1073/pnas.082080899 (May 2002).

24. Bopp, R. *Design for X-Methoden* in *Forschungs- und Entwicklungsmanagement: Simultaneous Engineering, Projektmanagement, Produktplanung, Rapid Product Development* (eds Bullinger, H.-J. & Warschat, J.) 195–203 (Vieweg+Teubner Verlag, Wiesbaden, 1997). ISBN: 978-3-663-05946-2. https://doi.org/10.1007/978-3-663-05946-2_10.

25. Bosfa, P. L. *Park Oceanografic* 2019. https://bosfa.com/case-study/park-oceanographic/ (2019).

26. Breitenberger, M., Bletzinger, K.-U. & Wüchner, R. *Isogeometric Layout Optimization of Shell Structures Using Trimmed NURBS Surfaces* in *10th World Congress on Structural and Multidisciplinary Optimization* (Orlando, Florida, USA, 2013).

27. Brell-Çokcan, S. & Braumann, J. *A New Parametric Design Tool for Robot Milling* in *ACADIA 10: LIFE in:formation [Proceedings of the 30th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)]* (2010), 357–363.

28. Brell-Çokcan, S., Reis, M., Schmiedhofer, H. & Braumann, J. *Digital Design to Digital Production* in *Computation: The New Realm of Architectural Design – Proceedings of the 27th eCAADe Conference* (Istanbul, 2009), 323–329.

29. Brooks, R. A. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation* **2,** 14–23. ISSN: 0882-4967. https://doi.org/10.1109/JRA.1986.1087032 (1986).

30. Brooks, R. A. Elephants Don't Play Chess. *Robotics and Autonomous Systems* **6,** 3–15. ISSN: 09218890. https://doi.org/10.1016/S0921-8890(05)80025-9 (June 1990).

31. Brooks, R. A. Chronicle of cybernetics pioneers. *Nature* **467,** 156–157. ISSN: 0028-0836. https://doi.org/10.1038/467156a (September 2010).

32. Brownlee, J. *Swarm Algorithms* in *Clever Algorithms* First, 423 (LuLu, 2011). ISBN: 9781446785065. https://github.com/clever-algorithms/CleverAlgorithms.

33. Brugnaro, G. & Hanna, S. *Adaptive Robotic Carving* in *Robotic Fabrication in Architecture, Art and Design 2018* (eds Willmann, J., Block, P., Hutter, M., Byrne, K. & Schork, T.) (Springer International Publishing, Cham, 2019), 336–348. ISBN: 978-3-319-92294-2. https://doi.org/10.1007/978-3-319-92294-2_26.

34. Bryson, J. J., Ando, Y. & Lehmann, H. Agent-based modelling as scientific method: a case study analysing primate social behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences* **362,** 1685–1699. ISSN: 0962-8436. https://doi.org/10.1098/rstb.2007.2061 (September 2007).

35. Burger, N. & Billington, D. P. Felix Candela, Elegance and Endurance: An examination of the Xochimilco shell. *Journal of the International Association for Shell and Spatial Structures* **47,** 271–278. ISSN: 1028-365X (2006).

36. Camedda, G. *Nuraghe Goni* [Photograph]. 2008. https://www.flickr.com/photos/24166512@N04/2294947340/ (2021).

37. Carmichael, T. *An Overview of Agent Based Models* in *Studies in Computational Intelligence* 19–28 (Springer Verlag, 2013). ISBN: 9783642392948. https://doi.org/10.1007/978-3-642-39295-5_3.

38. Carmichael, T. & Hadžikadić, M. *The Fundamentals of Complex Adaptive Systems* in *Understanding Complex Systems* 1–16 (2019). ISBN: 9783030203092. https://doi.org/10.1007/978-3-030-20309-2_1.

39. Chen, L. Agent-based modeling in urban and architectural research: A brief literature review. *Frontiers of Architectural Research* **1,** 166–177. ISSN: 2095-2635. https://doi.org/10.1016/j.foar.2012.03.003 (June 2012).

40. Coates, P. & Schmid, C. *Agent Based Modelling* in *Architectural Computing from Turing to 2000 [eCAADe Conference Proceedings]* (Liverpool, UK, 1999), 652–661.

41. Cohen-Steiner, D., Alliez, P. & Desbrun, M. Variational shape approximation. *ACM Transactions on Graphics* **23,** 905. ISSN: 0730-0301. https://doi.org/10.1145/1015706.1015817 (August 2004).

42. Consorzio Turistico dei Laghi. *Interior view of Nuraghe Is Paras* [Photograph]. 2014. http://laghienuraghi.it/it/content/nuraghe-paras (2021).

43. Copeland, B. J. *The Modern History of Computing* (ed Zalta, E. N.) 2017. https://plato.stanford.edu/archives/win2017/entries/computing-history/.

44. Costa, A. & Nannicini, G. RBFOpt: an open-source library for black-box optimization with costly function evaluations. *Mathematical Programming Computation* **10,** 597–629. ISSN: 1867-2949. https://doi.org/10.1007/s12532-018-0144-7 (December 2018).

45. Curtis, S. K., Mattson, C. a., Hancock, B. J. & Lewis, P. K. Divergent exploration in design with a dynamic multiobjective optimization formulation. *Structural and Multidisciplinary Optimization* **47,** 645–657. ISSN: 1615-147X. https://doi.org/10.1007/s00158-012-0855-8 (May 2013).

46. Davis, D. *Modelled on Software Engineering: Flexible Parametric Models in the Practice of Architecture* PhD thesis (RMIT University, Melbourne, 2013).

47. De Wolf, T. & Holvoet, T. *Emergence Versus Self-Organisation: Different Concepts but Promising When Combined* in *Engineering Self-Organising Systems: Methodologies and Applications* (eds Brueckner, S. A., Serugendo, G. D. M., Karageorgos, A. & Nagpal, R.) 1–15 (Springer, Berlin Heidelberg, 2005). ISBN: 978-3-540-26180-3. https://doi.org/10.1007/11494676_1.

48. Depalmas, A. & Melis, R. T. *The Nuragic People: Their Settlements, Economic Activities and Use of the Land, Sardinia, Italy* in *Landscapes and Societies* (eds Martini, I. P. & Chesworth, W.) 167–186 (Springer Netherlands, Dordrecht, 2010). ISBN: 978-90-481-9412-4. https://doi.org/10.1007/978-90-481-9413-1_11.

49. DHWR. *Der Wald in Deutschland - eine Erfolgsgeschichte* 2017. http://www.dhwr.de/themen.php?id=31 (2017).

50. Dorigo, M. Ant colony optimization. *Scholarpedia* **2,** 1461. ISSN: 1941-6016. http://www.scholarpedia.org/article/Ant_colony_optimization (2007).

51. Dorigo, M., Oca, M. & Engelbrecht, A. Particle swarm optimization. *Scholarpedia* **3,** 1486. ISSN: 1941-6016. http://www.scholarpedia.org/article/Particle_swarm_optimization (2008).

52. Dubberly, H. & Bigelow, J. *How cybernetics connects computing, counterculture, and design* in *Hippie Modernism: The Struggle for Utopia* (ed Blauvelt, A.) October, 1–12 (Walker Art Center, 2015).

53. Eigensatz, M. *et al.* Paneling Architectural Freeform Surfaces. *ACM Transactions on Graphics* **29** (2010).

54. Emery, C. *Candela exhibit explores marriage of art, engineering* (ed Quiñones, E.) Princeton, New Jersey, October 2008. https://pr.princeton.edu/pwb/volume98/issue05/candela/.

55. Fildhuth, T., Lippert, S. & Knippers, J. *Design and joint pattern optimisation of glass shells* in *Proceedings of The International Association for Shell and Spacial Structures* (Seoul, Korea, 20012).

56. Fleischmann, M., Knippers, J., Lienhard, J., Menges, A. & Schleicher, S. Material Behaviour: Embedding Physical Properties in Computational Design Processes. *Architectural Design* **82,** 44–51. ISSN: 00038504. https://doi.org/10.1002/ad.1378 (March 2012).
57. Forrest, S. & Mitchell, M. Adaptive computation. *Communications of the ACM* **59,** 58–63. ISSN: 00010782. https://doi.org/10.1145/2964342 (July 2016).
58. Fountain, H. *Wood That Reaches New Heights* New York, NY, 2012. http://www.nytimes.com/2012/06/05/science/lofty-ambitions-for-cross-laminated-timber-panels.html.
59. Frazer, J. *A Natural Model for Architecture* in *An Evolutionary Architecture* 9–22 (Architectural Association, London, 1995). ISBN: 1 870890 47 7.
60. Gamma, E., Helm, R., Johnson, R. & Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software* 62–65 (1994).
61. Garlock, M. E. M. & Billington, D. P. *Los Manatiales Restaurant at Xochimilco* in *Félix Candela: Engineer, Builder, Structural Artist* 142–153 (Princeton University Art Museum, Princeton, New Jersey, 2008). ISBN: 978-0-300-12209-1.
62. Garlock, M. E. M. & Billington, D. P. *Felix Candela and Heinz Isler: A comparison of two structural artists* in *Shell Structures for Architecture: Form Finding and Optimization* (eds Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C.) 247–258 (Routledge, London and New York, 2014). ISBN: 9781315849270.
63. Gendreau, M. & Potvin, J.-Y. *Preface* in *Handbook of Metaheuristics* (eds Gendreau, M. & Potvin, J.-Y.) Second, vii–viii (Springer US, Boston, MA, 2010). ISBN: 978-1-4419-1663-1. https://doi.org/10.1007/978-1-4419-1665-5.
64. Gerber, D. J., Pantazis, E. & Wang, A. A multi-agent approach for performance based architecture: Design exploring geometry, user, and environmental agencies in façades. *Automation in Construction* **76,** 45–58. ISSN: 09265805. https://doi.org/10.1016/j.autcon.2017.01.001 (April 2017).
65. Gobrecht, R. *Prinzipien in der Philisophie* ISBN: 9783732286652 (BoD - Books on Demand, Norderstedt, 2014).
66. Gordon, J. E. *The philosophy of design* in *Structures: or why things don't fall down* 2nd ed., 303–323 (Da Capo Press, Cambridge, MA, 2003). ISBN: 0-306-81283-5.
67. Graubner, W. *Flächenverbindungen* in *Holzverbindungen : Gegenüberstellung japanischer und europäischer Lösungen* 1st ed., 146–161 (Deutsche Verlags-Anstalt, Munich, Germany, 2015).
68. Grignard, A. *et al. GAMA 1.6: Advancing the Art of Complex Agent-Based Modeling and Simulation* in *PRIMA 2013: Principles and Practice of Multi-Agent Systems* 117–131 (2013). ISBN: 9783642449277. https://doi.org/10.1007/978-3-642-44927-7_9.
69. Grimm, V. *et al.* Pattern-Oriented Modeling of Agent-Based Complex Systems: Lessons from Ecology. *Science* **310,** 987–991. ISSN: 0036-8075. https://doi.org/10.1126/science.1116681 (November 2005).
70. Grimm, V. *et al.* The ODD protocol: A review and first update. *Ecological Modelling* **221,** 2760–2768. ISSN: 0304-3800. https://doi.org/10.1016/j.ecolmodel.2010.08.019 (2010).

71. Grimm, V. *et al.* The ODD Protocol for Describing Agent-Based and Other Simulation Models: A Second Update to Improve Clarity, Replication, and Structural Realism. *Journal of Artificial Societies and Social Simulation* **23,** 7. ISSN: 1460-7425. http://jasss.soc.surrey.ac.uk/23/2/7.html (2020).

72. Groenewolt, A., Schwinn, T., Nguyen, L. & Menges, A. An interactive agent-based framework for materialization-informed architectural design. *Swarm Intelligence* **12,** 155–186. ISSN: 1935-3812. https://doi.org/10.1007/s11721-017-0151-8 (June 2018).

73. Grun, T. B. *et al. The Skeleton of the Sand Dollar as a Biological Role Model for Segmented Shells in Building Construction: A Research Review* in *Biomimetic Research for Architecture and Building Construction: Biological Design and Integrative Structures* (eds Knippers, J., Nickel, K. G. & Speck, T.) 217–242 (Springer International Publishing, Cham, 2016). ISBN: 978-3-319-46374-2. https://doi.org/10.1007/978-3-319-46374-2_11.

74. Heath, B. L. & Hill, R. R. Some insights into the emergence of agent-based modelling. *Journal of Simulation* **4,** 163–169. ISSN: 1747-7778. https://doi.org/10.1057/jos.2010.16 (September 2010).

75. Heath, B. L., Hill, R. R. & Ciarallo, F. W. A Survey of Agent-Based Modeling Practices (January 1998 to July 2008). *Journal of Artificial Societies and Social Simulation* **12.** http://jasss.soc.surrey.ac.uk/12/4/9.html (2009).

76. Hegselmann, R. Thomas C. Schelling and James M. Sakoda: The Intellectual, Technical, and Social History of a Model. *Journal of Artificial Societies and Social Simulation* **20.** ISSN: 1460-7425. http://jasss.soc.surrey.ac.uk/20/3/15.html (2017).

77. Helbing, D., Buzna, L., Johansson, A. & Werner, T. Self-Organized Pedestrian Crowd Dynamics: Experiments, Simulations, and Design Solutions. *Transportation Science* **39,** 1–24. ISSN: 0041-1655. https://doi.org/10.1287/trsc.1040.0108 (February 2005).

78. Hensel, M. & Menges, A. Versatility and Vicissitude: An Introduction to Performance in Morpho-Ecological Design. *Architectural Design* **78,** 6–11. ISSN: 00038504. https://doi.org/10.1002/ad.635 (March 2008).

79. Hensel, M., Menges, A. & Weinstock, M. Emergence in Architecture. *Architectural Design* **74,** 6–9. ISSN: 00038504 (2004).

80. Hess & Co AG. *Sperrholz aus Buchenfurnieren* Berlin, 2013.

81. Holland, J. H. *Hidden Order* ISBN: 0-201-44230-2 (Basic Books, New York, NY, 1995).

82. Jennings, N. R. On agent-based software engineering. *Artificial Intelligence* **117,** 277–296. ISSN: 00043702. https://doi.org/10.1016/S0004-3702(99)00107-1 (March 2000).

83. Jiang, C., Wang, J., Wallner, J. & Pottmann, H. Freeform Honeycomb Structures. *Computer Graphics Forum* **33,** 185–194. ISSN: 01677055. https://doi.org/10.1111/cgf.12444 (August 2014).

84. Johns, R. L., Kilian, A. & Foley, N. *Design Approaches Through Augmented Materiality and Embodied Computation* in *Robotic Fabrication in Architecture, Art and Design 2014* (eds McGee, W. & Ponce de Leon, M.) 319–332 (Springer International Publishing, 2014). ISBN: 978-3-319-04662-4. https://doi.org/10.1007/978-3-319-04663-1_22.

85. Johnson, S. *Emergence: The Connected Lives of Ants, Brains, Cities, and Software* First. ISBN: 0-684-86876-8 (Touchstone, New York, NY, 2020).

86. Juziuk, J., Weyns, D. & Holvoet, T. *Design Patterns for Multi-agent Systems: A Systematic Literature Review* in *Agent-Oriented Software Engineering: Reflections on Architectures, Methodologies, Languages, and Frameworks* (eds Shehory, O. & Sturm, A.) 79–99 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2014). ISBN: 978-3-642-54432-3. https://doi.org/10.1007/978-3-642-54432-3_5.

87. Kaveh, A., Azar, B. F. & Talatahari, S. Ant Colony Optimization for Design of Space Trusses. *International Journal of Space Structures* **23,** 167–181. https://doi.org/10.1260/026635108786260956 (2008).

88. Kelly, K. *Hive mind* in *Out of Control: The New Biology of Machines, Social Systems, and the Economic World* 5–28 (Perseus Books, New York, NY, 1994). ISBN: 978-0201483406.

89. Kilian, A. *Steering of form* in *Shell Structures for Architecture: Form Finding and Optimization* (eds Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C.) 131–139 (Routledge, London and New York, 2014). ISBN: 9781315849270.

90. Kirby, I. J. *The Dovetail Family* in *The Complete Dovetail : Handmade Furniture's Signature Joint* First, 11–32 (Stobart Davies Ltd., Hertford, UK, 1999). ISBN: 0-85442-085-1.

91. Knippers, J., Bischoff, M., Nebelsick, J. H., Menges, A. & Verl, A. *A07: The Skeleton of the Sand Dollar as a Biological Role Model for Segmented Shells in Building Construction* in *Biological Design and Integrative Structures: Analysis, Simulation and Implementation in Architecture* 183–198 (Stuttgart, 2014).

92. Knippers, J. & Speck, T. Design and construction principles in nature and architecture. *Bioinspiration & Biomimetics* **7,** 015002. ISSN: 1748-3190. https://doi.org/10.1088/1748-3182/7/1/015002 (March 2012).

93. Kolb, J. *Timber engineering: loadbearing structures and component layers* 319. ISBN: 978-3-7643-8689-4 (Birkhäuser, Basel, 2008).

94. Kouadri, S. *et al. Self-Organising Applications : A Survey* in *Proceedings of the International Workshop on Engineering Self-Organising Applications* (Engineering Self-Organising Applications Working Group, 2003).

95. Kratzer, M., Binz, H. & Watty, R. *Integration von DFX-Kriterien in das agentenbasierte Unterstützungssystem PROKON* in *DFX 2009: Proceedings of the 20th Symposium on Design for X* (ed Meerkamm, H.) (Neunkirchen/Erlangen, 2009), 81–92. ISBN: 3-9808539-3-4.

96. Krause, J. *Agent Generated Architecture* in *Proceedings of ACADIA '97 Conference* (Cincinatti, Ohio, 1997), 63–70.

97. Krieg, O. D., Dierichs, K., Reichert, S., Schwinn, T. & Menges, A. *Performative architectural morphology: Finger-joined plate structures integrating robotic manufacturing, biological principles and location-specific requirements* in *Computational Design Modelling: Proceedings of the Design Modelling Symposium Berlin 2011* (eds Gengnagel, C., Kilian, A., Palz, N. & Scheurer, F.) 259–266 (Springer, Berlin, Heidelberg, 2011). https://doi.org/10.1007/978-3-642-23435-4_29.

98. Krieg, O. D., Dierichs, K., Reichert, S., Schwinn, T. & Menges, A. *Performative Architectural Morphology: Robotically manufactured biomimetic finger-joined plate structures* in *RESPECTING FRAGILE PLACES - 29th eCAADe Conference Proceedings* (Ljubljana, 2011), 573–580.

99. Krieg, O. D. & Menges, A. *Potentials of Robotic Fabrication in Wood Construction* in *ACADIA 13: Adaptive Architecture [Proceedings of the 33rd Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)]* (eds Beesley, P., Khan, O. & Stacey, M.) (Cambridge (Ontario), 2013), 253–260. ISBN: 978-1-926724-22-5.

100. Krieg, O. D., Mihaylov, B., Schwinn, T., Reichert, S. & Menges, A. *Computational Design of Robotically Manufactured Plate Structures Based on Biomimetic Design Principles Derived from Clypeasteroida* in *Digital Physicality - Proceedings of the 30th eCAADe Conference* (eds Achten, H., Pavlicek, J., Hulin, J. & Matejdan, D.) (Prague, Czech Republic, 2012), 521–530.

101. Krieg, O. D., Schwinn, T. & Menges, A. Neue Holztechnologien : Robotisch gefertigter Leichtbau. *Holztechnologie* **56,** 20–26 (2015).

102. Krieg, O. D., Schwinn, T. & Menges, A. *Integrative Design Computation for Local Resource Effectiveness in Architecture* in *Urbanization and Locality* (eds Wang, F. & Prominski, M.) First, 123–143 (Springer, Berlin, Heidelberg, 2016). https://doi.org/10.1007/978-3-662-48494-4_7.

103. Krieg, O. D. *et al. Biomimetic Lightweight Timber Plate Shells: Computational Integration of Robotic Fabrication, Architectural Geometry and Structural Design* in *Advances in Architectural Geometry 2014* (eds Ceccato, C., Hesselgren, L., Pauly, M., Pottmann, H. & Wallner, J.) 11, 109–125 (Springer International Publishing, Cham, 2015). ISBN: 978-3-7091-0308-1. https://doi.org/10.1007/978-3-319-11418-7_8.

104. Kühn, C. Christopher Alexander's pattern language - Von der „Notes on the Synthesis of Form“ zur „Pattern Language“. *Arch+,* 26–31. ISSN: 0587-3452 (2008).

105. KUKA Roboter GmbH. *KR QUANTEC extra* (2013).

106. Kwinter, S. The Computational Fallacy. *Thresholds* **26,** 90–92. https://archive.org/details/thresholds262003mass (2003).

107. La Magna, R., Schleicher, S. & Knippers, J. *Bending-Active Plates* in *Advances in Architectural Geometry 2016* (eds Adriaenssens, S., Gramazio, F., Kohler, M., Menges, A. & Pauly, M.) September, 171–186 (vdf Hochschulverlag AG an der ETH Zürich, Zürich, 2016). ISBN: 978-3-7281-3778-4.

108. La Magna, R., Waimer, F. & Knippers, J. *Nature-inspired generation scheme for shell structures* in *Proceedings of the International Association for Shell and Spatial Stuctures (IASS) Symposium* (Seoul, Korea, 2012).

109. La Magna, R. *et al.* From Nature to Fabrication: Biomimetic Design Principles for the Production of Complex Spatial Structures. *International Journal of Space Structures* **28,** 27–40. ISSN: 0266-3511. https://doi.org/10.1260/0266-3511.28.1.27 (March 2013).

110. Langton, C. G. *Artificial Life* in *Ars Electronica 93: Genetic Art - Artificial Life* (eds Gerbel, K. & Weibel, P.) 244 (PVS-Verleger, 1993). ISBN: 9783901196072.

111. Lawrence, A. *Timber renaissance* 2014. http://www.theengineer.co.uk/home/blog/guest-blog/timber-renaissance/1013772.article (2014).

112. Leach, N. Swarm Urbanism. *Architectural Design* **79,** 56–63. ISSN: 0003-8504. https://doi.org/10.1002/ad.918 (July 2009).

113. Levitis, D. A., Lidicker, W. Z. & Freund, G. Behavioural biologists do not agree on what constitutes behaviour. *Animal Behaviour* **78,** 103–110. ISSN: 00033472. https://doi.org/10.1016/j.anbehav.2009.03.018 (July 2009).

114. Li, J.-M. & Knippers, J. *Pattern and Form - Their Influence on Segmental Plate Shells* in *Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium 2015* (Amsterdam, 2015).

115. Li, J.-M. & Knippers, J. Segmental Timber Plate Shell for the Landesgartenschau Exhibition Hall in Schwäbisch Gmünd—the Application of Finger Joints in Plate Structures. *International Journal of Space Structures* **30,** 123–140. ISSN: 0266-3511. https://doi.org/10.1260/0266-3511.30.2.123 (June 2015).

116. Li, Y., Liu, Y. & Wang, W. Planar Hexagonal Meshing for Architecture. *IEEE Transactions on Visualization and Computer Graphics* **21,** 95–106. ISSN: 1077-2626. https://doi.org/10.1109/TVCG.2014.2322367 (January 2015).

117. Linkwitz, K. *Force Density Method: Design of a timber shell* in *Shell Structures for Architecture: Form Finding and Optimization* (eds Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C.) 59–69 (Routledge, London and New York, 2014). ISBN: 9781315849270.

118. Liu, Y. *et al.* General planar quadrilateral mesh design using conjugate direction field. *ACM Transactions on Graphics* **30,** 1. ISSN: 07300301. https://doi.org/10.1145/2070781.2024174 (December 2011).

119. Macal, C. & North, M. *Tutorial on Agent-Based Modeling and Simulation PART 2: How to Model with Agents* in *Proceedings of the 2006 Winter Simulation Conference* (IEEE, December 2006), 73–83. ISBN: 1-4244-0501-7. https://doi.org/10.1109/WSC.2006.323040.

120. Macal, C. M. *Agent-based Modeling and Artificial Life* in *Encyclopedia of Complexity and Systems Science* (ed Meyers, R. A.) 112–131 (Springer New York, New York, NY, 2009). ISBN: 978-0-387-75888-6. https://doi.org/10.1007/978-0-387-30440-3_7.

121. Macal, C. M. Everything you need to know about agent-based modelling and simulation. *Journal of Simulation* **10,** 144–156. ISSN: 1747-7778. https://doi.org/10.1057/jos.2016.7 (May 2016).

122. Macal, C. M. & North, M. J. *Agent-based modeling and simulation* in *Proceedings of the 2009 Winter Simulation Conference (WSC)* (2009), 86–98. ISBN: 978-1-4244-5770-0.

123. Macal, C. M. & North, M. J. *Toward Teaching Agent-based Simulation* in *Proceedings of the 2010 Winter Simulation Conference* (eds Johansson, B., Jain, S., Montoya-Torres, J., Hugan, J. & Yücesan, E.) (2010).

124. Macal, C. M. & North, M. J. Tutorial on agent-based modelling and simulation. *Journal of Simulation* **4,** 151–162. ISSN: 1747-7778. https://doi.org/10.1057/jos.2010.3 (September 2010).

125. MacLeamy, P. *Collaboration, Integrated Information and the Project Lifecycle in Building Design, Construction and Operation* tech. rep. WP-1202 (Construction Users Roundtable, 2004), 14. http://codebim.com/wp-content/uploads/2013/06/CurtCollaboration.pdf.

126. Manahl, M., Stavric, M. & Wiltsche, A. Ornamental Discretisation of Free-form Surfaces. *International Journal of Architectural Computing* **10,** 595–612. ISSN: 1478-0771. https://doi.org/10.1260/1478-0771.10.4.595 (December 2012).

127. Matloff, N. *Introduction to Discrete-Event Simulation and the SimPy Language* 2008. http://heather.cs.ucdavis.edu/~matloff/156/PLN/DESimIntro.pdf.

128. Mendes, R., Kennedy, J. & Neves, J. The Fully Informed Particle Swarm: Simpler, Maybe Better. *IEEE Transactions on Evolutionary Computation* **8,** 204–210. ISSN: 1089-778X. https://www.doi.org/10.1109/TEVC.2004.826074 (June 2004).

129. Menges, A. *Morphospaces of Robotic Fabrication* in *Rob | Arch 2012* (eds Brell-Çokcan, S. & Braumann, J.) 28–47 (Springer Vienna, Vienna, 2013). ISBN: 978-3-7091-1464-3. https://doi.org/10.1007/978-3-7091-1465-0_3.

130. Menges, A. & Schwinn, T. Manufacturing Reciprocities. *Architectural Design* **82,** 118–125. ISSN: 00038504. https://doi.org/10.1002/ad.1388 (March 2012).

131. Mesa. *Introductory Tutorial* 2016. http://mesa.readthedocs.io/en/latest/tutorials/intro_tutorial.html (2017).

132. Metz, T. *Agent-Based Modeling (ABM)* in *Neue Trends in den Sozialwissenschaften* (ed Jäckle, S.) 11–50 (Springer Fachmedien Wiesbaden, Wiesbaden, 2017). ISBN: 978-3-658-17189-6. https://doi.org/10.1007/978-3-658-17189-6_2.

133. Michel, F., Ferber, J. & Drogoul, A. *Multi-Agent Systems and Simulation: A Survey from the Agent Community's Perspective* in *Multi-Agent Systems: Simulation and Applications* (eds Uhrmacher, A. M. & Weyns, D.) 3–52 (CRC Press, 2009). ISBN: 978-1-4200-7023-1.

134. Minar, N., Burkhart, R., Langton, C. & Askenazi, M. *The Swarm Simulation System : A Toolkit for Building Multi-agent Simulations* Santa Fe, NM, 1996.

135. Miranda Carranza, P. & Coates, P. Swarm modelling. The use of Swarm Intelligence to generate architectural form. *3rd International Conference on Generative Art,* 1–16. https://www.generativeart.com/on/cic/2000/CARRANZA_COATES.HTM (2000).

136. Moro, J. L., Rottner, M., Alihodzic, B. & Weißbach, M. *Holzprodukte* in *Baukonstruktion - vom Prinzip zum Detail : Band 1 Grundlagen* 267–284 (Springer, Berlin, Heidelberg, 2009). ISBN: 978-3-540-27917-4. https://doi.org/10.1007/978-3-540-27917-4_15.

137. Murphy, R. R. *Biological Foundations of the Reactive Paradigm* in *Introduction to AI Robotics* 67–103 (Massachusetts Institute of Technology, Cambridge, MA, 2000). ISBN: 9781849208826.

138. Nachtigall, W. *Bionik als Ansatz zum strukturierten Erfinden* in *Bionik als Wissenschaft* 179–200 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010). ISBN: 9783642103209. http://link.springer.com/10.1007/978-3-642-10320-9_10.

139. Nachtigall, W. & Pohl, G. *Bau, Architektur und Bionik* in *Bau-Bionik* 2nd, 7–18 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2013). ISBN: 978-3-540-88994-6. https://doi.org/10.1007/978-3-540-88995-3_2.

140. Ney, L. & Adriaenssens, S. *Shaping forces* in *Shell Structures for Architecture: Form Finding and Optimization* (eds Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C.) 15–19 (Routledge, London and New York, 2014). ISBN: 9789490814007.

141. Nguyen, L. *et al. Evolutionary Processes as Models for Exploratory Design* in *Biomimetic Research for Architecture and Building Construction: Biological Design and Integrative Structures* (eds Knippers, J., Nickel, K. G. & Speck, T.) 295–318 (Springer International Publishing, Cham, 2016). ISBN: 978-3-319-46372-8. https://doi.org/10.1007/978-3-319-46374-2_15.

142. Nikolai, C. & Madey, G. Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. *Journal of Artificial Societies and Social Simulation* **12,** 2. ISSN: 1460-7425. http://jasss.soc.surrey.ac.uk/12/2/2.html (2009).

143. Office of Energy Efficiency & Renewable Energy. *2011 Buildings Energy Data Book* (US Department of Energy, 2012).

144. Oval, R., Rippmann, M., Van Mele, T., Baverel, O. & Block, P. *Patterns for Masonry Vault Design* in *Proceedings of the IASS Symposium 2017* (Hamburg, September 2017).

145. Patrick Sisson. *Felix Candela, the architect who showcased concrete's curves* 2018. https://www.curbed.com/2018/1/25/16932400/felix-candela-architect-concrete-los-manantiales (2019).

146. Pevsner, N. *Frankreich und der Protestantische Norden* in *Europäische Architektur: Von den Anfängen bis zur Gegenwart* 8th ed., 255–307 (Prestel, München, New York, 1994). ISBN: 3-7913-1376-2.

147. Pirmann, D. *Pantheon, Rome* [Photograph]. 2012. https://www.flickr.com/photos/dpirmann/11329011256/ (2021).

148. Pottmann, H., Liu, Y., Wallner, J., Bobenko, A. & Wang, W. Geometry of Multi-layer Freeform Structures for Architecture. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2007* **26,** Article No. 65 (2007).

149. Pottmann, H. & Wallner, J. The focal geometry of circular and conical meshes. *Advances in Computational Mathematics* **29,** 249–268. ISSN: 1019-7168. https://doi.org/10.1007/s10444-007-9045-4 (August 2007).

150. Prado, M., Dörstelmann, M., Menges, A., Solly, J. & Knippers, J. *Elytra Filament Pavilion : Robotic Filament Winding for Structural Composite Building Systems* in *FABRICATE - RETHINKING DESIGN AND CONSTRUCTION* (eds Menges, A., Sheil, B., Glynn, R. & Skavara, M.) 224–231 (UCL Press, London, UK, 2017). ISBN: 978-1-78735-001-4.

151. Prado, M., Dörstelmann, M., Schwinn, T., Menges, A. & Knippers, J. *Core-Less Filament Winding* in *Robotic Fabrication in Architecture, Art and Design 2014* (eds McGee, W. & Ponce de Leon, M.) 275–289 (Springer International Publishing, Cham, 2014). ISBN: 978-3-319-04662-4. https://doi.org/10.1007/978-3-319-04663-1_19.

152. RaBoe/Wikipedia. *Temple of Mercury, Baiae* [Photograph]. 2010. https://commons.wikimedia.org/wiki/File:Baia-Complesso_Termal_Romano_2010-by-RaBoe-116.jpg (2021).

153. Railsback, S. F. & Grimm, V. *Introduction to Part III* in *Agent-Based and Individual-Based Modeling: A Practical Introduction* First, 227–231 (Princeton University Press, 2012). ISBN: 9780691136745.

154. Railsback, S. F., Lytinen, S. L. & Jackson, S. K. Agent-based Simulation Platforms: Review and Development Recommendations. *SIMULATION* **82,** 609–623. ISSN: 0037-5497. https://doi.org/10.1177/0037549706073695 (September 2006).

155. Reckhaus, M., Hochgeschwender, N., Ploeger, P. G. & Kraetzschmar, G. K. A Platform-independent Programming Environment for Robot Control. *Applied Sciences.* arXiv: 1010.0886. http://arxiv.org/abs/1010.0886 (October 2010).

156. Reynolds, C. W. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics* **21** (ed Stone, M. C.) 25–34. ISSN: 00978930. https://doi.org/10.1145/37401.37406 (August 1987).

157. Reynolds, C. W. *Not bumping into things* in *ACM SIGGRAPH Course 27 Notes: Developments in Physically-Based Modeling* G1–G13 (ACM Press, 1988). https://doi.org/10.13140/RG.2.2.12202.62407.

158. Reynolds, C. W. *Steering Behaviors For Autonomous Characters* in *Proceedings of Game Developers Conference* (1999), 763–782.

159. RIBA Collections. *Los Manantiales restaurant, the Floating Gardens of Xochimilco, Mexico City* [Photograph]. 1958. https://www.ribapix.com/los-manantiales-restaurant-the-floating-gardens-of-xochimilco-mexico-city_riba76940 (2021).

160. Robeller, C., Stitic, A., Mayencourt, P. & Weinand, Y. *Interlocking Folded Plate: Integrated Mechanical Attachment for Structural Wood Panels* in *Advances in Architectural Geometry 2014* (eds Block, P., Knippers, J., Mitra, N. J. & Wang, W.) 281–294 (Springer International Publishing, Cham, 2015). ISBN: 978-3-319-11418-7. https://doi.org/10.1007/978-3-319-11418-7_18.

161. Robeller, C. & Weinand, Y. *Integral joints for timber folded plate structures* in *Advancing Wood Architecture: A Computational Approach* (eds Menges, A., Schwinn, T. & Krieg, O. D.) 73–83 (Routledge, London and New York, 2016). ISBN: 9781317392347.

162. Sabin, M. & Gray, R. Global convergence and empirical consistency of the generalized Lloyd algorithm. *IEEE Transactions on Information Theory* **32,** 148–155. ISSN: 0018-9448. https://www.doi.org/10.1109/TIT.1986.1057168 (March 1986).

163. Saint, A. *The Reputation of St. Paul's* in *St. Paul's: The Cathedral Church of London 604-2004* (eds Keene, D., Burns, A. & Saint, A.) 451–463 (Yale University Press, New Haven and London, 2004). ISBN: 0-300-09276-8.

164. Sasaki, M. *Structural design of free-curved RC shells: an overview of built works* in *Shell Structures for Architecture: Form Finding and Optimization* (eds Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C.) 259–273 (Routledge, London and New York, 2014). ISBN: 9781315849270.

165. Scheurer, F. *Turning the Design Process Downside-up* in *Computer Aided Architectural Design Futures 2005* 269–278 (Springer-Verlag, Berlin/Heidelberg, 2005). https://doi.org/10.1007/1-4020-3698-1_25.

166. Schindler, C. *Ein architektonisches Periodisierungsmodell anhand fertigungstechnischer Kriterien, dargestellt am Beispiel des Holzbaus.* Abhandlung zur Erlangung des Titels Doktor der Wissenschaften der ETH Zürich (Dr. sc. ETH Zürich) (ETH Zürich, 2009).

167. Schindler, C. Die Standards des Nonstandards. *GAM - Graz Architecture Magazine 06,* 181–193 (2010).

168. Schindler, C. & Scheurer, F. *Architektonische Anwendungen des fünfachsigen Flankenfräsens* in *Proceedings of the 4th Conference on CNC-Milling Technology in Architecture, Art and Design* (Vienna, 2007).

169. Schmitt, A. & Schwieger, V. *Quality Control of Robotics Made Timber Plates* in *FIG Working Week 2015* (Sofia, Bulgaria, 2015).

170. Schwinn, T. *Manufacturing Perspectives* in *Advancing Wood Architecture* (eds Menges, A., Schwinn, T. & Krieg, O. D.) 185–198 (Routledge, New York, 2016). ISBN: 978-1-138-93298-2.

171. Schwinn, T., Krieg, O. D. & Menges, A. *Robotically Fabricated Wood Plate Morphologies* in *Rob | Arch 2012* (eds Brell-Çokcan, S. & Braumann, J.) 48–61 (Springer, Vienna, 2013). ISBN: 978-3709114643. https://doi.org/10.1007/978-3-7091-1465-0_4.

172. Schwinn, T., Krieg, O. D. & Menges, A. *Behavioral Strategies: Synthesizing design computation and robotic fabrication of lightweight timber plate structures* in *Design Agency [Proceedings of the 34th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)]* (Los Angeles, 2014), 177–188. ISBN: 9781926724478.

173. Schwinn, T., Krieg, O. D. & Menges, A. *Robotic Sewing: A Textile Approach Towards the Computational Design and Fabrication of Lightweight Timber Shells* in *POSTHUMAN FRONTIERS: Data, Designers, and Cognitive Machines* (eds Velikov, K., Ahlquist, S., del Campo, M. & Thün, G.) (Ann Arbor, 2016), 224–233.

174. Schwinn, T., Krieg, O. D., Menges, A., Mihaylov, B. & Reichert, S. *Machinic Morphospaces: Biomimetic Design Strategies for the Computational Exploration of Robot Constraint Spaces for Wood Fabrication* in *ACADIA 12: Synthetic Digital Ecologies [Proceedings of the 32nd Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)]* (San Francisco, 2012), 157–168. ISBN: 978-1-62407-267-3.

175. Schwinn, T. & Menges, A. Fabrication Agency: Landesgartenschau Exhibition Hall. *Architectural Design* **85,** 92–99. ISSN: 00038504. https://doi.org/10.1002/ad.1960 (September 2015).

176. Schwinn, T. *et al. Prototyping Biomimetic Structures for Architecture* in *Prototyping Architecture: The Conference Papers* (ed Stacey, M.) (The Building Centre Trust, London, 2013), 224–244. ISBN: 978-0-901919-17-5.

177. Schwinn, T. *et al. Potential applications of segmented shells in architecture* in *Biomimetics for Architecture* (eds Knippers, J., Schmid, U. & Speck, T.) 116–125 (De Gruyter, Berlin, Boston, June 2019). https://doi.org/10.1515/9783035617917-015.

178. Shiffman, D. *Autonomous Agents* in *The Nature of Code* (ed Fry, S.) 260–322 (Daniel Shiffman, New York, NY, 2012). ISBN: 978-0985930806.

179. Siebers, P. O., Macal, C. M., Garnett, J., Buxton, D. & Pidd, M. Discrete-event simulation is dead, long live agent-based simulation! *Journal of Simulation* **4,** 204–210. ISSN: 1747-7778. https://doi.org/10.1057/jos.2010.14 (September 2010).

180. Silver, D. *Introduction to Reinforcement Learning* 2015. https://www.davidsilver.uk/teaching/ (2020).

181. Snooks, R. Volatile Formation. *Log,* 55–62 (2012).

182. Snooks, R. *Behavioral Formation: Multi-Agent Algorithmic Design Strategies* PhD (RMIT University, 2014).

183. Snooks, R. & Jahn, G. *Stigmergic Accretion* in *Robotic Fabrication in Architecture, Art and Design 2016* (eds McGee, W. & Ponce de Leon, M.) 2012, 398–409 (Springer International Publishing, Cham, 2016). ISBN: 978-3-319-04662-4. https://doi.org/10.1007/978-3-319-26378-6_32.

184. Sobek, W. Ultra-lightweight construction. *International Journal of Space Structures* **31,** 74–80. ISSN: 0266-3511. https://doi.org/10.1177/0266351116643246 (March 2016).

185. Souza, C., Kirillov, A., Catalano, M. D. & contributors, A. *The Accord.NET Framework* 2014. http://accord-framework.net.

186. Springer Fachmedien Wiesbaden. *Objektplanung* in *HOAI 2013-Textausgabe/HOAI 2013-Text Edition* (ed Springer Fachmedien Wiesbaden) (Springer Vieweg, Wiesbaden, 2013). ISBN: 978-3-658-03269-2. https://doi.org/10.1007/978-3-658-03270-8_3.

187. Sustainable Building and Climate Initiative (SBCI). *Buildings and Climate Change : Summary for Decision-Makers* ISBN: 978-92-807-3064-7 (United Nations Environment Programme, Nairobi, 2009).

188. Tamke, M., Riiber, J., Jungjohann, H. & Thomsen, M. R. *Lamella Flock* in *Advances in Architectural Geometry 2010* 37–48 (Springer Vienna, Vienna, 2010). https://doi.org/10.1007/978-3-7091-0309-8_3.

189. The Economist. *Agents of change* 2010. http://www.economist.com/node/16636121 (2017).

190. Thiel, C. *Kettenschluss* in *Enzyklopädie Philosophie und Wissenschaftstheorie* (ed Mittelstraß, J.) 390 (Metzler, Stuttgart, 1996). ISBN: 3-476-02012-6.

191. Tidwell, J. *Designing Interfaces : Patterns for Effective Interaction Design* One. ISBN: 978-0-596-00803-1 (O'Reilly Media, Inc., 2005).

192. Troche, C. *Planar Hexagonal Meshes by Tangent Plane Intersection* in *Advances in Architectural Geometry 2008* (Vienna, 2008), 57–60. ISBN: 978-3-902233-03-5.

193. Turing, A. *Lecture on the Automatic Computing Engine (1947)* in *The Essential Turing* (ed Copeland, B. J.) 362–394 (Oxford University Press, Oxford, 2004).

194. UN Environment and International Energy Agency. *Towards a zero-emission, efficient, and resilient buildings and construction sector. Global Status Report 2017* ISBN: 978-92-807-3686-1 (United Nations Environment Programme, Nairobi, 2017).

195. Vasey, L. *et al. Behavioral Design and Adaptive Robotic Fabrication of a Fiber Composite Compression Shell With Pneumatic Formwork* in *Proceedings of ACADIA 2015* (2015), 297–309.

196. Veenendaal, D. & Block, P. *Comparison of form-finding methods* in *Shell Structures for Architecture: Form Finding and Optimization* (eds Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C.) 115–129 (Routledge, London and New York, 2014). ISBN: 978-0-415-84060-6.

197. Vermeulen, B., Pyka, A. & Müller, M. *An agent-based policy laboratory for COVID-19 containment strategies* 2020. https://inno.uni-hohenheim.de/corona-modell (2021).

198. Vogel, D. *Evolving Steering Behaviours with Genetic Programming* 2004. http://www.nonsequitoria.com/2521/ (2014).

199. Völkl, G. Wie Programme lernen. *iX,* 42–50. https://heise.de/-3807263 (2017).

200. Von Bertalanffy, L. *The Meaning of General System Theory* in *Computational Design Thinking* (ed Achim Menges, S. A.) 50–57 (Wiley, Chichester, U.K., 2011). ISBN: 978-0470665657.

201. Von Mammen, S. & Jacob, C. *Evolutionary swarm design of architectural idea models* in *Proceedings of the 10th annual conference on Genetic and evolutionary computation - GECCO '08* (ACM Press, New York, New York, USA, 2008), 143. ISBN: 9781605581309. https://doi.org/10.1145/1389095.1389115.

202. Walter, W. G. An Imitation of Life. *Scientific American* **182,** 42–45 (1950).

203. Wang, W. & Liu, Y. *A Note on Planar Hexagonal Meshes* in *Nonlinear Computational Geometry* (eds Emiris, I. Z., Sottile, F. & Theobald, T.) 221–233 (Springer, New York, NY, 2009). ISBN: 978-1-4419-0998-5. https://doi.org/10.1007/978-1-4419-0999-2_9.

204. Wang, W. *et al. Hexagonal Meshes with Planar Faces (TR-2008-13)* tech. rep. (Department of Computer Science, University of Hong Kong, Hong Kong, 2008).

205. Werfel, J., Petersen, K. & Nagpal, R. Designing Collective Behavior in a Termite-Inspired Robot Construction Team. *Science* **343,** 754–758. ISSN: 0036-8075. https://www.doi.org/10.1126/science.1245842 (February 2014).

206. Wester, T. *Design of Plate and Lattice Structures Based on Structural Dualism* in *Proceedings of the IASS Annual Symposium 1989* **2** (Madrid, 1989).

207. Wester, T. *The Dualistic Symmetry Between Plane- and Point-Based Spatial Structures* in *Proceedings of the Interdiciplinary Symposium on Symmetry of Structure* (eds Darvas, G. & Nagy, D.) **72** (Budapest, Hungary, 1989).

208. Wester, T. Nature Teaching Structures. *International Journal of Space Structures* **17,** 135–147. ISSN: 0266-3511. https://doi.org/10.1260/026635102320321789 (September 2002).

209. Westermann, Stephan. *Architekt und Bauunternehmer* 2019. https://www.baunetz.de/recht/Haftung-weitere_Beteiligte-Architekt_u._Bauunternehmer_40699.html (2019).

210. Williams, C. *What is a shell?* in *Shell Structures for Architecture: Form Finding and Optimization* (eds Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C.) 21–32 (Routledge, London and New York, 2014). ISBN: 978-0-415-84060-6.

211. Wolfram, S. Cellular automata as models of complexity. *Nature* **311,** 419–424. ISSN: 0028-0836. https://doi.org/10.1038/311419a0 (October 1984).

212. Woodbury, R. *Elements of Parametric Design* ISBN: 978-0-41577986-9 (Routledge, New York, 2010).

213. Wooldridge, M. & Jennings, N. R. Intelligent agents: theory and practice. *The Knowledge Engineering Review* **10,** 115–152 (1995).

214. Wortmann, T. Genetic Evolution vs. Function Approximation: Benchmarking Algorithms for Architectural Design Optimization. *Journal of Computational Design and Engineering* **6** (September 2018).

215. Xue, X., Li, X., Shen, Q. & Wang, Y. An agent-based framework for supply chain coordination in construction. *Automation in Construction.* ISSN: 09265805 (2005).

216. Yablonina, M. & Menges, A. Distributed Fabrication: Cooperative Making with Larger Groups of Smaller Machines. *Architectural Design* **89,** 62–69. ISSN: 00038504. https://doi.org/10.1002/ad.2413 (March 2019).

217. Zachos, L. G. A new computational growth model for sea urchin skeletons. *Journal of Theoretical Biology* **259,** 646–657. ISSN: 1095-8541. https://doi.org/10.1016/j.jtbi.2009.04.007 (August 2009).

218. Zadravec, M., Schiftner, A. & Wallner, J. Designing Quad-dominant Meshes with Planar Faces. *Computer Graphics Forum* **29,** 1671–1679. ISSN: 0167-7055. https://doi.org/10.1111/j.1467-8659.2010.01776.x (September 2010).

219. Zimmer, H., Campen, M., Herkrath, R. & Kobbelt, L. *Variational Tangent Plane Intersection for Planar Polygonal Meshing* in *Advances in Architectural Geometry 2012* (eds Hesselgren, L. *et al.*) (Springer Vienna, 2012), 319–332. ISBN: 978-3-7091-1250-2.

Image Credits

Unless otherwise noted, copyrights and credits of the figures in this book belong to the author. All reprints with permission.

Busch, B., ©2011 by ICD/ITKE University of Stuttgart, figures 6.2, 6.40 (right), 6.41 (right), 6.42 (right), 6.46, 6.54 | **Camedda, G.**, ©2008, figure 4.2 (left) | **CMD Ingenieros**, ©1999, figure 1.2 (right) | **Consorzio Turistico dei Laghi**, ©2014, figure 4.2 (right) | **Faulkner, R.**, ©2019 by ICD/ITKE University of Stuttgart, figure 1.3 | **Grunder, W.**, ©1986 by W. Grunder, figure 5.2 | **Guzmán, J.**, ©1958 by Archivo Fotográfico Manuel Toussaint del Instituto de Investigaciones Estéticas de la UNAM, figure 1.2 (left) | **Halbe, R.**, ©2014, figure 7.2 | **Hübsch, H.**, c. 1835, public domain, figure 4.3 (left and center) | **ICD/ITKE University of Stuttgart**, ©2011, figures 6.26 (right), 6.27 (right), 6.29 (right), 6.30 (right), 6.50 | **ICD/ITKE/IIGS University of Stuttgart**, ©2014, figure 7.10 (right) | **Krieg, O. D.**, ©2010 by O. D. Krieg, figure 6.4 | **Krieg, O. D.**, ©2011 by O. D. Krieg, figures 6.6, 6.48; by ICD University of Stuttgart, figure 6.1; by ICD/ITKE University of Stuttgart, figures 6.8 (right), 6.39 (right) | **Krieg, O. D.**, ©2012 by O. D. Krieg, figures 6.37 (right), 6.38 (right); by ICD/ITKE University of Stuttgart, figures 6.5, 6.7 (right), 6.9 (right), 6.12 (right), 6.13 (right), 6.14 (right), 6.17 (right), 6.25 (right), 7.9 (right) | **Krieg, O. D.**, ©2014 by ICD University of Stuttgart, figures 7.4 (right), 7.8 (right), 7.45 (right), 7.50 (right), 7.52; by ICD/ITKE/IIGS University of Stuttgart, figures 1.3 (left), 7.1, 7.6 (right), 7.33 (right), 7.35 (right), 7.36 (right), 7.37 (right), 7.39 (right), 7.43 (right), 7.48 (right), 7.49 (right), 7.62, 7.66, 8.8 (right) | **Kubail Kalousdian, N. & Lochnicki, G.**, ©2019, figure C.1 | **La Magna, R.**, ©2012 by ICD/ITKE University of Stuttgart, figure 6.16 (right) | **Li, J.-M.**, ©2014 by ITKE University of Stuttgart, figures 7.13 (right), 7.15 (right), 7.54 (bottom row), 7.58 | **Li, J.-M.**, ©2015 by ITKE University of Stuttgart, figures 7.5 (right), 7.7 (right), 7.11 (right), 7.12 (right), 7.14 (right), 7.26 (right), 7.27, 7.28 (right), 7.29 (right), 7.54 (top row) | **Locatelli, D. & Chen, T.-Y.** ©2020, figure C.2 | **Menges, A.**, ©2012 by ICD/ITKE University of Stuttgart, figures 6.10 (right), 6.44 | **Pirmann,**

D., ©2012, figure 4.1 (right) | **RaBoe/Wikipedia**, 2010, CC BY-SA 3.0 DE[5], figure 4.1 (left) | **Reichert, S.**, ©2014 by ICD/ITKE/IIGS University of Stuttgart, figure 7.68 | **RIBA Collections**, ©1958, figure 1.1 | **Sahin, E. S., Skoury, L. & Treml, S.**, ©2020, figures C.3, C.4 | **Schmitt, A.**, ©2014 by ICD/ITKE/IIGS University of Stuttgart, figures 7.41 (right), 7.42 (right), 7.64 | **Schwinn, T.**, ©2011 by ICD/ITKE University of Stuttgart, figures 6.28 (right), 6.31 (right), 6.33 (right), 6.34 (right), 6.52 | **Schwinn, T.**, ©2014 by ICD University of Stuttgart, figures 6.3, 7.3, 7.16 (right), 7.17 (right), 7.19 (right), 7.20 (right), 7.21 (right), 7.22 (right), 7.44 (right), 7.47 (right) | **Schwinn, T.**, ©2015 by ICD University of Stuttgart, figures 2.2, 7.23 (right), 7.38 (right), 7.56 | **Schwinn, T.**, ©2018 by ICD University of Stuttgart, figure 9.1 | **van der Steld, B.**, ©2012, figure 6.25 (center-right), 7.9 (center-right).

---

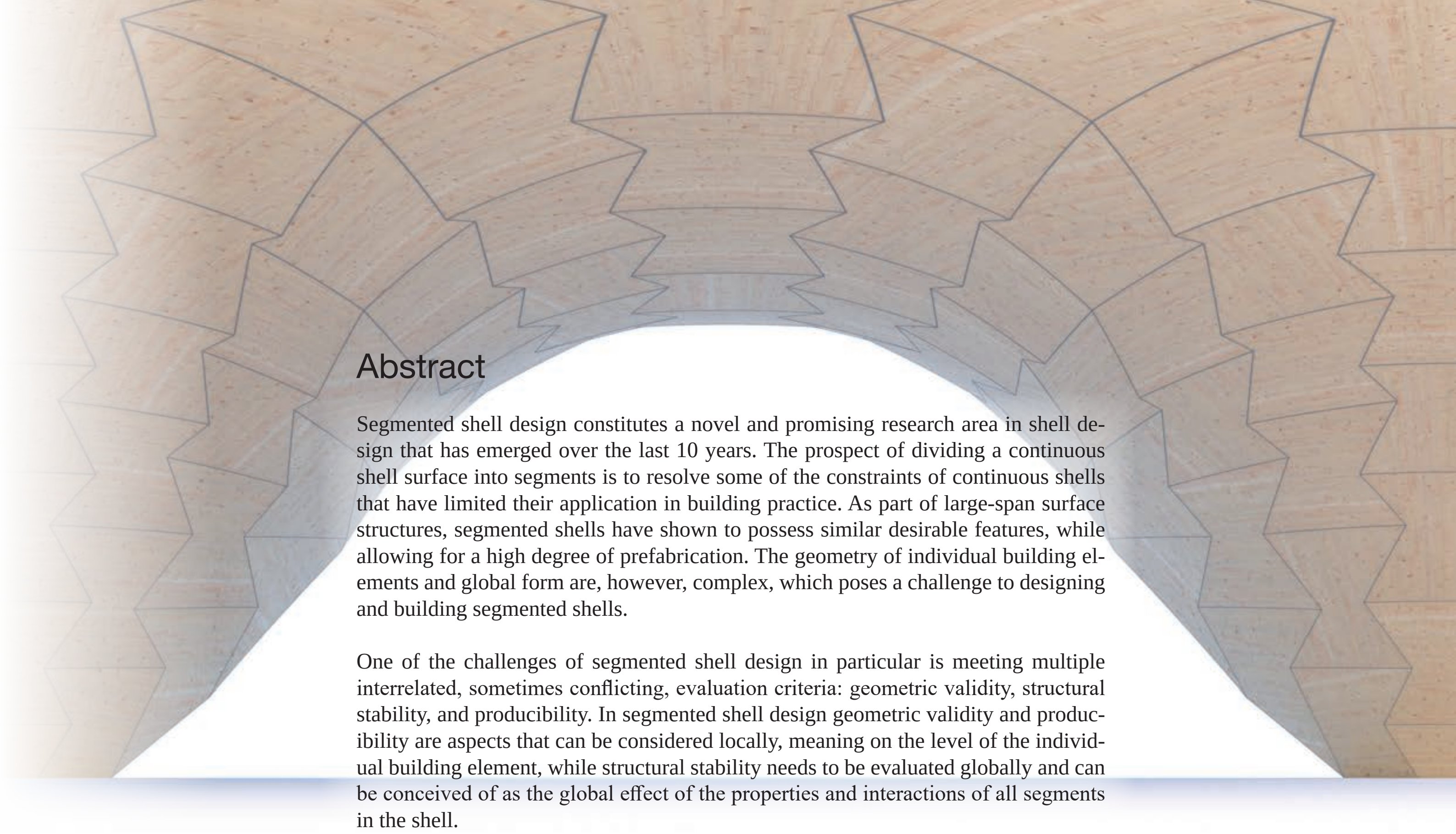
[5] https://creativecommons.org/licenses/by-sa/3.0/de/legalcode

Curriculum Vitae

Tobias Schwinn was born in 1977 in Marktredwitz, Germany. From 1998 to 2005, he studied architecture at the Bauhaus-University in Weimar, Germany, the École d'Architecture de Lyon, France, and at the University of Pennsylvania in Philadelphia as part of the US-EU Joint Consortium for Higher Education. He received a German engineering degree (Dipl.-Ing.) from Bauhaus-University in 2005.

Between January 2006 and December 2010, Tobias worked as a Senior Designer for Skidmore, Owings and Merrill in their New York and London offices applying computational design approaches to early stage architectural design in the areas of form-finding, rationalization, complex geometry, automation, and environmental design. In January 2011, he joined the Institute for Computational Design and Construction at the University of Stuttgart, Germany, as a research associate. He has been actively engaged in research and development of many of the widely published research demonstrators, most notably, the ICD/ITKE Research Pavilions 2011, 2012, 2013-14, 2015-16, and the Landesgartenschau Exhibition Hall. Since 2013, he investigated the use of agent-based modeling as a method of integration of potentials and constraints of robotic fabrication into architectural design focusing on the fabrication-oriented design of segmented shells. In 2016, he became one of the founding members of ICD's "Agent-based Modeling and Simulation" research group and in 2020 its leader. In 2018, he became a tenured lecturer (Akademischer Rat) and head of the institute's research and research infrastructure, which includes the institute's robotic fabrication equipment in the Robotic Prototyping Lab (Robolab) and the Computational Construction Laboratory (CCL) in Wangen. He has tutored master theses and taught pavilion design studios and seminars on the topics of computational design, digital fabrication, and behavioral fabrication in the University's international MSc program "Integrative Technologies and Architectural Design Research" (ITECH). He has been an invited studio critic at Columbia University's GSAPP, Harvard GSD, Pratt Institute, AA London, and CITA Copenhagen.

Tobias has both published and lectured internationally in the fields of robotic fabrication and agent-based modeling in architecture. His collaborative research has been recognized collectively through various awards, most notably the German Design Special Award in 2016, the Wilhelm-Klauditz Award for Timber Research and Environmental Protection, the Timber Construction Award of the State of Baden-Württemberg in 2015, the Pionieering Research Award of the Association for Robots in Architecture in 2014, and the Advances in Architectural Geometry 2012 Best Paper Award. Selected works have been exhibited as part of the traveling exhibitions "Out of Hand: Materializing the Postdigital" and "Anything goes! Die neue Lust am Material", the 10th Shanghai Biennale 2015, and the exhibition "Baubionik – Biologie beflügelt Architektur" at the Rosenstein Museum of Natural History in Stuttgart in 2016-17.



Abstract

Segmented shell design constitutes a novel and promising research area in shell design that has emerged over the last 10 years. The prospect of dividing a continuous shell surface into segments is to resolve some of the constraints of continuous shells that have limited their application in building practice. As part of large-span surface structures, segmented shells have shown to possess similar desirable features, while allowing for a high degree of prefabrication. The geometry of individual building elements and global form are, however, complex, which poses a challenge to designing and building segmented shells.

One of the challenges of segmented shell design in particular is meeting multiple interrelated, sometimes conflicting, evaluation criteria: geometric validity, structural stability, and producibility. In segmented shell design geometric validity and producibility are aspects that can be considered locally, meaning on the level of the individual building element, while structural stability needs to be evaluated globally and can be conceived of as the global effect of the properties and interactions of all segments in the shell.

Agent-based modeling and simulation (ABMS) provides the opportunity to bridge the gap between local characteristics and global performance. By focusing on the detailed description of the individual building elements and their interactions and by conceiving of the global form as the result of a myriad of local interactions of virtual agents representing building elements, the global design problem can be solved in parallel on the level of the individual building elements.

The work thus proposes a methodology for developing agent-based models of buildings where agents constitute building elements. The research pursues and synthesizes two investigative strands: on the one hand, generalizing findings from previously built plate shells as part of a case study-based, inductive research approach, which is geared towards building a catalog of validated design principles for plate shells; on the other hand, systematizing the agent-based modeling approach for architectural design-oriented applications in general, and plate shell design in particular.

Cover image:
Bi-lateral feedback loop. Tobias Schwinn I ICD University of Stuttgart I 2021

Submitted to the

University of Stuttgart
Institute for Computational
Design and Construction

ISBN 978-3-9819457-5-1