University of Stuttgart

ICD Institute for Computational
Design and Construction

ISBN 978-3-9819457-0-6

01

ICD Research Reports
ICD Forschungsberichte

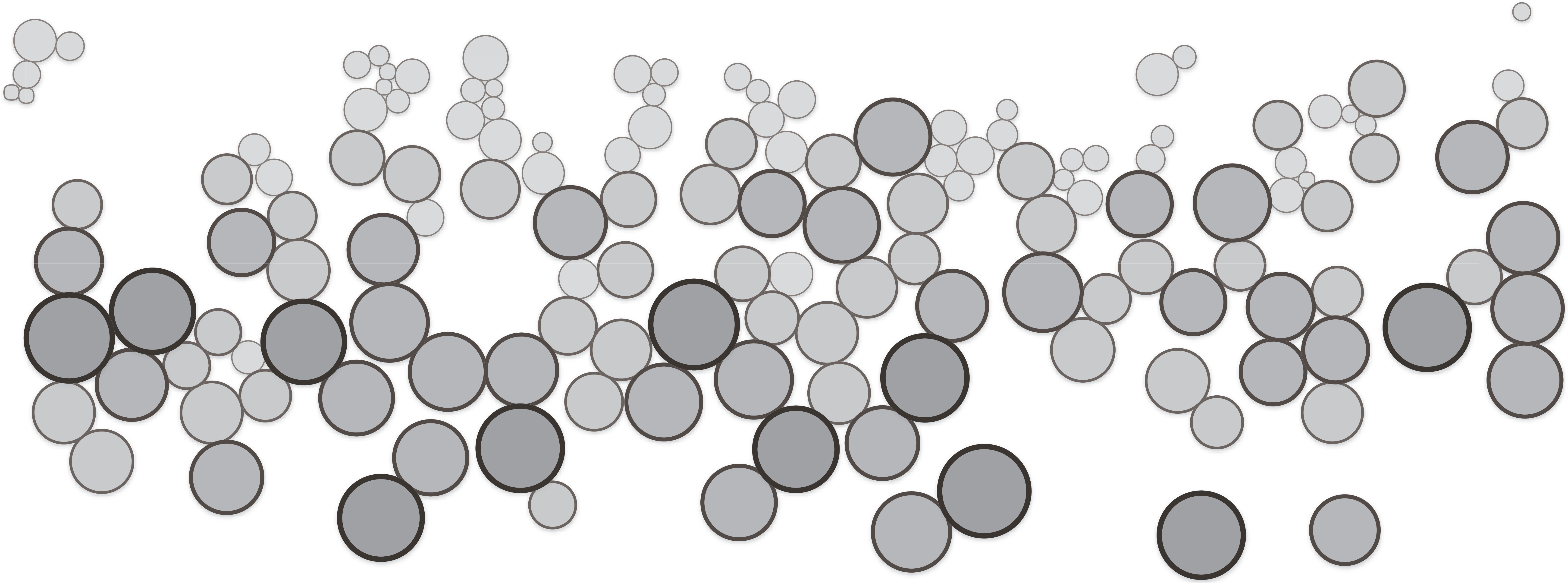
ICD 01

Ehsan Baharlou

GENERATIVE AGENT-BASED ARCHITECTURAL DESIGN COMPUTATION

Behavioral strategies for integrating material, fabrication and construction characteristics in design processes

Editor I Herausgeber: Professor AA Dipl. (Hons.) Achim Menges



ICD 01

Ehsan Baharlou

GENERATIVE AGENT-BASED ARCHITECTURAL DESIGN COMPUTATION

Behavioral strategies for integrating material, fabrication and construction characteristics in design processes

Editor I Herausgeber: Professor AA Dipl. (Hons.) Achim Menges

Generative Agent-Based Architectural Design Computation

Behavioral Strategies for Integrating Material, Fabrication and Construction Characteristics in Design Processes

von der Fakultät Architektur und Stadtplanung der Universität Stuttgart
zur Erlangung der Würde eines Doktor-Ingenieurs
(Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von
Ehsan Baharlou
aus Teheran

Hauptberichter:	Prof. Achim Menges
Mitberichter:	Prof. Dr.-Ing. Oliver Tessmann

Tag der mündlichen Prüfung 22. Februar 2017

Institut für Computerbasiertes Entwerfen und Baufertigung
der Universität Stuttgart, 2017

Abstract

The aim of this thesis is to investigate the generative potential of agent-based systems for integrating material and fabrication characteristics into design processes. This generative agent-based system reflects the significance of behavioral strategies in computational design and construction. This work presents a generative behavioral approach for integrating fabrication processes with material specifications. The development of a computational framework facilitates this integration via an agent-based system. A series of experiments with related case studies emphasizes behavioral strategies within the processes of formation and materialization.

This research proposes the integration of material and fabrication processes through an agent-based system. The utilization of this system reflects a theoretical framework in developing an integrative computational method. The implementation of this theoretical framework in practical studies demonstrates the applicability of this research. The practical developments highlight the importance of behavioral strategies to establish integral design computation.

Chapter 1 introduces the extended behavioral strategies to integration design. Chapter 2 provides a study about integrative design computation to abstract the main drivers of design integration through agent-based modeling. Chapter 3 presents agent-based systems in architectural design, specifically, in regards to material, fabricational, and environmental principles. Chapter 4 explores experiments and case studies to adjust the development of a generative agent-based system for integrating material and fabrication characteristics in design processes. Chapter 5 explains procedures for setting-up a generative agent-based design computation. Chapter 6 discusses the significance of behavioral strategies to develop different behavioral layers within a generative agent-based architectural design. Chapter 7 concludes the integral behavioral strategies by proposing trends to minimize the gap between formation and materialization through coalescing computational and physical agent-based systems.

Zusammenfassung

Ziel dieser Arbeit ist es, die generativen Potentiale von Agenten-basierten Systemen zur Integration von Material- und Fertigungseigenschaften im Entwurfsprozess zu untersuchen. Diese generative, Agenten-basierten Systeme spiegeln die Bedeutung von Regel- und Verhaltens-basierten Strategien für das digitale Entwerfen, Planen und Konstruieren wider. Die vorliegende Forschungsarbeit stellt einen generativen Ansatz zur Integration der Charakteristika von Material und Fertigung dar. Dies erfolgt über die Entwicklung einer digitalen Methode, die die Integration in ein Agent-basiertes System ermöglicht, was an einer Reihe von Experimenten und Fallstudien und der dazugehörigen Verhaltensstrategien für die Formgenerierung und Materialisierung erprobt wurde. Das operative Potential des theoretischen Rahmens wird in diesen praktischen Studien demonstriert und belegt die Anwendbarkeit der Forschung. Die theoretischen und praktischen Entwicklungen zeigen die Bedeutung von Verhaltensstrategien für das architektonische Entwerfen und einen ganzheitlichen digitalen Gestaltungs- und Bildungsprozess.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Achim Menges, for providing a unique research environment and for his continuous support. His guidance played a significance role on the development of this dissertation.

I would also like to thank Prof. Dr. Oliver Tessmann, the thesis committee, for his insightful suggestions and valuable comments, which open my research to the valuable aspects of design processes.

My sincere thanks also go to Prof. Dr. Jan Knippers from ITKE for his great advice and guidance during the course of this research.

To my friends and colleagues at the Institute for Computational Design and Construction (ICD), I owe you all your friendship and companionship that provided a unique and memorable research environment, and for all the fun moments we had together during this period. With special thanks to Saskia Frank and Britta Kurka for their administrative and organizational supports.

Much appreciation to the Diploma students Mark Baur, Georg Ladurner, Boyan Mihaylov, and Stefana Parascho; Master students Giulio Brugnaro, Gergana Rusenova, Djordje Stanojevic, Kenryo Takahashi, Maria Yablonina, and Julian Wengzinek for the enlightening contributions obtained from their research. To Sean Campbell, I am indebted and grateful for the correction of the final draft. Special thanks to Jasmin Sadegh and Omid Shekari for their efforts on correcting the initial draft. Sincere thanks to Alexandre Mballa Ekobena for his graphical work on the final presentation poster.

I am grateful to Juni- Prof. Dr.-Ing Jutta Albus for her friendship and her support.

Last but not be the least, I would like to express my deepest gratitude to my mother and father for their uninterrupted support in all the steps I have gone through, and my deepest gratitude goes to my brother who has supported me with his intellectual exchange that has enlightened this work.

Stuttgart, February 2017 *Ehsan Baharlou*

to my parents
and my brother

Contents

Abstract i

Acronyms xiii

1 Introduction 1

1.1 Preamble. Extended Behavioral Strategies to Integration Design 1
1.2 General Context of Design Integration 2
1.3 Behavioral Agencies 4
1.4 Overall Objectives 5
1.5 Dissertation Structure 8

2 Towards a Behavioral Paradigm for Architectural Design Integration 11

2.1 Inclusive Design Computation 11
2.2 Inclusive Paradigms of Morphologic Studies 23
2.3 Soft Design Computation 37
2.4 Agent-Based Design Computation 56

3 Current State: Agent-Based Systems in Architectural Design 77

3.1 General Areas of Applications 77
3.2 Categorization 78

4 Introduction of Experiments and Case Studies 89

4.1 Reflection of the Adaptive Behavioral Significances on Design Processes 89
4.2 Agent-Based Digital Morphogenesis for Plate Shell Systems 105
4.3 Fabricational Morphogenesis for Panelizing Free-Form Surfaces 120
4.4 Agent-Based Morphodynamic Generation of Fibrous System 144

5 Towards Generative Agent-Based Architectural Design Computation 165

5.1 Preamble to Generative Agent-Based Design Computation 165

5.2 Developing Agent-Based Models in Integrative Design 169

5.3 Developing Agent-Based Systems: Mapping Active Agencies into Agent-Based Systems 171

6 Behavioral Strategies for Inclusive Design Computation 187

6.1 Preamble to Behavioral Inclusive Design Computation 187

6.2 The Development of Generating Agent-Based Systems 189

6.3 Controlling Systems' Behaviors 192

6.4 Internal and External Behavioral Mechanisms 196

6.5 Intervening Mechanisms: Monitoring Systems 205

7 Conclusion and Discussion 209

7.1 General Review 209

7.2 Generative Agent-Based Design Computation 210

7.3 Behavioral Design Computation and Construction 213

7.4 Outlook and Trends 216

Glossary 221

References 225

List of Figures 243

Curriculum Vitae 255

Acronyms

ABM	Agent-Based Modeling
BIM	Building Information Modeling
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAM	Computer-Aided Manufacturing
CAO	Computer-Aided Optimization
CAS	Complex Adaptive System
CFD	Computational Fluid Dynamics
CGP	Constrained Generating Procedure
CNC	Computer Numerical Control
FEA	Finite Element Analysis
FEM	Finite Element Method
TPI	Tangent Plane Intersection

1 Introduction

1.1 Preamble. Extended Behavioral Strategies to Integration Design

Research on generative agent-based architectural design computation has focused on the behavioral strategies of integrating materials and fabrication characteristics to realize design processes. This study concentrates on a computational method for considering material and fabrication capacities and constraints within the process of integral formation and materialization[1]. The result of this fusion can further integrative design by manifesting constructible forms. The manifestation of form can be described through two different approaches, one constructional and the other theoretical. The constructional approach considers the process of materialization during the stage of formation, while in the other approach, forms are theoretical intensions that impose a formal structure on the materialization process. These two approaches determine at which point fabrication tools are considered during the process of design. The first approach deals with fabrication constraints during the process of formation, and the latter requires another process to overcome the deficiency of theorized forms and fabrication systems.

The behavioral negotiation between material organizations and fabrication processes requires a computational framework to provide an explorative model to study the emergence of forms. This research considers agent-based systems as a promising method to develop a computational design framework. This computational framework investigates adaptive procedures to adjust for arising complexities of behavioral integration within form manifestation. This research tries only to extend agent-based systems to the field of behavioral strategies. These strategies are then applied

---

[1] The materialization processes are considered as a general term in which it can be involved within the physical and virtual form manifestation.

for computational design and construction. This work presents behavioral integration via agent-based systems to provide ample support for integrating fabrication environments with material organizations. Behavioral integration describes the emergent features exhibited from micro interactions within agent-based systems. These micro interactions establish inclusive architectural design processes at macro-levels. In order to gain insight into the potential of this method of integration, the research is based on virtual experiments with supporting case studies and examples. The experiments also investigate the difficulty and complexity of behavioral integration. In a scientific manner, these experiments suggest that agent-based systems as adaptive processes are appropriate for adjusting the complexity of behavioral integration within design systems.

1.2 General Context of Design Integration

In recent years, integrated computational tools have been investigated to facilitate building constructions. It is necessary to study the significance of fabrication and construction procedures from the early stage of design processes to the final stage of construction systems. The integration within computational applications establishes comprehensive modeling systems to decrease the fragmentation of design and fabrication processes into cohesive approaches. A building database model, which designates a knowledge-based system, comprises of a notational means of organizing design applications. The notational applications represent top-down hierarchies of interconnected objects and instances from different class libraries of building database models. Utilizing hierarchical structures within parametric modeling systems furthers architectural design applications by developing seamless integrations between design and fabrication processes. Although, parametric modeling facilitates the integration of design processes, it is restricted to the early stage of design notations.

In the context of notational design, knowledge-based systems are associated with ultimate forms that include geometrical definitions, components, and elements. The subdivision of forms into small parts limits structuring the design database, where the final geometry cannot substantially be revised and only negligible changes can be tolerated. Thus, integrated design and fabrication, which is a knowledge-based approach, constrains designers to consider fabrication processes at the early stages of design. Otherwise re-designing becomes an obligatory procedure. In contrast, the integrative computational design and materialization strategy might benefit from developing new methodologies that closely resemble those of natural morphogenesis. Natural morphogenesis is adapted from morphologic studies, wherein biologists study morphology from theoretical and constructional perspectives. In this sense, the growth and development of form relies on intrinsic and extrinsic influences that are indicated via studies on morphology, specifically the concept of morphodynamics.

Implementing morphologic studies into computational frameworks allow engineers and architects to use computational morphogenesis, which is the process of computing the formation and realization of a design. The development of computational morphogenesis is compatible with system behaviors resulting from internal interactions between different layers of information. This method fosters computational frameworks to establish layers of information to model morphogenesis. The complexity of this framework cannot be simplified to a top-down system due to the emergent behaviors that arise out of interactions among low-level elements. Therefore, computational morphogenesis investigates the linkage between two levels of micro and macro interactions to provide the self-organization within intrinsic constituents for exploring emergent properties. This level of organization facilitates the mechanisms to overcome and to present the adaptabilities of behavioral complexities. Hence, computational morphogenesis as an inclusive method of design computation is intimately related to the adaptation or learning procedures. Accordingly, computational morphogenesis as a complex adaptive system can be investigated under behavior-based approaches through which this process includes basic characteristic features of material and fabrication systems along with their associated behavioral rules.

As a general term, classifying fabrication environments into three categories of fabrication, assembly, and construction systems relies on material organizations and the way that material properties are considered in the fabrication environment. The material dependencies within the fabrication environments allow for further investigation on analytical and heuristic methods to coalesce fabrication machineries into the computational design. Concurrently, developing an inclusive computational framework facilitates mediations between material systems and fabrication tools in which their negotiations should include all their properties and capacities. Accordingly, the computational tools that facilitate behavioral negotiation between different parties are intermediary systems utilized with programmable building blocks or agents. In this context, agents as distributed computation units can establish relationships among separated drivers through behavioral mediations. Architects with agent-based modeling and simulation tools can generate constructional forms where the form generations are simultaneous with the process of materialization. Synchronicity of this process allows constructional forms to benefit from the constraints and capacities of both fabrication tools and material systems by considering the ultimate possibilities of fabrication tools. Utilizing agent-based systems with different methods and techniques enables simulation and modeling the materialization process with form generation. This method of materializing forms is far beyond realizing an individual designed form through a customized method.

1.3 Behavioral Agencies

Generative agent-based architectural design computation is a behavioral system that relies on establishing rules and procedures to exhibit behaviors within design systems. Behavioral agencies of this generative tool arise from computation units that are informed with abstracted data. The computational units as autonomous or semi-autonomous agents can act upon simple embedded rules. The rules of agents' behaviors are derived from properties and capacities of material and fabrication drivers. Accordingly, rule-based agent systems can merge these separated drivers together, wherein their behaviors are based on their embedded rules and their contextual environment.

Consequently, the behaviors of agents emerge from interactions with other agents and their related environment. The complexity of these behaviors necessitates the investigation of inhibiting and coordinating mechanisms. These mechanisms may provide effective control over agents' behaviors to exhibit emergent complexities that are adapted to design requisitions. The complex adaptive system that results from behavioral negotiations among agents and their environment are parameterized to explore system lever points. The lever points indicate the fulcrum position that the behavior of system changes from one state to another. Abstracting materials and fabrication features into this framework requires a transition function to virtually materialize the target design. Gaining insight on the rules and parameters required to develop a formal computation framework. The coalescence of materializing processes into the design computation is the primary focus of this dissertation. This investigation also follows the self-organizing properties of agents' behaviors to exhibit the emergence of the fusion of materialization and formation.

Concisely, generative agent-based systems postulate that the development of a modeling system requires research on the morphological definition of agents and its relationship to environmental and fabricational factors. From this perspective, the different agent morphologies correspond to the level of participation in fabricational morphogenesis. Accordingly, agents are part of morphogenetic processes or factors of developing the processes. Thus, the generative system is developed to cover the common properties of both types of agents. In addition, the fabricational movements that are recognizable as material agencies and fabrication agencies are interpreted into several rules and behaviors, denoted with vector-based systems to prepare controlling platforms. Therefore, the generative tool relies on several classes of agents, environments, and behavioral rules, which are under the control of system behaviors. These classes are parameterized to deal with differences between various types of agents, their rules, and behaviors.

1.4 Overall Objectives

1.4.1 Development of a behavior-based approach through a generative agent-based design computation

Of interest is the development of a generative agent-based system, where the system is associated with generating methods, such as Constrained Generating Procedures (CGP's), to generate limited alternative solutions for coalescing fabrication and material constraints into design processes. Developing this generative agent-based system requires significant study of material characteristics, fabrication constraints, and environment effectiveness within computational design and construction models. Furthermore, the generating methods will emphasize behavioral insights within these computational tools. The focus on behavior-based methodologies may enable designers to define relationships between design processes and fabrication techniques specifically considering computer-aided design and computer-aided fabrication.

Accordingly, this research investigates the potential of behavior-based approaches to expand the design space, which previously was unavailable for designers restricted to top-down processes. Due to the disregarded limitations of material and fabrication at the early stage of design, a design space is limited to the adjustment between given forms and construction processes. Therefore, the development of a behavioral model could reduce the prominence of post-rationalization in design. Informing design computation models about constraints will enhance the generative behavioral model to effectively interrelate the design and fabrication processes. Thus, recognizing the effectiveness of constrained generating models requires modeling a behavioral system, which allows to integrate behavioral approaches within design processes. Accordingly, utilizing agent-based systems promise to not only consider all constraints within the design process, but also adapt generative systems to arising complexities.

The significance of agent-based systems is of interest to model and simulate complex behaviors that arise from merging material and fabrication strategies into non-standard design procedures. The consistency within this behavioral system requires research on the different levels of communication and interactions, through which synchronicity of amalgamating several factors allows system maintenance within its critical adaptive states. Therefore, this research furthers agents' behaviors with internal and external mechanisms to signify the intrinsic relation between material properties and fabrication tools as fabricational morphogenetic factors, while this relation is under influence of external factors, such as environmental effectiveness. It also conveys essential knowledge to each agent for behaving within information spaces, such as a hyper-dimensional morphospace. This access to the information space leads agent with adaptive behaviors toward constructional regions, conducted via developing a hypothetical morphospace.

1.4.2 Development of an inclusive design computation through the investigation of design agencies

Agent-based modeling techniques necessitate an investigation of appropriate methods for modeling and simulation that are significantly associated with generative approaches. In the sense of a high-level of integration, the study of a generative agent-based system provides the development of a new integral computational platform to coalesce formation and materialization. Integral computation requires the inclusion of behavioral parameters that are interpreted from material properties and fabrication characteristics to provide constructive interactions. Abstracting and transferring behavioral parameters to agents' structures and mechanisms require an investigation on the influential factors obtained from a comparison between morphological studies and integrative design architecture.

In detail, the inclusive drivers will be established through investigating morphologic studies to understand the relation between theoretical morphology and constructional morphology. After that, comparing these drivers with architectural design aspects, specifically integrative architectural design, will project the importance of biological factors to the architectural and engineering principles. Hence, the inclusive drivers will try to overarch between morphologic studies and architectural design by adopting biological factors into design systems. This adaptation requires the prioritization of each factor within design procedures. In the sense of construction movements, the inclusive design computation involves developing mechanisms to implement materials and fabrication tools within the design processes. Accordingly, developing an abstract model to effectively merge these mechanisms will rely on correlation between fabrication tools and applied materials. This process will postulate the development of a theoretical morphospace, which will consider the geometrical, functional, and developmental principles of association between material systems and fabrication tools.

Abstracting material properties will allow for the consideration of geometrical characteristics and behaviors. This abstraction will help in the investigation of the effectiveness and versatility of geometrical definitions for the fabrication tools. In addition, investigating fabrication tools will further the development of the theoretical morphospace, where geometric constructability of the materials is influential in developing the theoretical morphospace. In both of these investigations, material and fabrication systems should consider the time-consuming computation to the level of abstraction and the abstracted properties, such as avoiding excessive abstraction of inessential material behaviors.

This research presents a behavioral method for integrating fabricational constraints into the design process. This method will establish the integral relation between formation and materialization, which will be achieved through behavior-based approaches, such as agents or building blocks, instead of knowledge-based approaches, such as the conventional integrated design methods. Agent-based systems will be developed with the constraints and capacities of the material and fabrication systems. These constraints and capacities will reflect the theoretical morphospace upon the realization of design processes. In other words, this method informs a genitive process, which is obtained through developing a system of agents, in return, the generating mechanisms are constrained by the morphospace fabrication tools.

1.5 Dissertation Structure

In Chapter 2, Sections 2.1–2.3 provide comprehensive studies on the backgrounds and ramifications of integrative design computation to synthesize inclusive drivers for design integrations. Section 2.4 introduces the computational method of agent-based modeling to represent modeling and simulation techniques within the informal and formal systems. Chapter 3 reviews the related works in architectural design to categorize the use of agent-based systems within architectural design processes. Chapter 4 describes four experiments and their related case studies to explain the development of generative agent-based systems for integrating material and fabrication aspects into design processes. The development of a generative agent-based system for integrative design computation requires a solid framework to adopt the behavioral integration with design processes. Therefore, Chapter 5 introduces a discussion about methods and procedures to develop generative agent-based design computation. These methods and procedures were then applied within the case studies and experiments. In addition, the development of the generative agent-based design computation that coalesce inclusive drivers of architectural design requires to investigate on behavioral strategies. Through behavioral strategies, Chapter 6 discusses and analyzes the development of different layers of behaviors, which were experienced when developing generative behavioral systems.

According to §2.4 of the University Stuttgart Dissertation regulations, the findings of this dissertation have been pre-published selectively. These papers have been indicated in the references chapter with relevance to the main text.

2 Towards a Behavioral Paradigm for Architectural Design Integration

2.1 Inclusive Design Computation

2.1.1 Approaching the concept of integration in architectural design

Preamble to integration in architectural design

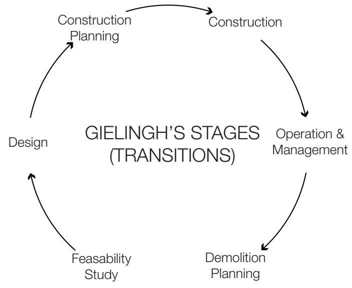


Fig. 2.1.1: The Gielingh's classification of the building lifecycle; redrawn by author based on: Eastman (1999, p. 6).

In the building industry, the theoretical foundation of architecture relies on a dichotomy between Brunelleschi's and Alberti's legacies; the former values the architect as a master builder, while the latter values the architect as a notational designer (Carpo 2011, p. 16). Mass production in the industrialization of architectural design uses "notational systems" to standardize traditional architecture, with an emphasis on distinguishing between design processes and design construction. In particular, the concept of notational systems differentiates architectural design from other sectors, such as engineering and construction. In the context of the building industry, standardizations are followed by segregating different sectors of design, fabrication, and construction to increase productivity and reduce production times and costs. According to Eastman (1999), one approach of this standardization is Gielingh's phases (Figure 2.1.1) through which the building lifecycle is classified into the "feasibility study," "design," "construction planning," "construction," "operation," and "demolition planning" (Eastman 1999, p. 5).

Growing distances between newly-established architectural sectors and other sectors, such as material studies, engineering, and construction, excludes architects from the process of materializing their design. It limits architects to develop only design documentation with limited knowledge of building materials and construction techniques. In contrast, Kieran and Timberlake (2004) noted that other disciplines, such as the "automotive, shipbuilding, and aircraft" industries blurred the lines between design and making. Designers and makers advanced into different sectors, but for solving a problem, their intelligence coalesced into one system (Kieran and Timberlake 2004, p. 13). In the context of the building industry, the lack of accessibility of materializing knowledge in the design process has promoted the development of building modeling applications. Figure 2.1.2 illustrates the schematic idea of developing a model to facilitate collaboration and communication. Communicating between the sectors of design and construction requires the development of a standard method for exchanging design data. The use of computers promises the establishment of a standard platform for designers and manufacturers to collaborate actively throughout the entire process of building.

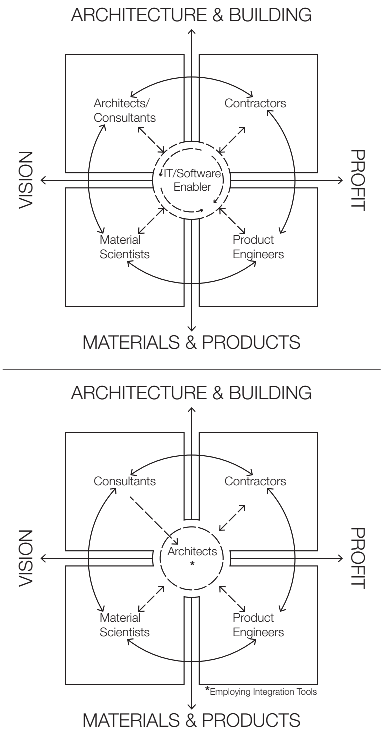


Fig. 2.1.2: Kieran and Timberlake (2004) propose the development of a model to manage the knowledge of different sectors to reinstate architects as a "twenty-first century maestro," rather than a "master builder"; redrawn by author based on: Kieran and Timberlake (2004, pp. 12-22).

The documents used in production and drafting can particularly benefit from CAD[1]/CAM[2] applications, including different standard classes to provide data transition between designers and manufacturers. Transferring design information along with data-management systems prepares the technology for developing a building database. The background of developing a building database goes back to 60's, when scientists tried to provide simple ways of communication between human and machine, such as the Sutherland's Sketchpad (Sutherland 1964). The development of geometric building blocks coincided with the advancement of computer science wherein programming languages extended from *procedure-based programming* to *object-oriented programming*. In this sense, Björk (1989) argues that structuring a building database is related to developing a knowledge-based system and object-oriented modeling techniques (Figure 2.1.3). Elements for the entire building are abstracted into spatial systems, materials, and parts; these abstracted elements are then organized as objects under specific standardized classes (Björk 1989, p. 73).

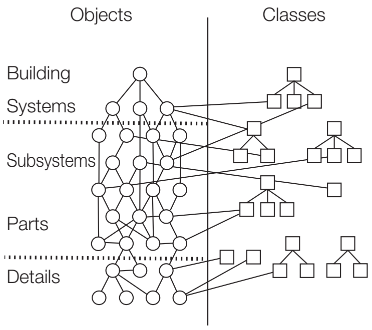


Fig. 2.1.3: Relating building elements to object-oriented programming; redrawn by author based on: Björk (1989, p. 73).

In the context of architecture, engineering, and construction (AEC), the development of a building modeling system has been associated with several applications to facilitate the modeling database for exchanging information within the building industry. In the context of integrating design and construction, several applications have been developed, for example, CIC (Computer-Integrated Construction), CMB (Computer Models of Building), VDC (Virtual Design and Construction), and VBE (Virtual Building Environment) (Penttilä 2009, p. 464). However, the most coherent building modeling system is Building Information Modeling

---

[1] Computer-Aided Design.

[2] Computer-Aided Manufacturing.

(BIM), which is established through a comprehensive integrated knowledge-based system. Corresponding to the development of a building modeling system, data-exchange systems have also been advanced to provide a single approach to standardize a knowledge-based system. Approaches like STEP (Standard for the Exchange of Product Model Data), IAI (International Alliance for Interoperability), IFC (Industry Foundation Classes), and RUCAPS[1] obtain data-exchange platforms to manage access and edit integrated information in BIM software (Holzer 2007).

Building modeling systems consist of hierarchical knowledge-based systems that rely on preliminary developed design. The developed design establishes an ultimate form at the top level of the hierarchy, from which the building modeling systems are organized in accordance with that ultimate form. The ultimate form, as a design intention, determines interdependence of geometrical characteristics with their parts, components, and elements (Figure 2.1.4). Accordingly, building modeling systems are robust in response to minor design modifications, but they are fragile against extensive design alterations, which are mostly caused by neglecting construction levels. Late consideration of material and construction processes might impose major changes on the pre-developed design. This consideration demands early collaboration among construction sectors.

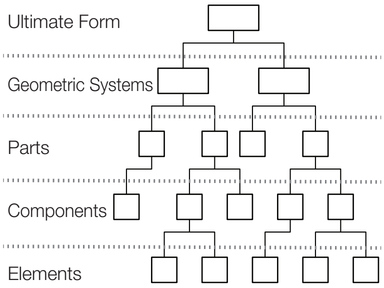


Fig. 2.1.4: Top-down relation between ultimate forms and constituents' elements.

Therefore, considering material and fabrication as individual drivers, which are active in the early stages of design, necessitate an investigation on methods of integration. In contrast to the top-down integration system, developing a framework that constitutes the basis for these active drivers requires to structure a bottom-up approach, which generates a form out of their amalgamation. The amalgamation among these segregated stages requires a comprehensive insight into design processes, where the realization and generation of form is coalesced together. The synthesis of material, constructional, and environmental factors together produces efficient, constructible, and adaptive forms that are effective solutions for an inclusive design process. The development of this process requires an investigation on existing methods of integration within architectural practices.

Integration in architectural design practice

Linear processes of integration between different sectors of the building industry require "access and incorporation of appropriate

---

[1] Really Universal Computer Aided Production System.

data, interpretation of results and possibly iterative use and exchange with other members of the building team" (Eastman 1999, p. 6). Building information modeling establishes a basic framework for the exchange of data among architects, engineers, fabricators, and constructors. This notational framework facilitates collaboration among different sectors. Therefore, the structure of building information modeling has a great role in communication between different sectors in the building industry. The National BIM Standard (NBIMS) defines building information modeling as "a digital representation of physical and functional characteristics of a facility" (NIBS 2007, p. 21). Similar to the manufacturing industry, BIM represents the building as a virtual computer model, instead of simple two-dimensional drawings (Eastman 2009, p. 16). By virtually modeling the buildings rather than representing them through rigid geometries in CAD systems, it extends information modeling to "object-based parametric modelling," in which geometrical organization is associated with non-geometrical qualities (Mitchell 2009, p. 13).

The top-down hierarchical framework links the early stage of design with the last stage of construction by connecting different parties to enhance virtual design developments. Accordingly, involving various sectors in building modeling systems reduces the gap between designers and fabricators. Preliminary design that is located at the top level of the hierarchy is accessible to engineers, fabricators, and constructors. According to the hierarchical model of organizing sectors, each sector will gradually be informed about the design information to evaluate stability, constructability, production costs, and time. Therefore, all sectors are directly involved in the process of design and they gradually develop the design documentation for the construction phase.

Even though it uses knowledge-based systems, BIM is deficient in transferring mathematical design parameters, such as material properties and environmental characteristics, where behavioral characteristics cannot be fully reduced and formalized to mathematical modeling parameters (Willis and Woodward 2010, pp. 188-189). Although BIM facilitates communications among different sectors, its deficiencies in materializing non-standard design requires a strategy to consider material characteristics and fabrication features. To overcome these deficiencies, BIM-based workflow also classifies building components into three categories: "made-to-stock" (MTS) components, "made-to-order" (MTO) components, and "engineered-to-order" (ETO) components (Eastman et al. 2011, pp. 305-308). The components in the first two categories are mass produced for general use, while the components in third category are customized for specific purposes (Eastman et al. 2011, pp. 305-308).

In addition to these methods, materializing non-standard designs require mediators to rationalize the given designs and prepare the fabrication documents. For example, Gehry technology has developed a digital process for rationalizing non-standard designs (Shelden 2002). This process was used in the preparation of accurate construction and fabrication documents for the Guggenheim Museum Bilbao (1991-1997) (Lindsey 2001; Shelden 2002). The Guggenheim Museum is an early example how digitizing the design process with CAD/CAM applications can improve communication between designers and builders.

Embracing digital form representation, without considering fabrication processes, requires post-rationalization. Post-rationalization acts as a bridge between digital design and physical construction, which proves the benefits of digital fabrication tools within architecture. At first glance, the process of digitizing design and construction seems to be fully integrated through the process of rationalization. However, digital design and fabrication are limited by the processes of rationalization and materialization. The requisite for the intermediary processes of rationalization and materialization stems from the widening gaps between design and construction.

The development of information modeling ensures that architects have access to all aspects of construction, while at the same time fabricators are incorporated early in the preliminary design phase (Kolarevic 2008, p. 655). However, active collaborations between architects and fabricators require a novel method that develops their collaborations without any hierarchical organization, while supporting dynamic collaborations and integrations between the design and fabrication phases. Kolarevic (2008) categorizes the integration of design processes into “integrated design,” which is a well-organized system to connect different sectors of the building industry, and “integrative design,” which is an open system that is receptive to implement idea, processes, and techniques for further developments (Kolarevic 2008, p. 656). According to Kolarevic (2008), integrated design is a linear process with top-down strategies in a closed system that is limited to the initial design setup. In contrast, integrative design, which is an open system, might resist any kind of linearization to ensure integration.

Integrative design computation: Development and setup

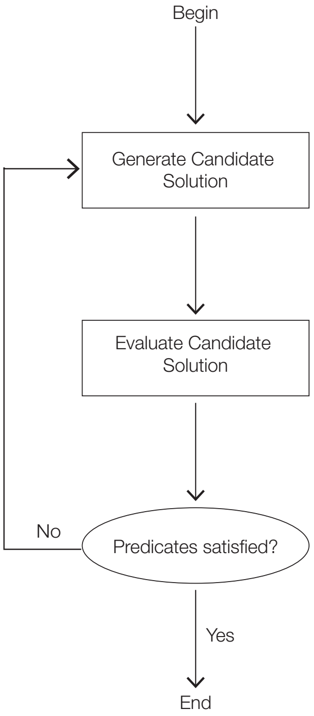


Fig. 2.1.5: The schematic diagram of the design computation process; redrawn by author based on: Mitchell (1990, p. 180).

The development of a computational framework to establish generation, simulation, and modeling of design could advance the integration of design and fabrication into one general system. Mitchell (1990) proposed a computational framework based on trial and error mechanisms to solve design problems (Figure 2.1.5). This framework coupled a generative mechanism with a test and a control mechanism to assess whether the generated alternatives were worth further consideration (Mitchell 1990, p. 179). In addition, identifying the best solution requires a strategy based on performance criteria, materialization sequences, and assembly aspects of developed productions to enhance the generated design (Klinger 2008, p. 28). In the context of "morphogenetic design," Hensel and Menges (2008) suggest that a computational framework is gradually informed by different aspects of materialization, such as the attributes and limitations of materials, fabrications, and assembly procedures (Hensel and Menges 2008, pp. 56-57). A computational "morphogenetic design" that is manifested in the integration of formation and materialization suggests an integrative design computation method (Menges 2013, p. 28). This method may produce benefits by implementing different computational design and engineering techniques. Therefore, integrative design computation combines segregated applications, such as CAD[1] (geometric logics), CAM[2] (constraints and procedures of manufacturing), CFD[3] (analyzing performative criteria), and FEA[4] (analyzing structural properties) within a computational framework (Menges 2008, p. 198). In addition, the development of integrative design computation relies on an open framework where material organizations are intertwined with fabrication environments. Moreover, evaluation mechanisms within this framework assess the structural and performative properties of a generated design. The formation process in this framework relies on the recognition of critical elements, which actively participate in the process of form generation. Developing an effective integrative application redefines the manifestation of form within the design context.

---

[1] Computer-Aided Design.

[2] Computer-Aided Manufacturing.

[3] Computational Fluid Dynamics.

[4] Finite Element Analysis.

2.1.2 Dissociation between the process of formation and materialization

Preamble to the manifest of form

> "The manifest form–that which appears–is the result of a computational interaction between internal rules and external (morphogenetic) pressures that, themselves, originate in other adjacent forms (ecology)." (Kwinter 2008, p. 147)

In his seminal book, *Notes on the Synthesis of Form*, Alexander (1964) conveys that design is a "process of inventing physical things which display new physical order, organization, form, in response to function" (Alexander 1964, p. 1). Accordingly, the design process is analogous to the invention of physical forms by which materialization and realization necessitates innovative methods for responding to functions. In this context, the design process is related to the form-function dichotomy, where the form is a general description of a physical object, influenced by various functions. According to Hensel (2010, pp. 42-43), the dichotomy between form and function has historical roots within architectural design, as Sullivan (1896) states "form ever follows function" (Sullivan 1896, p. 408). Function can be interpreted in two different ways: the external function is imposed on the form, while the internal function arises out of the form. Considering form as a passive result of employing functions within the design process claims that the constituent elements of the form act as obedient structures without any internal influences on the process of formation. Therefore, the form is acquiescent to the commanding functions that can be imposed from external elements.

Defining functions and embedding them into the design process has its own difficulties. The form will follow the consequences of embedding appropriate or inappropriate functions within the process. Recognizing the importance of each function will lead the design process to manifest desired forms. In this sense, the desired form requires the development of a method that explores design problems to find substantial solutions. The design processes fall within different layers that might be considered within various contexts. The evolvement of forms is subsequent to properly describing and encoding related layers into design problems. The design problem correlates forms to the process of recognizing "contexts," where "the form is the solution to the problem; the context defines the problem" (Alexander 1964, p. 15). Discerning the design problem within a specific context encourages appropriate methods for solving identified problems. Therefore, a critical evaluation of the design process informs a clearer context, where there is a distinction between influential and insignificant layers. Identifying influential layers determines a well-established context that clearly identifies the problem, which will in turn lead to an appropriate solution.

D'Arcy Thompson (1942) argues that form is a "diagram of forces" that are exerted on physical objects (Thompson 1942, p. 16). Accordingly, the design factors embedded within the context can be described through a diagram of forces, where each one of the contextual factors is mapped to the specific force. Vibrations of these forces modulate the physical objects to explore possible solutions for design problems. However, an inaccurate encoding of design information will generate unwanted results that are directly related to the inexplicit definition of design context and the physical objects. The diagram of forces, which are imposed upon the physical objects, effectively shape the process of formation and after accomplishing this process, this diagram maintains form's conformations (Thompson 1942, p. 16). The organization of contexts utilizes a diagram of forces to retain the distinction between internal and external boundaries of objects. Accordingly, boundaries are described as "active zones" that facilitate interactions between different levels of energy (Addington and Schodek 2005, pp. 7-8). Failures in the retention of boundaries lead forms to lose their consistency and robustness, until they gradually dissipate into the context.

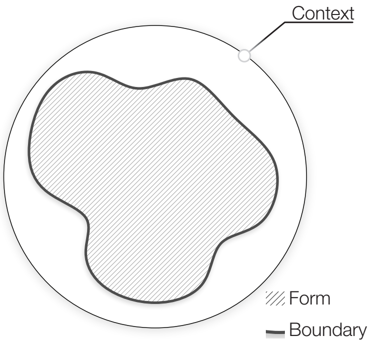


Fig. 2.1.6: A schematic diagram to define relation between form and context with defined boundary.

Bilateral relationships between internal and external objects recognize the association between its form and the intrinsic organization of its characteristics. These internal organizations and external contexts explore the design problem to improve the "fitness" level (Alexander 1964, p. 18). Identifying the internal characteristics and external factors of an object is one of the main issues of design processes. The internal organization consists of constituent elements that have diverse properties and capacities. This organization is surrounded by design contexts, which manifests form out of physical objects (Figure 2.1.6). One of the integral elements of this design process is the material's properties and behaviors. Considering material during the process of form generation will separate the philosophy of design into two material systems, one passive and one active. According to DeLanda (2001), forms, in the passive system, are imposed by "concepts" on isolated *homogeneous materials*. In the active system, forms evolve out of applying *heterogeneous materials* as dynamic elements (DeLanda 2001, p. 132). The consideration of both the homogeneous and heterogeneous properties of materials is essential in both philosophical points of view. Especially in the latter where the materials are expected to evolve in response to the external contexts with coordination from internal fitness criteria. Identifying materials as active drivers within the process of formation initializes the hidden capacities of materials to emerge as forms from their morphogenetic movements.

Morphogenetic movements rely on internal fitness criteria to coordinate the interaction between internal forces and external influences. Considering a material agency in design processes requires an understanding of material characteristics and methods of applying them to generate and realize form. Form realizations entail novel approaches to reflect a knowledge of manufacturing as a fabrication agency during the development of form. Reflecting a fabrication agency involves a design process with two levels of rigid and flexible systems from which these systems emphasize the dichotomy between hard and soft systems.

In the context of soft systems, Deleuze and Guattari (1987) speculate that the hybrid relation between "nomad space" and "sedentary space" to contextualize the significance of "matter-movement" in the "machinic phylum," where the phylum is "always *connected* to nomad space, whereas it *conjugates* with sedentary space" (Deleuze and Guattari 1987, p. 415). Accordingly, generating form out of "matter-movements" relies on the cognitive knowledge of fabricators to comply with the rogue state of matters. These cognitive approaches are a set of practical knowledge that is gained from artisans, when they are applying tools and materials to fabricate artifacts, such as a blacksmith (DeLanda 1997; DeLanda 2001, p. 133). Formalizing the cognitive knowledge of artisans within the design process allows the development of a machinic structure to deal with different aspects of design. Accordingly, the design process is accompanied by assembling several agencies through which machinic approaches facilitate their amalgamation.

In accordance with developing a machinic system, the cognitive knowledge of fabricators could be considered as an internal set of rules for hybridizing agencies between active material systems and soft fabrication procedures. The internal rules require a framework for computing nonlinear integrations of material and construction behaviors by which this platform should be extended to integrate different agencies of design processes. The institution of this framework is based on embedding cognitive approaches within form materializations. Accordingly, form generations are accompanied with the cognitive system that has knowledge of materials and their fabrication procedures to evolve forms within specific time intervals. Kwinter (2008) elaborated this approach as “algorithmic formalism,” which was originally conceived by Goethe (Kwinter 2008, p. 147). Algorithmic procedures convert internal rules that are qualified with cognitive systems to generative computational methods. These generative procedures correlate design principles with the intrinsic and extrinsic material properties identified by cognitive knowledge about construction systems.

Extended design methodologies to materializing form derivations

The processes of morphogenesis, which are studied in the biological sciences, introduce the simulation and modeling of morphogenesis to architectural design. The processes of morphogenesis, in natural and simulation environments, encourage architects to recognize growth and the development of organic forms as alternative approaches to derive forms. Natural formation, which considers the interactions between constituent materials and their physical and chemical properties, self-organizes their interdependencies into a coherent system. In this context, the conventional top-down design approach is replaced by a bottom-up approach where material entities are coalesced into construction procedures. In architectural design, the top-down approach is recognized as a “form-making” approach that applies materials for representing forms, while in the bottom-up approach materials significantly contribute to the process of “form-finding” (Kolarevic 2003, p. 13; Leach 2009, p. 34). Therefore, materials with intrinsic and extrinsic behavioral characteristics organize active systems within the process of form-finding by which these processes are contingent to a material agency. The development of form is affiliated with fabrication tools to entrench forms in material systems. The mutual correlations between fabrication tools and material systems steer generation of forms toward optimal equilibrium states.

In physical form-finding, material systems that are comprised of mechanical and chemical properties are engineered to yield optimal forms. Antoni Gaudi, Frei Otto, and Heinz Eisler studied the self-organizing properties of material systems to integrate the structures and properties of materials into architectural form (Hensel 2010, p. 46; Weinstock 2010, p. 144). Their investigations included several studies that incorporated suspended chains, soap bubbles, and ice structures to develop novel methods of integrating material organization within design processes (Otto et al. 1995; Chilton 2000). Since these form-finding methods are derived from material characteristics, the structural properties of materials arrange internal organizations to optimize the derived forms. Furthermore, the process of form-finding allows designers to modulate the material system through fabrication tools and to develop desired forms out of the materials self-organization properties. The negotiation between design intentions and material systems results in forms, where physical or structural characteristics emerge out of material self-organization.

Synthesizing forms within virtual environments requires an understanding of formations as algorithmic procedures. Embedding these algorithms within a computer system deals with series of binary codes. According to Terzidis (2003), encapsulating forms in binary systems is accomplished through "computerization" and "computation." Computerization employs computers to digitize and encode defined forms into the digital realm, while the computational approach deals with the process of form generation (Terzidis 2003, p. 69). In general, the digital representation of architectural forms inherits the conventional methods of architectural drawing, accompanied with drafting and layering procedures. This conventional method is embedded within computer-aided design (CAD) applications. These applications are developed to assist designers with design interfaces for digitizing their conceptual forms and ideas.

In design computerization, the processes of formation are drastically reduced by the syntax of geometrical characteristic that are embedded within the digital interfaces of CAD applications (Menges 2010, p. 331). In contrast, computational design provides a basis for manifesting forms as a bottom-up process. In general, computation employs algorithmic processes that are associated with mathematical and geometrical logics. The associative and algorithmic logics provide a bottom-up exploration of the different aspects of formation. Accordingly, computational design thinking relies on generative algorithms to develop forms from different layers of rules and procedures, such as "shape grammar" with specific rules to generate three-dimensional digital shapes (Stiny and Gips 1971, p. 125). In this context, the process of morphogenesis is a development of generative algorithms for simulating and modeling the growth patterns of natural organisms, such as cellular automata (Von Neumann 1958), L-systems (Lindenmayer 1968), Conway's game of life (Gardner 1970), and boids algorithm (Reynolds 1987). These generative algorithms are entirely based on the logical description of behavioral or geometrical representations without being influenced by other factors, such as material properties, constructional attributes, or environmental criteria.

In the context of natural morphogenesis, the process of formation emphasizes the chemical and physical properties of material organizations. In nature, growing patterns are the result of environmental factors that externally and internally affect morphogenetic movements. Similar to natural morphogenesis, the process of physical form-finding emphasizes appreciating material systems from the perspective of a bottom-up design approach. In contrast, Menges (2008) argues that digital morphogenesis segregates the process of materialization from the process of formation; thus, the generated forms impose on the material systems. Rationalization attempts to reconcile the process of materialization with process of formation (Menges 2008, p. 196). Due to natural morphogenesis, computational procedures are expected to benefit from a digital materialization method in which materialization is encoded as an active driver within the digital formation. Otherwise, distinguishing form generation from materialization requires another phase to impose digitally derived forms to the materialization processes. Accordingly, the process of formation should consider materialization as an embedded process that consists of both material and fabrication agencies within its own morphogenetic developments.

2.2 Inclusive Paradigms of Morphologic Studies

2.2.1 Preamble to the theory of morphology and morphogenesis

From 1786-1788 Goethe investigated “plant metamorphosis” (Goethe 1989). In this investigation, Goethe determined that separating a form into its essential parts will not necessarily entail back to the original form, regardless of its organic or inorganic features (Goethe 1989, pp. 22-23). Furthermore, Goethe declared that morphology, as a theory, will “include the principles of structured form and the formation and transformation of organic bodies” (Goethe 1988, p. 57). Goethe interpreted the theory of morphology as two German terms of “Gestalt” and “Bildung.” The former designates a moment of formation as a static model, which abstracts the “structured form,” and the latter describes the process of formation as a permanent condition, an interminable process that is continuously in progress (Goethe 1989, pp. 23-24). The form is a moment of the process of formation, which is dynamically changing. An observer can identify a single moment in this process to choose a new form. When an observer differentiates appearance aspects from structured aspects during the process of formation, new forms arise out of those immediate conceptions.

In the context of embryology, Murray (1990) defined morphogenesis as “the development of pattern and form in living systems” (Murray 1990, p. 119). Alan Turing (1952) in his pioneering paper, “The Chemical Basis of Morphogenesis,” suggested a mathematical model to describe a growing embryo as a reaction-diffusion system where chemical substances acting as morphogens catalyze the genes (Turing 1952, p. 37). Turing (1952) identifies morphogens as chemical substances that catalyze genes to facilitate growth processes. The embryological pattern that emerges from this process has a diversity forms depending on the chemical and mechanical properties of the embryo (Turing 1952, pp. 37-38). However, Turing’s mathematical model concentrates only on morphogen gradients that consider the chemical properties of the embryo. In an effort to include the intricacies of mechanical and chemical relations in simulating morphogenesis, Turing proposed a computer model with the mechanical properties of the embryo (Turing 1952, pp. 71-72). Since then computational models have been used to model complex morphogenesis.

Investigating embryological morphogenesis, as a natural development, furthers the study of growth processes under genetic modifications. The interrelation between different aspects of mathematical and geometrical insights of organic bodies is associated with the mechanical and chemical properties of constituent's materials. Considering only one aspect of morphogenesis, such as a mathematical simulation that only considers organisms as geometric patterns, undermines the contribution of other factors to the process of natural growth. Therefore, using mathematical abstractions of growth processes like cell differentiation only provide a theoretical method for generating form. This method will not necessarily generate forms that can be materialized. The theoretical form developed through the mathematical analysis of organic bodies is mainly evaluated by its geometrical properties. In the context of natural organisms, processes of form generation and growth are studied through morphology and morphogenesis. Morphologic studies provide insight into the form developments of architectural designs to derive constructible forms out of theoretical forms.

2.2.2 Identifying effective factors in morphologic studies

Theoretical morphology

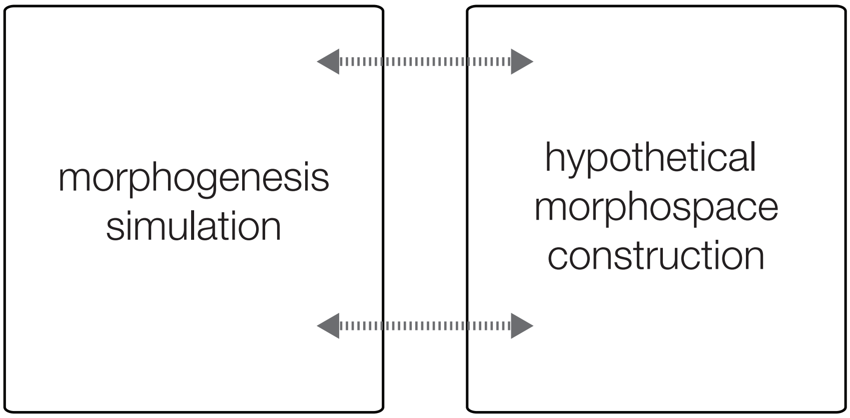


Fig. 2.2.1: The development of a "theoretical morphology" in bilateral relation between the simulation of morphogenesis and the construction of a hypothetical morphospace (McGhee 1999, pp. 1-2).

In theoretical biology, the development of forms are conceptually studied through the morphologic paradigms of "theoretical morphology," "constructional morphology," and "functional morphology" (Russell 1982, p. xi). McGhee (1999) differentiated "theoretical morphology" into two different areas of study (Figure 2.2.1): (a) the simulation and modeling of morphogenesis by means of simple mathematical explorations; (b) developing a "hypothetical morphospace" to evaluate and analyze the generated form (McGhee 1999, pp. 1-2). McGhee (1999) applied the "theoretical morphospace" to explore the mutability of form in nature and to extract new forms by manipulating the geometric parameters of the morphology. Moreover, he used this investigation to develop "n-dimensional geometric hyperspaces" as "theoretical morphospaces" (McGhee 1999, pp. 1-2).

> "In studying the functional significance of the coiled shell, it is important to be able to analyze the types that do not occur in nature as well as those represented by actual species. Both digital and analog computers are useful in constructing accurate pictures of the types that do not occur." (Raup and Michelson 1965, p. 1294)

According to Raup and Michelson (1965), the theoretical morphology emphasizes generating and analyzing the "nonexistent form," instead of investigating the "existent form" (McGhee 1999, pp. 4-5). Parametric modeling techniques simulate forms without considering any other criteria for form generation (Figure 2.2.2). In theoretical morphology, mathematical models investigate both existent and non-existent forms in nature. Hence, the theoretical form considers all the possible geometric forms that are producible with mathematical formulas. Pivotal parameters that distinguish existent forms in nature from non-existent forms are described by a hyper-dimensional morphospace. Changing any of the forms geometric parameters will alter the spatial distribution of derived forms within the hypothetical morphospace. Accordingly, a theoretical morphospace will examine the effectiveness of each parameter within the n-dimensional system. The analysis of these parameters will provide better insight into the behavior of form within morphogenetic movements.

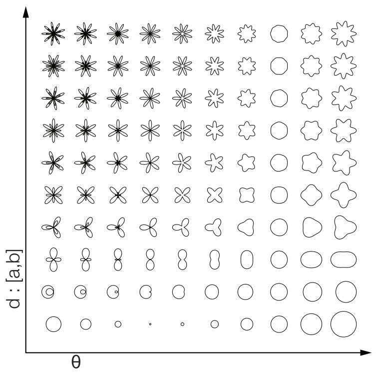


Fig. 2.2.2: The Geometrical differentiations of Limacon (Weisstein n.d.) denoted to the equation $r = b + a \times \cos(\Theta)$ in which a and b are parameterized to d.

Furthermore, the "empirical morphospace," which explores the empirical qualities of existent forms in nature, supports the hypothetical morphospace with evaluation criteria that are used to compare simulated forms with existent forms (McGhee 1999, p. 22). Accordingly, the hypothetical morphospace is marked with empirical regions to explain the behaviors of evolution pathways (Eble 2000, pp. 520-521). According to McGhee (1999, pp. 22-26), the borderlines between theoretical and empirical morphospaces differentiate the existent from nonexistent species (Figure 2.2.3). This differentiation is comparable with constructible and non-constructible simulated forms. An effective evaluation method, which is comparable with integrative design computation methods, can link the generation of forms with evaluation processes. However, the theoretical morphospace concentrates morphogenetic simulations in material properties and fabrication principles to determine the constructability of generated

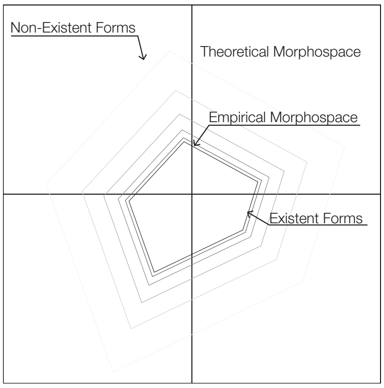


Fig. 2.2.3: The empirical morphospace articulates the theoretical morphology in two areas of existent and nonexistent forms; adapted from McGhee (1999, p. 25).

forms. In addition, developing constructible forms necessitates that morphogenetic movements follow empirical morphology. This process requires integrating mechanisms to monitor the movement in form generation. Cyclic feedbacks that are generated from this controlling mechanism lead integrative design methods to generate forms that are defined within a range of constructible forms.

Morphospace concept.

> "Morphological spaces (morphospaces) are spaces describing and relating organismal phenotypes." (Mitteroecker and Huttegger 2009, p. 54)

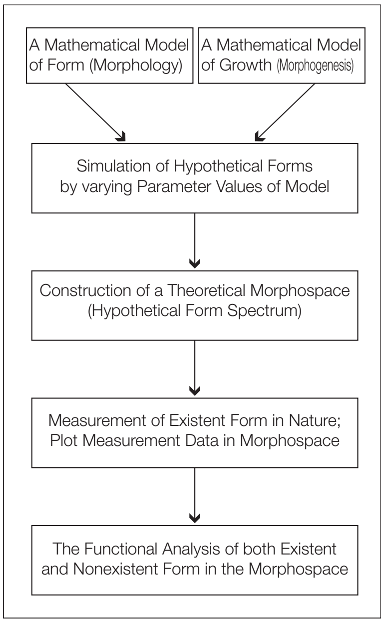


Fig. 2.2.4: A schematic diagram of developing the theoretical morphospace; redrawn by author based on: McGhee (2006, p. 61).

According to Mitteroecker and Huttegger (2009, p. 55), the morphospace, as a phenotypical definition of an organism, is a mathematical space that explains the cause of fracture between existent and non-existent forms. In accordance with this consideration, morphospace is an analytical tool that separates actual organismal structures occurring in nature, from their theoretically possible structures (Hickman 1993, p. 170). The morphospace determines areas of possible and impossible forms with multivariate model of forms from which each variable corresponds to a particular instance of forms. The theoretical morphospace is "n-dimensional geometric hyperspaces produced by systematically varying the parameter values of a geometric model of form"[1] (McGhee 1999, p. 18). Accordingly, analyzing forms through a theoretical morphospace requires three procedural steps: (a) developing a theoretical morphospace from existent forms; (b) scrutinizing form distribution in the morphospace in comparison with existent forms; (c) examining the functional aspects of form to discover the effects of adaptation on the development of form (McGhee 1999, p. 15). Later, McGhee (2006) considers two further steps to describe the development of a theoretical morphospace, see Figure 2.2.4. McGhee's (2006) flowchart illustrates existent forms that are explicitly investigated as mathematical models of morphology and morphogenesis. Parameterizing developed models provide a framework to simulate theoretical forms (McGhee 2006, pp. 61-63). One approach to the theoretical morphospace involves the development of a geometric model of morphology that exists in nature. Parameterizing the mathematical model of morphology facilitates the exploration of the mathematical structure of morphology from the empirical definitions of organisms.

---

[1] The theoretical morphospaces are originally defined by McGhee (1991), in his article "Theoretical Morphology- the Concept and its Applications" (McGhee 1991, p. 87).

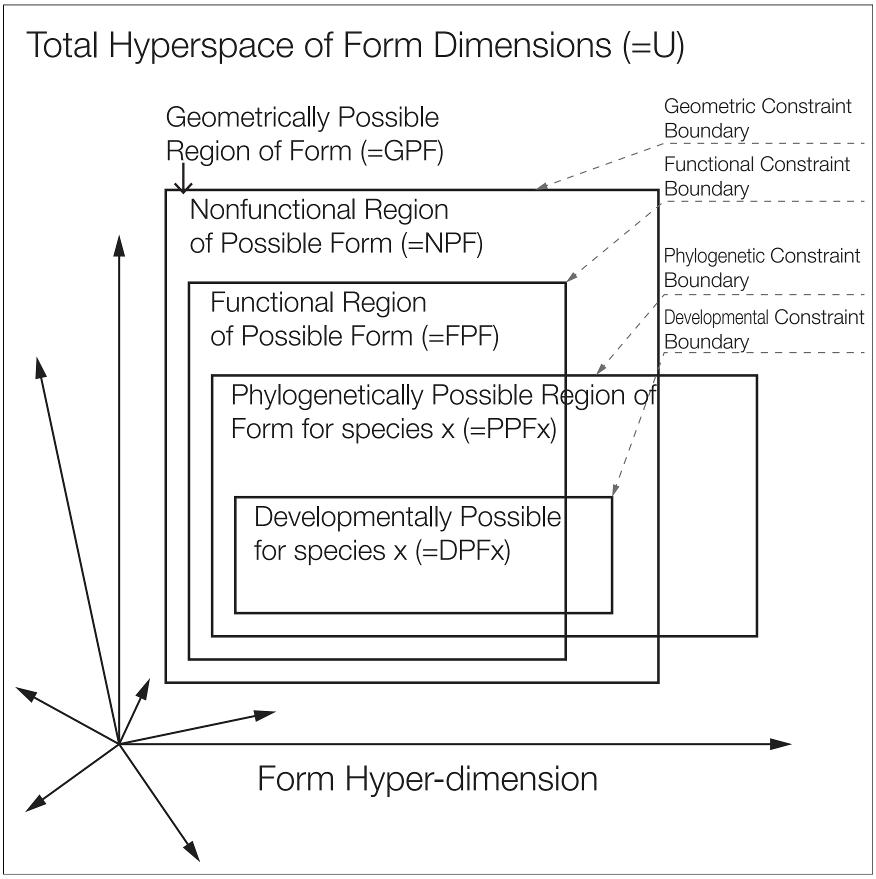


Fig. 2.2.5: The development of a Hyper-dimensional morphospace; redrawn by author based on: McGhee (2006, pp. 110-116).

In the context of the theoretical morphospace, analyzing possible forms considers a hyper-dimensional mathematical space (Figure 2.2.5) that is designated by two groups of evolutionary constraints: (a) extrinsic constraints that emphasize the extrinsic effects of geometric rules and physic laws; (b) intrinsic procedures that rely on biological significance in the morphogenetic development of an organism (McGhee 2006, p. 109, p. 116). Interplay between intrinsic developments and extrinsic constraints determine a region for exploring the potential of generating forms in nature. McGhee (2006) indicates that extrinsic constraints consist of a geometric boundary, which divide the hyper-dimensional space into “Geometrically Possible Forms” (GPF) and “Geometrically Impossible Forms” (GIF). The geometric possible forms include the subset of functional constraints with two regions of “Functional Possible Forms” (FPF) and “Nonfunctional Possible Forms” (NPF) (McGhee 2006, pp. 110-112).

McGhee (2006) further argues that, intrinsic constraints consider a phylogenetic boundary that determines “Phylogenetically Possible Forms for species x” (PPFx) and “Phylogenetically Impossible Forms for species x” (PIFx), in which the phylogenetically possible set specifies a developmental subset region for species. The possible region of existent form in nature is also determined by intersecting two subsets of “Developmentally Possible Form for species x” (DPFx) and “Functionally Possible Forms” (FPF) (McGhee 2006, pp. 112-114). Figure 2.2.6 shows the relation between extrinsic constraints and intrinsic constraints. The overlap area determines the possible set of developing existent forms.

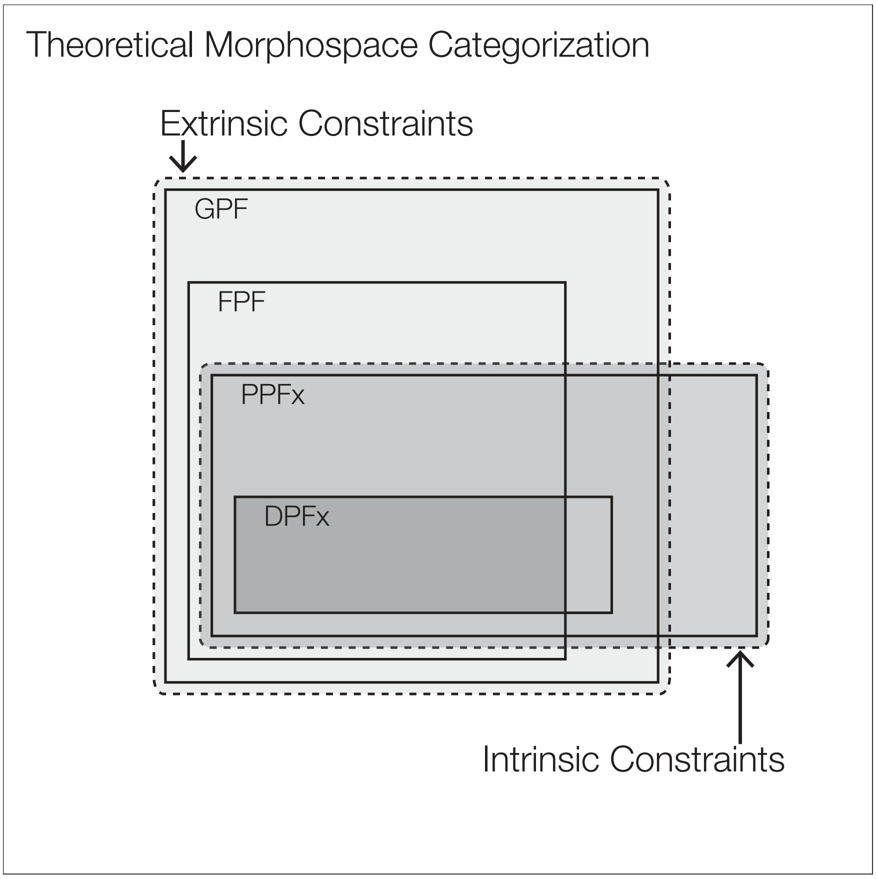


Fig. 2.2.6: Classification of theoretical morphospace into extrinsic and intrinsic constraints; redrawn by author based on: McGhee (2006, p. 116).

Extrinsic constraints are effective on the process of morphogenetic development while intrinsic procedures introduce the potential for an organism to evolve out of its genetically determined features. Categorizing the constraints in this way enables the correlation of the analytical aspects of a morphospace to the heuristic methods of a morphospace. In addition, extrinsic constraints modulate the process of form development from phylogenetic determination. Moreover, extrinsic procedures feed the phylogenetic developments to generate possible forms in nature. Heuristic methods are determined as a “theoretical developmental morphospace,” which is “a hypothetical spectrum of developmental possibilities” (McGhee 2006, p. 168). Eble (2003) elaborates on the concept of a “developmental morphospace” by emphasizing that it is vital “to encapsulate implication of process” rather than saturating morphospace with “a more pattern-oriented quality” (Eble 2003, pp. 40-41).

Constructional morphology

Seilacher (1970) introduced the term “constructional morphology” to address the different criteria that are involved in the process of generating constructible forms. Figure 2.2.7 illustrates the triangle of constructional morphology that was originally shown by Seilacher (1970). In comparison to theoretical morphology, constructional morphology involves the development of organic forms with a focus on the constructional aspects that participate in the process of formation toward the existent form in nature. From the perspective of paleontology, a constructional morphology that consists of morphogenetic (or fabricational) constraints is accompanied by phylogenetic and functional factors (Seilacher 1991a, p. 251). These factors directly restrict the evolvement of organisms to factors beyond materials and fabrications where they actively participate in the development of forms. Seilacher (1973) expands the concept of constructional morphology from the description of an organism’s form to the description of an organism’s built environment. He applies this concept to human artifacts with “fabrication noise,” which means utilizing different fabrication techniques to develop the same “functional morphology,” such as making one type of artifact during different generations (Seilacher 1973, p. 451).

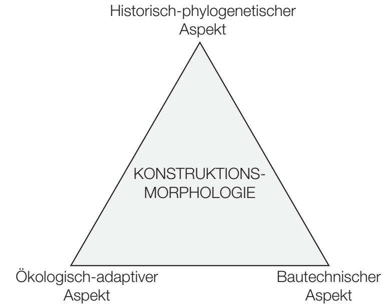


Fig. 2.2.7: The diagram of constructional morphology developed by Seilacher (1970). Seilacher (1973) considered these approaches as “traditional,” “functional,” and “fabricational” aspects (Seilacher 1973, p. 451); redrawn by author based on: Seilacher (1970, p. 394).

Synthesizing forms through constructional morphology is a heuristic method that includes “theoretical morphology,” “functional morphology,” and the concept of “Bauplan”[1] (McGhee 1999, pp. 8-9). According to Seilacher (1991a), utilizing the term constructional morphology within zoology only focuses on the internal aspects of functional morphology. Therefore, he introduced the term “morphodynamics” or “biomorphodynamics” to avoid any misunderstanding (Seilacher 1991a, p. 251). Investigating functional morphology in terms of both constructional morphology and theoretical morphology (morphospace) emphasizes the adaptive significance of morphologies in performing their main functions (McGhee 1999, p. 5). The dynamism of morphodynamics arises from the interrelation between constructional factors and “environment effectiveness,” which is a further extension of

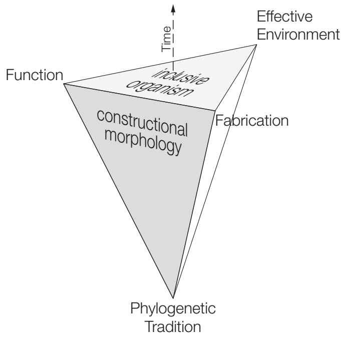


Fig. 2.2.8: The schematic diagram of morphodynamics or biomorphodynamics illustrates the relation between the triangle of constructional morphology and the tetrahedron of morphodynamics, which results in the face of “inclusive organism” (Seilacher 1991a,b); redrawn by author based on: Seilacher (1991a, p. 252), Seilacher (1991b, p. 6), and Seilacher and Gishlick (2014, p. 13).

---

[1] see McGhee (1999, pp. 6-7).

functional morphology representing the internal influence of external factors, see Figure 2.2.8 (Seilacher 1991a, p. 252).

Seilacher (1991a,b) described the relation between constructional morphology and morphodynamics in four main factors: "phylogenetic," "functional," "fabricational morphogenetic," and "environmental" factors. Each one of these drivers denotes one aspect of morphodynamics and its effect on morphogenetic movements. In the context of morphodynamics, inclusive organisms originate with phylogenetic factors that are described by the Bauplans with a specific evolutionary history (Seilacher and Gishlick 2014, p. 12). Within the evolutionary movement, Seilacher and Gishlick (2014) noted that the functional factors are formulated to reach desired goals, which are located at the top of the "adaptive landscape."[1] However, these goals are often unachievable due to the influence of other constraints (Seilacher and Gishlick 2014, p. 12). The major constraint that inhibits the development of an organism is fabricational factors, which are determined by material properties and the process of growth developments (Seilacher and Gishlick 2014, p. 12). Ultimately, both the organism and the environment determine an interactive system as an "inclusive organism," which explicitly employs the environmental parameters within the process of formation (Seilacher and Gishlick 2014, p. 12). Therefore, the three main factors of a "functional," "fabricational," and "effective environment" evolve into an inclusive organism that is connected to the "phylogenetic" or historical factors with a vertical time axis (Seilacher 1991a, pp. 252-253).

2.2.3 Inclusive design computation

The transition from biological context to integrative design necessitates the development of an inclusive design computation. *Inclusive design computation* corresponds to the active aspects of morphodynamics, where the inclusion of "effective environment," "morphogenetic fabrication," and "biological function" participate in the dynamic growth of species (Seilacher and Gishlick 2014, p. 13). Including these aspects within a design computation methodology advances the process of integrative design by employing functional, fabricational, and environmental aspects within a computational framework. Extending design computation with the concept of morphodynamics provides insight to the self-organizing integral drivers that derive and inform constructible forms. Morphodynamics considers constructional morphology and effective environment

---

[1] see McGhee (2006, p. 1).

within an organism in which the inclusive organism embraces the significance of theoretical morphology and functional morphology in interaction with the environment. The environmental adaptation of an inclusive organism with the functional (performative) aspects of form is accompanied by fabricational morphogenesis to evaluate the constructability of generated forms (Figure 2.2.9).

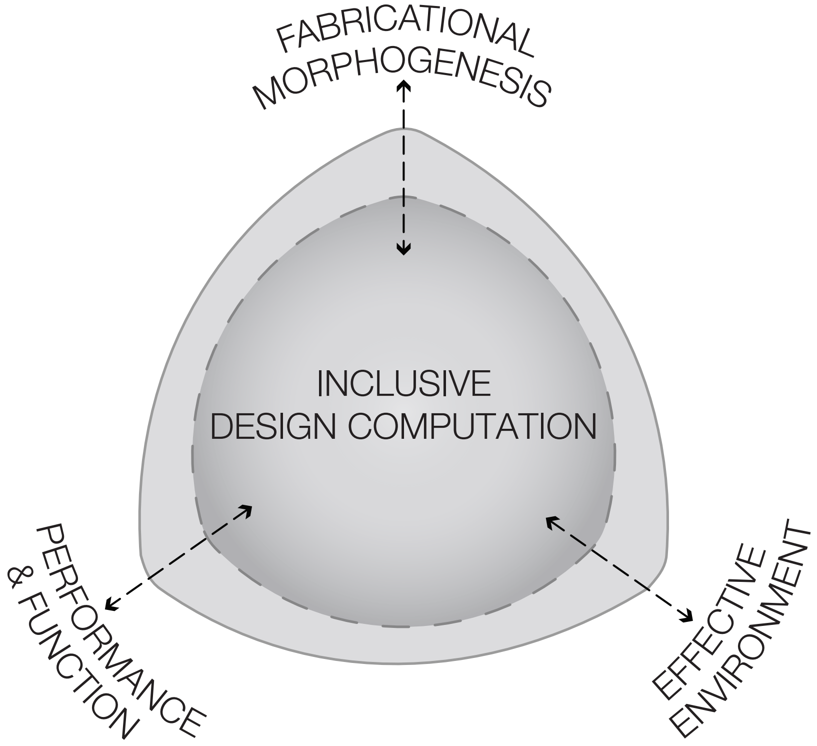


Fig. 2.2.9: Inclusive design computation is shown as the core of interactions among fabricational morphogenesis (fabrication), environment effects, and performative criteria.

Performative agencies

In nature, organisms adapt to their ecological niche in order to perform varied functions to ensure their survival. The development of organisms describes a transition from genetic constitutions to material structures in which the materialization processes are influenced by external environment factors, such as gravity (Seilacher and Gishlick 2014, p. 10). External pressures effectively modulate the process of formation in which organisms adjust their morphogenetic developments at both local and global levels. Within the local modulation, the organism alters “the [material] structure[s] from homogeneous to heterogeneous,” preparing the organs to perform appropriate functions in response to its related conditions (Vincent 2006, p. 227). This shows that the development of organisms follow adaptation at different scales from microcellular levels to macrostructures. The modulation of each level necessitates a multi-functional capability to adjust their functionalities in correspondence to external influences. Therefore, parts of organism are multi-functional while the whole organism performs singular tasks (Vincent 2006, p. 227). Hensel (2009) extends a "higher-level functionality" in biological structures to the "performative capacities" of built environments in which the transition from "multi-functionality" to "functional specificity" requires an application of "multi-objective optimization" instead of relying on "single-objective optimization" (Hensel 2009, pp. 2-3). The main challenge is shifting the paradigms from top-down analysis of the structural system of organisms to bottom-up developments of multi-functional elements and sub-elements. The functionality of an organism is determined at the early stage of growth and developments. The organism then adapts simultaneously to its own structural requirements and the external pressures of its environment. One approach that addresses the bottom-up development is "adaptive growth," which is a self-optimization process investigated through a "Computer-Aided Optimization" (CAO) method, in which this method facilitates flexible adaptation with environmental influences (Mattheck et al. 1991, pp. 15-16). It can be simplified to the principle of increased materials at high strain-stress zones ("overloaded zones") or decreased materials at low strain-stress zones ("underloaded zones") (Mattheck 1998, p. vii, 26, 31).

The logic of "adaptive biological growth" manifests geometric similarities among various parts of an organism, for instance, in the shape of bones or trees (Mattheck 1998, p. 26). Seilacher and Gishlick (2014) indicate that adaptation, within a specific ecological niche, can also be the overall shapes that organisms collectively make when performing similar tasks. For example, "functional convergence" can be exemplified by the general shape of a bird as an individual organism; or, by a flock of birds as a collection of organisms (Seilacher and Gishlick 2014, p. 2). "Functional convergence" can be extended to the built environments that follow specific functional manners. For example, envelopes, skins, or walls that indicate a separation between two different states also specify a functional convergence. Additionally, functional convergence categorizes the structural properties of a building system to beams and columns, which resembles the geometric categorizations of a building structure. Unlike current integrative design computation in the building industry, where performative criteria is examined by

FEM[1] and separated from the early stage of design. This process of functional adaptation considers the performative capacity of design principles as integrative drivers within the process of formation and materialization. Including performative criteria within design processes necessitates an understanding of both levels of the internal and external performance. At the internal level, the local quality of material development is associated with the global behaviors of structural systems.

Fabricational morphogenetic agencies

In the context of constructional morphology, fabricational morphogenesis, which is described by Seilacher (1973) as an essential driver, focuses on the effects of fabrication on the development of morphology. Fabricational morphogenesis includes two separated agencies: material and fabrication. Similar to theoretical morphology, the relation between material and fabrication are evident in the theoretical morphospace in which a simulated morphogenesis is used to understand the cause of variation between existent and non-existent forms. The theoretical morphology of species, which are further described as theoretical forms, are mathematical definitions of forms in which the regularities within one species are explicable in terms of their genome's information. However, a regularity between two different species that have different theoretical morphospaces, as in the case of horns and shells, is explainable by "fabricational noise" in the growth process (Seilacher and Gishlick 2014, p. 10). Extending the concept of "fabricational noise" to architectural design follows the mutual influences of material and fabrication agencies.

Material agency. Abstracting materials to simple physical and chemical properties reduces the material significance from active elements to passive constituents within design processes. Reconsidering materials as active agencies requires the use of material properties, capacities, and behaviors to perform specific functions (Hensel 2010, p. 38; Hensel 2011, p. 8). In addition, investigating material agencies are isolated from the material organization at a system level in the material properties. The capacity of material properties produce adaptive organizations in their surrounding environments. Addington and Schodek (2005) categorize material characteristics–"mechanical, thermal, electrical, chemical and optical"–into intrinsic properties that emphasize micro-level interrelations between the atomic and molecular

---

[1] Finite Element Method.

structures of chemical compounds. Or, extrinsic properties that are determined by the macrostructure of materials by modulating material compounds, such as energy fields of the environments (Addington and Schodek 2005, pp. 38-39). Extrinsic properties mediate the intrinsic characteristics of a material with external stimuli. Triggering material behaviors by external stimuli, such as field of energy, actuates intrinsic properties to exhibit their ultimate capacities. Activating microstructures alters the interrelations between atomic and molecular level of elements to self-organize and self-regulate their level of energy. This alteration directly affects extrinsic behaviors from which the material consistency might initiate phase transition or change the material compositions to new compounds.

In addition, extending external influences to fabrication agencies promotes a varied approach to manufacturing. The chosen method of manufacturing could include milling, cutting, welding, winding, or any combination of additive and subtractive manufacturing. The chosen approach to manufacturing can directly affect the extrinsic and intrinsic material characteristics by imposing or associating with material agencies. Correlating two different agencies requires a common communication platform to mediate between two different structures. Accordingly, geometric constitutions of materials, which underlie the geometry of forms, enable fabrication tools to negotiate directly with material agencies. Instrumentalizing the geometrical definition of a material advances the process of fabrication by recognizing impossible geometries at the level of construction. Conventionally, the impossible geometries necessitate rationalization to impose form on materials by legitimizing non-standard geometry. However, including materials within the process of design promotes active negotiations between material and fabrication agencies to determine the feasibility of construction. The bottom-up perspective differentiates material agencies to consider all constructible configurations with the benefits of extrinsic and intrinsic material properties.

Fabrication agency. In nature, fabricational movements rely on the process of growth and development. They are also associated with both internal and external influences, such as material agencies and environmental factors. The adaptation of an organism to its environment is concurrent with coalescing growth processes into materialization and formation. In contrast to adaptation, growth constraints as "fabricational noise" affects the process of morphogenesis (McGhee 1999, pp. 6-7). "Fabricational noise" indicates another level of adaptation through the "physical differentiation" of morphology, which might increase the functionality of morphology, such as the "wavy lines" of "burrowing ribs" (Seilacher 1973). Constraining growth by external factors, such as "fabricational noise," allows the process of construction to participate actively in form generation. Fabrication agencies follow understanding the constraints that arise out of the negotiation between material agencies and fabrication tools. In the context of digital fabrication, the constraints of fabrication tools can be simulated by developing a fabrication setup with specific material systems to compute the limitation of fabrication systems. Considering material and fabrication agencies enables designers to determine the "fabricational morphospace" (Seilacher and Gishlick 2014, p. 10) from which the developmental process embedded within the system allow a generative tool to expand using fabrication tools.

2.2.4 Effective environment

In nature, the process of transferring genome information to physical entities promotes an understanding of the importance of environment. According to von Uexküll (1926), the environment affects the process of an organisms' development. While some environmental factors are effective on the genesis of an organism, others have local influences (Von Uexküll 1926, p. 238). It seems that the environmental effects are identifiable with the intrinsic and extrinsic properties of organisms. In the context of morphodynamics, the environmental effects are extrinsic factors that also consider intrinsic drivers, at the same level of fabrication and functional factors, to construct inclusive organisms (Seilacher 1991a, p. 252). In addition to extrinsic effects, including environmental effects as intrinsic properties enables organisms to adapt themselves gradually to contextual environments. Growth under environmental pressures requires an organism to shape material properties for further "congruity" with the surrounding environment (Von Uexküll 1926, p. 315). Changing the environment of an organism reveals its range of adaptation. This adaptation will also reveal the internal links between the properties of an organism and environmental factors. The uniqueness of environments for each organism denotes "Umwelt Theory" in which each organism has a unique interaction with the environment, due to the specific sensory networks that transform environmental stimuli into characteristic properties (Von Uexküll 2009, pp. 145-146). If "Umwelt" is a bubble surrounding the objects, this environment coalesces the separated external drivers with the intrinsic properties of objects.

This means that the environment is directly embedded in the process of formation. Figure 2.2.10 illustrates the relation among different agencies in design computation.

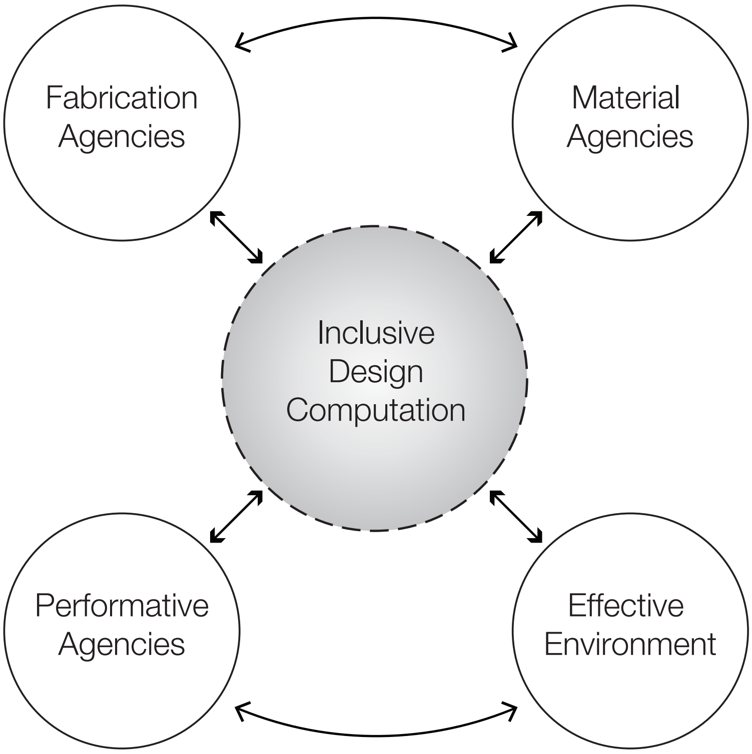


Fig. 2.2.10: Inclusive design computation represents the inclusion of different active agencies in the form evolvements.

2.3 Soft Design Computation

2.3.1 A system approach to generative design

The investigation of computational morphogenetic processes promotes the advancement of integrative design through *generative "design techniques," "production technologies,"* and *analytical design strategies* (Menges 2008). The inclusion of these drivers within a computational framework expands linear integrated design to nonlinear design processes. Effective parameters in "performance-oriented architecture" reflect the interdependencies of four active agencies that include "spatial and material organization" in a complex interaction with the "subject (inhabitant)" and the "environment" (Hensel 2010). Considering active agencies or drivers in the process of design emphasizes the significance of a system's ability to mediate between individual drivers. The exploration of biological aspects within existent forms indicate a dynamic inclusion of "effective environment," "biological function," and "fabricational morphogenetic" principles (Seilacher 1991a,b). Consequently, an inclusive design paradigm suggests a significant correlation between fabricational agencies, performative agencies, and environment effectiveness. Accordingly, developing a system that fuses these factors is essential to gain insights about the structure of interconnection and interaction between them. Soft design computation considers a design computation framework that adapts to arising behaviors of this fusion. Considering a soft system theory within design processes offers enough flexibility to intermediate among different inclusive design agencies. Subsection 2.3.4 provides further clarification to gain better insight into soft systems.

System approach

In the field of architectural design and practice, active drivers that are involved in the development of form require systems to synthesize active drivers from performative, fabricational, and environmental agencies. The mediators facilitate dynamic interrelations among embedded drivers. This mediation comprises of a systematic structure to facilitate decision-making within the design processes. In the context of designing inclusive forms, inclusive organisms within the realm of morphodynamics, structuring systems effectively enables interactions among embedded elements to synthesize dynamic forms. However, developing inclusive organisms through morphodynamic processes describes the system as an integrated form. From this perspective, the inclusive form is irreducible to the constituent elements. When the form is reducible to the constituent drivers, then the driven form will represent discernible arrangements of internal components. The passive agency of embedded components is distinct from the active agency of drivers, which pursue various taxonomies from materials to fabrication systems.

System definition

In order to structure inclusive mediator systems, it is necessary to understand the process of organizing and assembling various drivers within an integral system. The definition of systems posit that "sets of elements [stand] in interrelation" (Von Bertalanffy 1968, p. 37), this broad definition is required to establish the relationships among their constituent parts of a system. From an architectural design perspective, the integration and interrelation of components, regardless of their properties and behaviors, necessitates the investigation of open systems to provide a high-level of dynamic relations, integrations, and interactions (Kolarevic 2009, p. 338; Hensel 2010, p. 37). An inclusive system that is accompanied with design principles encapsulates complex forms where the flow of resources, such as energy and matter, organizes a high-level of system complexity (Weinstock 2004, p. 17).

System theory

Mediating between inclusive design drivers requires "isomorphism," a generalization approach to find resemblances between structures, behaviors, and properties of embedded parts (Van Gigch 1991, p. 62). Generalization processes allow systems to diversify assembly elements by first finding commonalities, and then differentiating them over those common structures. Accordingly, system theory describes a general approach to understand the underlying principles of all systems, such that their elements are connected by feedback loops (Anderson 1999, p. 219). Von Bertalanffy (1968) addresses this need for generalizations in "general system theory," which facilitates "integration in the various sciences, natural and social" (Von Bertalanffy 1968, pp. 36-37). General system theory is a novel substitute for "the analytical-mechanistic approach," which considers a reductionist approach that emphasizes a system reducible to its parts and elements (Van Gigch 1991, pp. 77-78).

System theory, which consists of several fields that use a variety of methods, takes a holistic approach, in which the whole is irreducible to the parts (Van Gigch 1991, pp. 77-78). In the sense of design principles, general system theory describes a form as a whole that is synthesized from different elements within an inclusive design computation. According to design computation, the fluidity of design information between encapsulated building blocks advances design systems to utilize a high-level of connectivity over the binary definition of data. Associating generalization approaches to the design system promotes the development of generative algorithms to construct an inclusive form as a whole.

Generative systems | Holistic systems

In the paper, "Systems Generating Systems," Alexander (1968) describes a system as a whole that must exhibit holistic phenomena. From Alexander's perspective, holistic behaviors are directly related to the elements within the system, their interactions, and the way that they interact (Alexander 1968, p. 607). Developing generating systems that produce holistic behaviors, requires an abstraction of elements and their individual interactions. From this perspective, interacting elements that are constituted at the generative level of a system are effectively involved in the behaviors of the whole system. However, constituent elements are completely unaware of the exhibited behaviors. Therefore, the interactions between embedded elements are essential to achieve holistic properties. Synthesizing a holistic system is a combination of heterogeneous and homogeneous factors, and not only the sequential arrangements of them. In the context of inclusive design, the holistic properties of synthesized forms emerge from interactions among constitutional integrative elements that are abstracted into algorithmic blocks to facilitate integration process within a complex system.

Constrained generating procedures (CGP's)

Leading a generative system towards holistic behaviors requires a structure that will simultaneously constrain its generative features. Holland (2000) defines Constrained Generating Procedures (CGPs) as dynamic models that underlie mechanisms to generate possibilities and procedures to constrain them (Holland 2000, p. 126). Providing interactions between individual mechanisms advances generative systems to probe the solution space while developing constraining mechanisms that dynamically evaluate the generated behaviors. Simultaneous evaluation of generated behaviors permits generative systems to regulate the flow of generated dynamic behaviors.

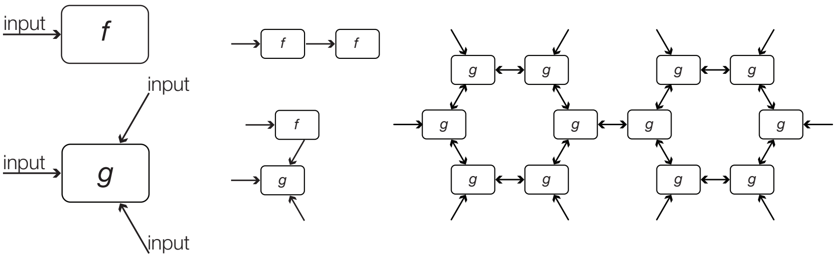


Fig. 2.3.1: Conducting CGPs through two primitive mechanisms as f and g; redrawn by author based on: Holland (2000, p. 135).

Holland (2000) elaborates on the development of Constrained Generating Procedures (CGPs) in four steps: (a) transferring rules into mechanisms; (b) developing a network of linked mechanisms; (c) determining a transition function to permit the transition between the state of mechanisms over time; (d) defining assembly procedures to develop “more complex mechanisms” out of “basic mechanisms” (Holland 2000, pp. 126-129). Figure 2.3.1 highlights the relationship between two primitive mechanisms as f and g, which are differentiated by the number of their input parameters, from which interconnecting primitive mechanisms together form new CGPs (Holland 2000, p. 135). According to Holland (2000), the state of each primitive mechanism at time $t+1$ relies on input parameters, in which state of mechanism f with the input parameter I, is denoted as S, and state of mechanism g with the input parameters $\{I_1, I_2, I_3\}$, is denoted as S'. Therefore:

$$\begin{aligned} S(t+1) &= f[I(t), S(t)] \\ S'(t+1) &= g[I_1(t), I_2(t), I_3(t), S'(t)] \end{aligned} \tag{2.1}$$

(Holland 2000, pp. 132-135).

The network of assembly mechanisms articulates a global state of CGPs, which generates complex behaviors under transition functions. According to Holland (2000), extending the notion of mechanisms to CGPs allows a network of CGPs to build a new level of CGPs with high complexity. These bottom-up hierarchies organize a complex system to exhibit emergent phenomena (Holland 2000, p. 129).

2.3.2 Modeling a complex adaptive system

Complex system

Simon (1962) describes a “complex system” as a system in which a large number of parts interact in such a way that the generated "whole is more than the sum of the parts" (Simon 1962, pp. 467-468). Accordingly, comparing a complex system to holistic concepts suggests a consideration of complex systems as systems that exhibit emergent phenomena. Anderson (1999) explains that a complex system resists reductionist approaches. This resistance is due to the interrelations of feedback loops established between the subsystems (Anderson 1999, pp. 217-218). In general, the behavior of the whole system, which represents emergent phenomena, is discernable from the summation of individual behaviors. However, Minsky (1988) contends that emergent phenomena are explicable by considering the interactions between elements and the particular significance of observers (Minsky 1988, p. 328). This means that explaining emergent properties relies on two levels of interactions and observations. Estimating the behaviors of complex system corresponds to the observers' knowledge and their perspectives about the behaviors of system interactions.

In system theory, emergence is the irreducible phenomena in which "the global properties defining higher order systems or 'wholes' (e.g. boundaries, organization, control, ...) can in general not be reduced to the properties of the lower order subsystems or 'parts'" (Heylighen 1989, p. 23). Emergence that is described as a global behavior is an aggregation of local behaviors from which the derived global behavior is irrelevant and disconnected from the initiative local behaviors of elements (Miller and Page 2007, p. 44). However, the whole system behavior is dependent on an appropriate level of details and relevant mechanisms, which determine the rules of interaction between elements to consider emergent phenomena (Holland 2000, p. 44).

The interactions among elements are defined by a state of linearity and nonlinearity, in which nonlinear interactions emphasize failures to follow "superposition principles" (Kwinter 1993, pp. 211-212). Accordingly, complex systems with nonlinear interactions require mechanisms to control the level of order within the system. The development of such mechanisms is directly proportional to the relation of a system to its environment. For example, a living organism, which has constant interactions with the environment, is described as an open system that exchanges matter and energy with its environment (Von Bertalanffy 1968, p. 31). A mechanism that controls the effects of interaction with an environment comprises of feedback (Figure 2.3.2), in which the system's responses are regulated and monitored (Von Bertalanffy 1968, p. 42; Van Gigch 1991, p. 74).

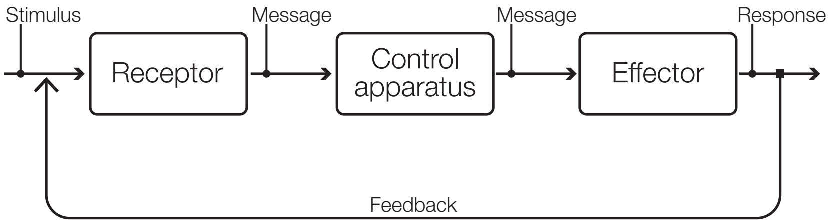


Fig. 2.3.2: The simple mechanism of Bertalanffy's feedback loop; redrawn by author based on: Von Bertalanffy (1968, p. 42).

The amplifications within positive cycles increase the level of system activities towards instability, wherein stabilizing the system requires negative feedback to modulate feedback loops (Miller and Page 2007, p. 50). For instance, Figure 2.3.3 illustrates that positive (birth) feedback loops reinforce initial inputs with modified outputs of the system, and negative (death) feedback loops stabilize the amplified system (Camazine et al. 2003, p. 17). Therefore, feedback loops regulate the system's behaviors through intensifying and reducing the initial state of the system.

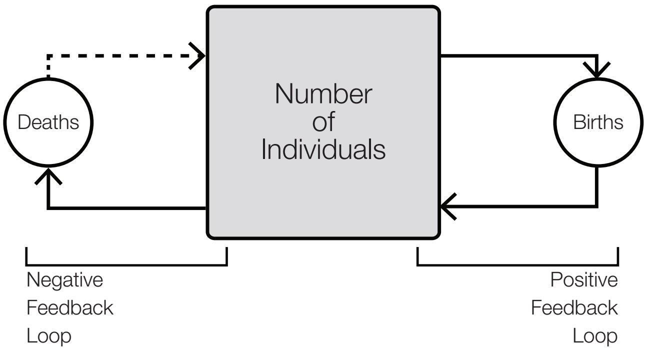


Fig. 2.3.3: A schematic diagram of population growth with two negative and positive feedback loops; redrawn by author based on: Camazine et al. (2003, p. 17).

The process of self-regulating within an open system, which includes nonlinear interactions, is called "self-organization" or "autogenesis" (Anderson 1999, p. 222). In the computational realm, self-organization indicates a regulation of complex systems through digital algorithms. The state of generative systems leads system's components toward a dynamic equilibrium with intrinsic and extrinsic pressures. De Wolf and Holvoet (2005) indicate that a complex and dynamic system may have self-organization without emergence, emergence without self-organization, or it may have either (Figure 2.3.10). A dynamic system that is self-organized may exhibit emergent phenomena (De Wolf and Holvoet 2005, p. 13).

Complexity theory

> "The central task of a natural science is to make the wonderful commonplace: to show that complexity, correctly viewed, is only a mask for simplicity; to find pattern hidden in apparent chaos." (Simon 1996, p. 1)

The development of a complex system that consists of various layers requires a recognition of underlying layers and their interrelations. The interplay among layers typically generates complex behaviors in which excessive detailing with a high-level of interrelation among layers yields unnecessary complexities. These complexities then challenge generative systems to produce alternative possibilities. Therefore, abstracting and simplifying constituent layers avoids further difficulties for generating systems. In addition, this modeling technique must achieve the purpose of the model, where an inadequate level of details would make it impossible to achieve the purpose of the modeler (Starfield et al. 1990, pp. 1-8).

In general, increasing the number of elements in a system directly relates to the level of that system's complexity. The interactions among elements might increase exponentially as the number of elements increase. Gershenson (2007) introduces a mathematical equation to represent "Random Boolean Networks" (Kauffman 1993) in which the complexity of a network can be formalized with the following equations:

$$C_{sys} \sim \left\{ \left(\#\bar{E}, \#\bar{I}, \sum_{j=0}^{\#\bar{E}} C_{ej}, \sum_{k=0}^{\#\bar{I}} C_{ik} \right) \right\} \tag{2.2}$$

, where C_{sys} is the complexity of network, $\#\bar{E}$ is the number of elements, $\#\bar{I}$ is the number of interactions, C_{ej} is the complexity of elements, and C_{ik} is the complexity of interactions (Gershenson 2007, p. 13). Changing each one of these features modulates the complexity of the system. For example, adding more elements to the system causes more interactions between the systems. In addition, the complexity of the system is amplified when the complexity of each element and their interrelations is fed back to the system.

The abstraction of a layer and its interactions with other layers may describe the complexity of the entire system. Cohen and Stewart (1994) challenge the view of "cause and effect" in complexity theory; specifically, by which "simple causes" generate "complex effect" (e.g., *chaos*) and "simple effects" emerge from "complex causes" (e.g., *antichaos*) (Cohen and Stewart 1994, p. 20).

Therefore, other criteria may exist within complexity theory that indicate non-linear behaviors of complex systems. However, the central aspect focuses on elements and their interactions to gain insight into how the organizing elements within a model explain emergent behaviors.

Disorganized complexity | Organized complexity

Weaver (1948) developed a theory of organized and disorganized complexities to study complex behaviors within a system. From Weaver's (1948) point of view, "disorganized complexities" consider a large number of elements through probability theory and statistical mechanisms. Weaver (1948) exemplifies his theory through a large billiard table with millions of balls, which are in motion and interaction. In addition, the behaviors of each individual ball are random, or unknown, but the overall behaviors of the system are predictable and analyzable through statistical techniques (Weaver 1948, pp. 3-4). The theory of disorganized complexity has roots in the second law of thermodynamics, which was how the behavior of a large number of gas molecules was described (Von Bertalanffy 1968, p. 33).

In contrast, Weaver (1948) indicated that "organized complexity" is a system with a finite number of elements that are organically interrelated. In organized complexity, statistical techniques are insufficient to find the average behaviors of the whole system because many factors are involved (Weaver 1948, p. 6). In organized complexity, the feedback cycles among interrelated elements reinforce each other instead of canceling each other out (Miller and Page 2007, p. 53).

Complex adaptive system (modeling complex behavior)

Complex Adaptive System (CAS) is a framework with a large number of building blocks that include both rules and agents (Holland 1992, p. 197). The interconnected networks of individuals advance a complex system to "adapt" or "learn" via their interactions (Holland 2006, p. 1). The distinctive features of a Complex Adaptive System (CAS) in comparison with other systems are its emergent properties, which arise from the interactions of individuals at "the lower-level of aggregation" (Anderson 1999, p. 219). Holland (1992) introduced a framework that constructs a CAS in his influential book, *Adaptation in Natural and Artificial Systems*. This framework involves three mechanisms: (a) "parallelism," which facilitates the activities of individuals associated with changing states; (b) “competition,” which enables the system to arrange its resources to avoid any inundation of the system with flows of unrelated information; (c) “recombination,” which forms a basis for generating conceivable new rules from experimented rules by individuals (Holland 1992, p. 197). Holland (1992) argues that describing a theory of complex adaptive systems entails formalizing the aforementioned framework with an emphasis on processes instead of an emphasis on results (Holland 1992, pp. 197-198).

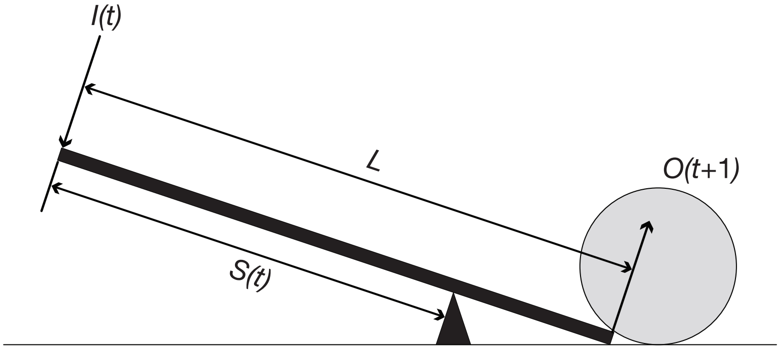


Fig. 2.3.4: A simple diagram of a lever emphasized within Complex Adaptive System (CAS); redrawn by author based on: Holland (2000, p. 127).

Accordingly, the behaviors of individuals are important to estimate the behavior of the entire system. The behavior of the individual includes a set of “stimulus-response rules,” which is a collection of “IF/THEN” mechanisms to facilitate the negotiation between stimuli and responses, such as “IF stimulus *s* occurs THEN give response *r*” (Holland 1995, p. 7). Within complex adaptive systems, the aggregation of individuals’ behaviors is adaptive to changing situations, where each individual process embeds a rule to find the most fit response. The adaptation that arises from the accumulation of appropriate behaviors is sensitive to any stimuli. Holland (1995) describes the critical moments of system adaption as “lever points,” “wherein small amounts of input produce large, directed changes” (Holland 1995, pp. 39-40).

Figure 2.3.4 highlights the formalization of lever points through the following equations:

$$O(t+1) = f(I(t), S(t)) = \begin{cases} I(t) \cdot S(t) \\ L - S(t) \end{cases} \tag{2.3}$$

, where $O(t+1)$ is the output of primitive mechanisms (transition function) f, $I(t)$ is input forces, and $S(t)$ is the fulcrum point (Holland 2000, p. 127). Extending the concept of a lever point from complex adaptive systems to constrained generating procedures means that the "transition function" of the adaptability state of a complex system becomes the fulcrum with the values of input parameters at time *t* (Holland 2000, pp. 126-127). Therefore, re-establishing adaptation in complex adaptive system relies on recognizing the significance of lever points and determining the exact amount of system inputs. Accordingly, large changes are required to explore a new level of adaptation through drastic changes within the recombination mechanism. Finding the lever points enables the constitution of an effective Complex Adaptive System (CAS) framework in which the hidden potential of CAS is revealed.

2.3.3 Approaching to the computational modeling

> "The world is its own best model." (Brooks 1991)

Model definition

Complex systems, which exhibit holistic behaviors, are difficult to comprehend. Complex behaviors, which arise from interactions among parts, need a model to gain insight into processes that exhibit emergent phenomena. In general, models are representation methods to solve problems that cannot be modeled without an exact definition of purpose (Starfield et al. 1990, pp. 1-8). Exploring the problem to recognize appropriate purposes allows a model to find solutions that focus on central problems. Therefore, a model should be simple enough to represent its main purpose. The purpose of self-organizing a system is adaptation with dynamic complexity arising from emergent phenomena. Avoiding unnecessary details allows models to maintain a reasonable level of complexity (Miller and Page 2007, pp. 36-37). This method of modeling allows further analyses and investigations; otherwise, another level of simplification would be required to explain the growing complexity.

Modeling relation

The main purpose of modeling is to find a solution for a complex problem that exists. The transition from a real world problem as an informal system to the formal system is called "modeling relation," see Figure 2.3.5 (Casti 1994, pp. 274-275). Modeling relation consists of "nature, or (real-world system)," "the natural system," and "the formal system" (Van Gigch 1991, p. 123). Casti (1994) describes a modeling relation that includes a process of encoding a natural system to a formal logical system through mathematical or symbolic logics. For example, Turing's machine uses modeling relations to formalize the informal notion of computation (Casti 1994, pp. 274-275). The modeling relation therefore consists of abstraction processes that observe the complexity of the real world and simplify informal systems into mathematical logics. This simplification is also associated with rules and procedures to formalize the generative aspects of interactions among abstracted parts.

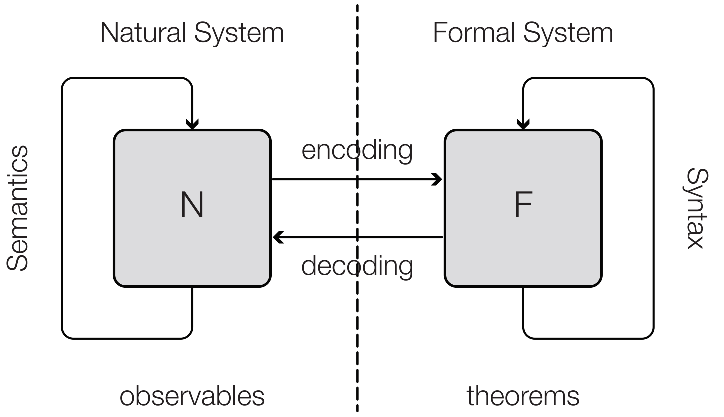


Fig. 2.3.5: A diagram of modeling relation, where *N* is computation and *F* is the Turing machine; redrawn by author based on: Casti (1994, p. 274).

Homomorphism

Miller and Page (2007) describe a formal model of models with a schematic diagram (Figure 2.3.6). In this diagram, the real world consists of different states at both the given time t and time $t+1$. These times are mapped by a transition function, which can be described through $S_{t+1} = F(S_t)$ (Miller and Page 2007, pp. 38-39). According to Miller and Page (2007), due to the complexity of $F(S)$, modelers are required to reduce the size of states S_t and S_{t+1} through an equivalence class of $E(S)$. Modelers are also required to pursue a simpler transition function between time t and time $t+1$ as $f(s')$(Miller and Page 2007, pp. 38-39). They further explain that the equivalence class $E(S)$ transforms the state S_t into $S'_t = E(S_t)$. This equivalence class can transform the state S_{t+1} into the state $S'_{t+1} = E(S_{t+1})$ (Miller and Page 2007, pp. 39-40). Miller and Page (2007) conclude that this equivalence class predicts the transition function of $F(S)$ as $f(s')$, in which it states that $S'_{t+1} = f(s'_t)$. The real world coincides with the model if $E(S_{t+1}) = f(s'_t)$ or $E(F(S_t)) = f(E(S_t))$ as a "homomorphism" (Miller and Page 2007, pp. 39-40). The comparison between predicted states in a formal system (model) and informal states (real world) defines the level of accuracy of the developed model.

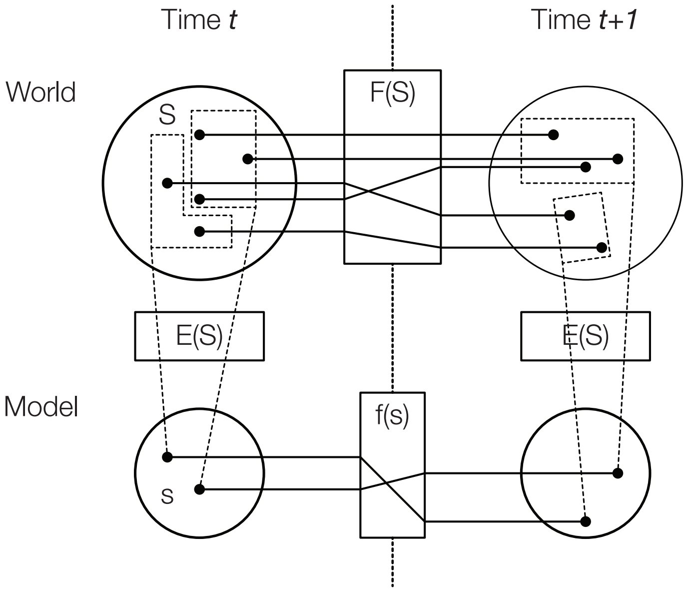


Fig. 2.3.6: A "formal model of models," where S is the state of the model, $F(S)$ is a transition function, $f(s)$ is a simpler transition function and $E(S)$ is the equivalence class (Miller and Page 2007, p. 38-40); redrawn by author based on: Miller and Page (2007, p. 38).

Model of inclusive forms

Extending formal modeling to include forms promotes the abstraction of different layers of integral materialization into a computational model. The forms that consist of several layers of constituent elements designate three states of performative (functional adaptation), fabricational morphogenesis, and environmental effectiveness. The dynamism of inclusive forms, which is similar to the concept of morphodynamics, is accompanied by the time sequence. The modeling relation can also describe the transition of inclusive forms from time t to time $t+1$. Equivalence classes reduce various levels of forms to essential parameters. These parameters describe the main objectives of each active driver. In fabricational states, intrinsic and extrinsic material properties, which comprise of chemical and physical characteristics, are reduced to the geometrical interpretations of different material organizations. Therefore, inclusive states are reduced to selective states in accordance with the purpose of modeling. Inclusive modeling also has a transition function that produces a new state at time $t+1$ through the generative integration of selective states. The transition function that amalgamates those states into new states requires a proper evaluation to confirm that the generated states exist within the acceptable range.

Computation modeling

As a logical system, a formal model underlies the symbolic significances of the real world. According to Casti (1994), Turing's machine, as the first computational model, is a formal system to encode natural systems into logical systems (True or False). The logical formalization within Turing's machine is the conceptual basis for structuring current computers (Casti 1994, p. 275). The formalization of computational concepts suggests that the development of a computational formal model is not the only way to formalize the natural system. Therefore, a computer machine is not necessary for developing a computational model, e.g., Schelling's model (1978) developed originally with coins and papers (Miller and Page 2007, pp. 64-65). However, computer machineries are useful for learning how to develop a simple organization that is intrinsically complex (Kwinter 2003, p. 91). Accordingly, inclusive forms differentiate interactions among inclusive drivers towards a series of constructible forms from which the mediation between these drivers correlate the real world to the simulated forms. Developing a model with the logical structure of a computer formalizes the model due to the modeling relation. The accuracy of the model relies on its homomorphic properties.

2.3.4 Comparison between formal and informal models

Hard system and soft system

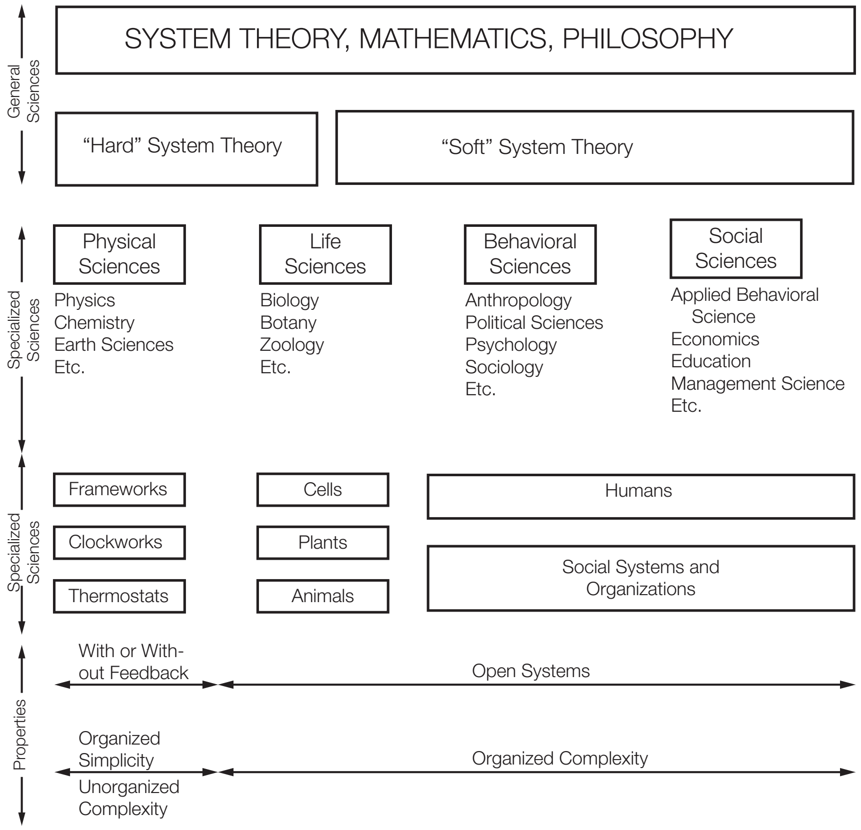


Fig. 2.3.7: A system classification, developed by Boulding (1968); redrawn by author based on: Van Gigch (1991, p. 66).

Van Gigch (1991, pp. 65-66) expanded Boulding's classification of systems (1968) by categorizing systems as either "soft" or "hard" (Figure2.3.7). Van Gigch (1991) describes "hard systems" as formalized reasoning processes that are traditionally applied in the physical sciences. In contrast, Van Gigch (1991) describes "soft systems" as informal reasoning processes where the behavioral characteristics are of interest. Formal reasoning processes are scientific that rely on analysis and deduction; however, informal reasoning processes depend on synthesis and induction (Van Gigch 1991, p. 79).

Kwinter (1993) characterized the softness of a system by the amount of flexibility needed to adapt to its changing environments. The adaptability of a system is accomplished through feedback loops and internal regulating mechanisms (Kwinter 1993, pp. 211, 218). According to Van Gigch (1991), modeling a soft system without accurate predictabilities necessitates heuristic methods that consider algorithmic approaches from a hard system. This method might only generate satisfactory solutions; it might not produce optimum solutions (Van Gigch 1991, pp. 84-85). Soft systems are applicable when the modeler wants to investigate nonlinearity or behavioral characteristics. In particular, when a system interacts with a dynamic system, in which its behaviors are continuously changing, the system requires its behaviors to sustain the new states. The concept of a soft system in architectural design was introduced by Brodey (1967) when he proposed the theory of "soft architecture" to conceptualize "intelligent environments." He exemplified the idea of soft architecture or soft environments by describing a "dynamic transit system which maintains its purpose in relation to the town" (Brodey 1967, p. 12).

In their pioneering paper, "Behavior, Purpose and Teleology," Rosenblueth, Wiener, and Bigelow (1943) emphasized behavior classifications and "teleology" as an early concept of self-organization (Figure 2.3.8). Controlling mechanisms achieve the "purpose" of a system through a "feed-back" mechanism (Rosenblueth et al. 1943, p. 23). Soft systems control their actions through environmental inputs and change their courses to maintain their state and then capture their embedded purpose. In contrast to "hard architecture," Negroponte (1975) developed the concept of a "soft architecture machine" in which the machine provides a "custom-made" design to personalize artifacts through "physical responsiveness" (Negroponte 1975, p. 145). Accordingly, a soft system can be considered a modeling technique that consists of a set of stimuli and responses associated with feedback loops. A complex system with adaptive behaviors is more flexible and robust to environmental perturbations.

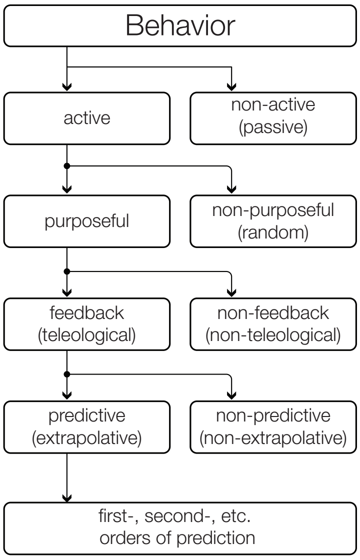


Fig. 2.3.8: The classification of behavior; redrawn by author based on: Rosenblueth et al. (1943, p. 21).

Knowledge-based system | Behavior-based system

The comparison of hard systems with soft systems helps designers recognize the importance of behaviors in the development of a computational design framework. When a knowledge-based system is replaced with a behavior-based system, the computational framework must inherit paradigms from artificial intelligence. In addition, this paradigm shift is accompanied by two approaches, one top-down and one bottom-up. The top-down approach tends to dominate the overall system by complete knowledge, while the bottom-up approach attempts to adapt itself with limited knowledge (Maes 1993). Moreover, the top-down approach uses an algorithmic mechanism to solve the problems, while the bottom-up approach uses algorithms heuristically to exploit the solution space.

The way that these two approaches exploit the solution space advances the study of artificial intelligence by exposing the structure of intelligence systems. Brooks (1990) in his pioneering paper, “Elephants Don’t Play Chess,” differentiates symbolic artificial intelligence (AI) from the nouvelle AI by proposing “the physical grounding hypothesis.” According to Brooks (1990), this hypothesis enables direct interactions between physical agents with the environment. From his perspective, classical AI structures intelligent systems using overall behaviors, while modern AI uses the interplay of individual behaviors embedded within subsystems (Brooks 1990, p. 3). Situating physical agents within the environment promotes the development of different layers of actions. Each layer responds to specific stimuli from the environment. The distribution of knowledge among the layers generates different behaviors. In combination, these behaviors exhibit complex emergent properties. Developing a control system that yields emergent behaviors requires that individual behaviors are engaged in the overall decisions. In contrast to the traditional control system, Brooks (1986) distributes the control system into different layers. The controlling layers act as “subsumption architecture,” where “the higher-level layers” subsume “the lower-level layers” (Brooks 1986, p. 14). In addition, this layering of a control system facilitates the addition of new control layers, at least while the lower-level layers are still functioning (Brooks 1986, p. 16). Figure 2.3.9 illustrates the early structure of the subsumption architecture, wherein the higher-level layers are structured upon the low-level behaviors.

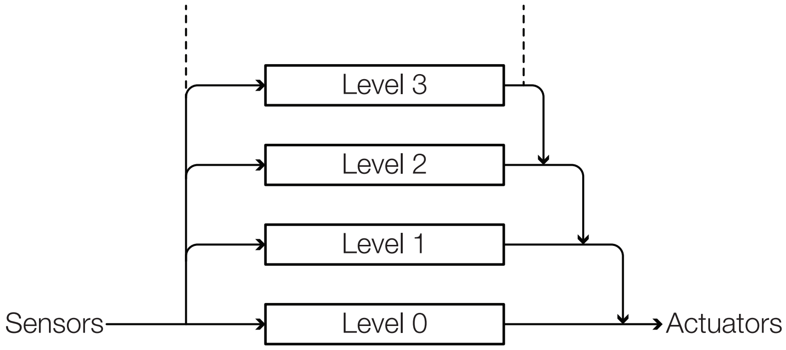


Fig. 2.3.9: A schematic diagram of a layered control system; redrawn by author based on: Brooks (1986, p. 17).

Maes (1993) extended the concept of “subsumption architecture” by emphasizing the role of knowledge and behaviors in artificial intelligence. According to the nouvelle AI, emerging intelligence relies on interactions between a system and its environment. Exhibited intelligence contrasts the tendency of creating a fully informed system that is a system with whole knowledge about the solution space. In addition, the informed system demonstrates the behaviors that are already embedded within the system. The information embedded in the system is limited to the specific setup of the known environment. Maes (1993) argues that a behavior-based system as a complete system will benefit from the experience of other systems that are situated in certain environments. This ability to share experience will also lead to less modeling (Maes 1993, p. 5). Advancing a complete system with distributed knowledge requires communication mechanisms for exchanging data. It is important to exchange enough data to consider the complete system an open system, which can adapt to unknown situations. Accordingly, a complete system with subsumption architecture will try to adapt to unexplored environments. According to Maes (1993), adaptations that arise from dynamic interactions among the components of a complete system, the system and its environment, and the system and other systems can lead behavior-based systems to exhibit "emergent complexity." Emergent complexity is "often more robust, flexible and fault-tolerant than programmed, top-down organized complexity" (Maes 1993, pp. 5-6). Accordingly, the behavior-based system is accompanied by different characteristics, such as soft system, complex adaptive system, bottom-up organized complexity, and most importantly emergent phenomena.

2.3.5 Towards agent-based objects

Developing a system to generalize different types of elements provides a framework to integrate the inclusive fabrication aspects of derived forms. The interconnection among these elements generates complex behaviors that require procedures to lead the generated behaviors toward emergent phenomena. Considering CGPs that use rules to constrain generative possibilities enables a system to yield complex behaviors that are organized from the bottom-up. Figure 2.3.10 illustrates the relation between self-organization and emergence within a system. De Wolf and Holvoet (2005) speculate that self-organization can be considered in both micro and macro levels. According to De Wolf and Holvoet (2005), interactions among micro-level entities compel systems to demonstrate emergent behaviors. Self-organization at the macro-level increases the level of order, which is unachievable through complicated interactions among micro entities (De Wolf and Holvoet 2005, pp. 11-12). Micromechanisms conducted within CGPs indicate the way that interactions among elements lead micro-behaviors to become macro-regularities. However, self-organization at the global level increases the adaptability of systems. Local levels of interactions are incapable of creating such regularities.

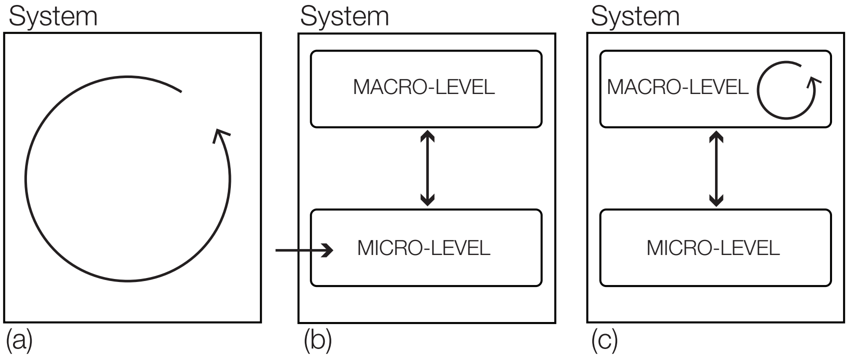


Fig. 2.3.10: The schematic illustration, developed by De Wolf and Holvoet (2005), represents the self-organization of a system (a), the emergence of a system (b), and the combination of self-organization and the emergence (c); redrawn by author based on: De Wolf and Holvoet (2005, p. 10).

Extending CGP to adaptive systems allows generative systems to consider some level of adaptations to generate complex adaptive behaviors. This complex adaptive system implicitly considers generative mechanisms with limited knowledge to explore the problem domains. The bottom-up structure of these exploratory systems exploits problem domains to find adequate solutions. The interaction among agents and the problem domain uses feedback loops to trigger behavioral mechanisms. These mechanisms use embedded knowledge within the solution space to solve problems. The behavior-based system with the feedback loops reflects a soft system, which self-organizes the system with unknown problems via behavioral mechanisms.

In contrast to knowledge-based system that tackles only one predefined problem, the behavior-based system uses distributed controlling layers to adapt to problems with multiple layers. In the context of behavior-based adaptation (learning), Matarić and Michaud (2008) consider both “primitive behaviors” and “abstract behaviors” to accomplish tasks. In this context, “basis behaviors” are essential to achieving goals, while “abstract behaviors” consider the condition for activating behaviors (Matarić and Michaud 2008, pp. 897, 899). Separating these behaviors into these two levels allow the behavior-based systems to confront extrinsic domains. A comparison between the subsumption behavioral layers and the primitive behaviors suggests that primitive behaviors control the activation of lower-level behaviors. However, the subsumption behaviors suggests that adding more layers of behaviors at the lower-levels will allow them to become more sophisticated. When modeling a computational framework that considers generative, exploratory, and adaptive behaviors, building blocks with rules and regularities for interactions must be included. Since most modeling techniques use building blocks or agents as the basis of a model, this level of abstraction emphasizes the difference between "top-down modeling," which uses "abstraction-based objects," and "bottom-up modeling," which uses "agent-based objects" with limited abstraction (Miller and Page 2007, pp. 65-67). Accordingly, the bottom-up structure of "agent-based objects" promises to include generative mechanisms with constraining approaches to explore the problem domain, while intrinsic and extrinsic self-organization processes organize the complex system to emergent complexities.

2.4 Agent-Based Design Computation

2.4.1 Introduction to agent-based modeling

It is difficult to model complex systems that consist of many active elements; more specifically, the complex behaviors of these active elements are difficult to formalize mathematically (Helbing 2012, p. 27). In particular, when modelers try to formalize a system that relies on its constituent elements to generate novel approaches. Modeling an integrative design has its own level of complexity. Especially, when the model acts as an exploratory system that integrates inclusive drivers. Examples of these drivers could include the ones that are captured from material and fabrication agencies. Developing a model that prioritizes constituent elements over the complexity of the entire system suggests a model based on individual building blocks, as an agent-based modeling system (ABM). Developing a model that relies on lower-level elements aggregates the micro-level of behaviors to generate the macro-level of orders (Schelling 1978, pp. 13-14; Epstein 2006, p. 7; Epstein 2008; Gilbert 2008, pp. 30-31). The orders describe the desired behaviors that modelers explicitly intend to exhibit. Generating a macro-level of orders requires an understanding of the significance of micro-level agencies in desired behaviors.

Unlike “Equation-based Modeling”[1] and “Differential Equations,”[2] agent-based modeling provides a computational method for modeling individual heterogeneities that are situated with an environment and that have the ability to decide upon their embedded rules (Gilbert 2008, p. 1). Modeling with agents furthers the study of different aspects of developing complex systems. These aspects include the emergent properties of Constrained Generating Procedures (CGPs) and the general regularities obtained via Complex Adaptive Systems (CAS). The simple configuration of a system that generates these properties requires an understanding of simulation and modeling with agents. Agent-based modeling consists of a collection of autonomous decision-making entities and a set of rules that govern their interactions with other entities in the model (Bonabeau 2002, p. 7280). According to Gilbert (2008, p. 2), interactions among these entities are established by situating agents within the environment. Understanding various types of agent-based modeling requires insight into the behavioral rules, the environment, and the communication of agents.

---

[1] see Helbing (2012, p. 27).

[2] see Bonabeau (2002, p. 7280).

Agent

The term "agent" is used in a variety of scientific fields; however, the term generally entails essential characteristics. For example, in economics, the term agent is characterized by autonomy and adaptability features. In this description, a Complex Adaptive System (CAS) is similar to the agent-based system (ABS). Figure 2.4.1 illustrates an adaptive agent interacting with other agents, in which the whole system demonstrates a Complex Adaptive System (CAS) (Holland 1995, p. 6). Holland and Miller (1991) describe a Complex Adaptive System (CAS) consists of a "network of interacting agents," which generate dynamic and aggregate behaviors. In a complex adaptive system, the behavior of the system is distinguishable from the behavior of individuals (Holland and Miller 1991, p. 365).

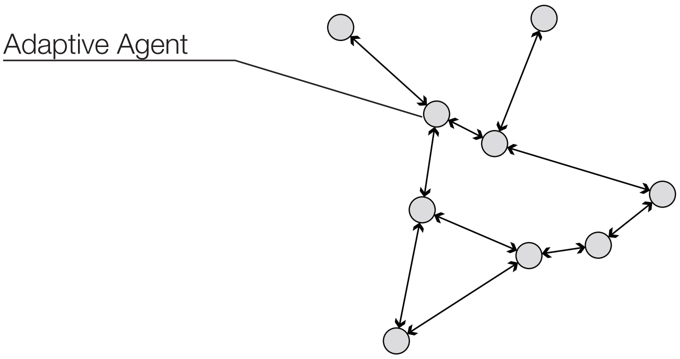


Fig. 2.4.1: A schematic representation of an adaptive agent in a Complex Adaptive System (CAS); adapted from Holland (1995, p. 6).

The behaviors of the agents arise out of a set of rules that define agents' strategies in confronting "perpetually novel" situations (Holland 1995, p. 35). Casti (1997a) states that intelligent and adaptive behaviors rely on embedded rules within agents. The early decision-making processes rely on primitive rules, and then agents are able to modify the basic rules with acquired information and generate new rules (Casti 1997a, p. 214). Dynamic modifications of the primitive rules provide system adaptations with perpetual novelty.

The frequent use of agent terminology necessitates a clarification of ambiguous uses of the term. The classification of agent terminology within different fields will clarify these ambiguities. From biological taxonomies, Franklin and Graesser (1997) classify autonomous agents (Figure 2.4.2) into three levels: (a) the "kingdom" level, which categorizes autonomous agents into "biological agents," "robotic agents," and "computational agents";

(b) the "phylum" level, which classifies computational agents as either "software agents" or "artificial life agents"; (c) the "class" level, which further classifies software agents into "task-specific agents," "entertainment agents," and "viruses" (Franklin and Graesser 1997, p. 30).

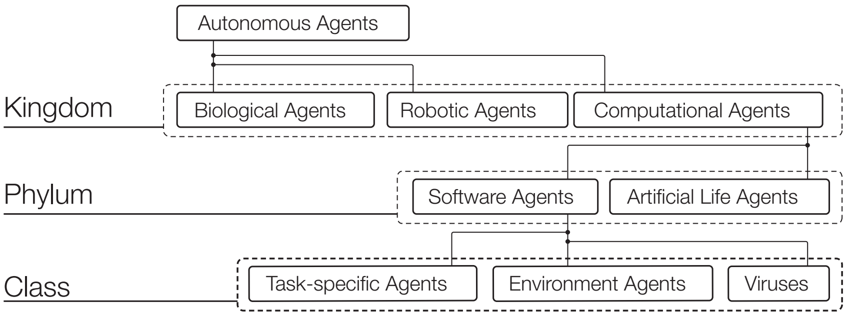


Fig. 2.4.2: Taxonomy of autonomous agents; redrawn by author based on: Franklin and Graesser (1997, p. 31).

Exploring the taxonomies of agents leads to the Pfeifer and Scheier's (2001) categorization, where autonomous agents are classified as "biological agents," "robotic agents," and "computational agents" (Figure 2.4.3). The latter is specifically used as "simulated agents," "artificial life agents," and "software agents" (Pfeifer and Scheier 2001, p. 26). A comparison between these two taxonomies separates the computational agents, used for simulating and modeling systems, from the robot agents, used to different approaches in industry.

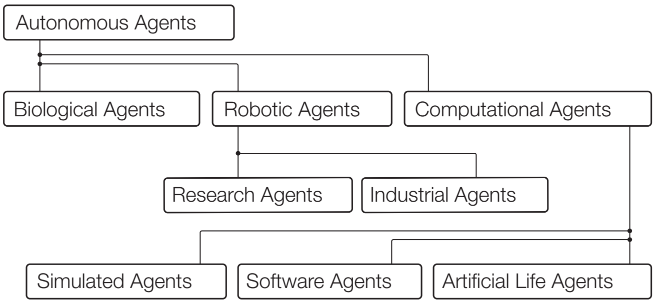


Fig. 2.4.3: A classification of Autonomous agents; redrawn by author based on: Pfeifer and Scheier (2001, p. 26).

Macal and North (2009) argue that the significance of agents in agent-based modeling is distinguishable from the significance of other agent classifications. For example, mobile agent systems are defined as proxies to perform user demands with some autonomous behaviors (Macal and North 2009, p. 88). An early example of these autonomous mobile agents is "Braitenberg vehicles" (Figure 2.4.4). Braitenberg (1984) developed a series of vehicles with a simple structure of sensors and motors. Even though the structure was simple, he was able to generate complex behaviors by differentiating "sensory-motor connections" (Braitenberg 1984). The idea implemented within Braitenberg vehicles had great significance on developing locomotion behaviors within computational agents.

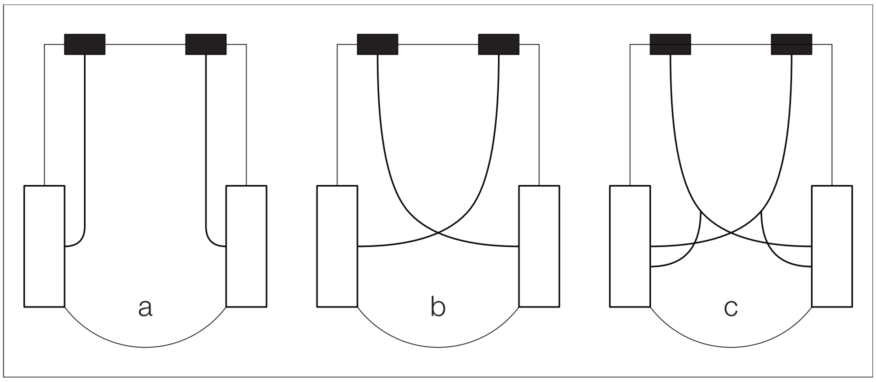


Fig. 2.4.4: Braitenberg (1984) developed mobile agents with two sensory systems and two motors with different connections of a, b, and c; redrawn by author based on: Braitenberg (1984).

Wooldridge and Jennings (1995) generalized the term agent in hardware and software-based computer systems. In these systems, the agent has certain features, such as "autonomy," "social ability," "reactivity," and "pro-activeness" (Wooldridge and Jennings 1995, p. 116). Wooldridge and Jennings's (1995) characterization show that the computational agents without external intervention (autonomy) interact with each other (social ability). And, based on their initiated goals (pro-activeness), they respond to the environment (reactivity) (Wooldridge and Jennings 1995, p. 116). Autonomous agents, which are "behaving systems," perceive the environment and then modulate it according to their internal mechanisms (rules) (Pfeifer and Scheier 2001, p. 25).

In computer science, computational agents, which are "encapsulated in computer systems," are active entities situated in the environment, not passive objects that are significantly affected by external factors (Jennings 2000, pp. 280, 283). According to Russell and Norvig (2003), agents will observe the environment through sensory mechanisms and allow actuators to manipulate their environment (Figure 2.4.5). This development is essentially an extension from software agents to robotic agents (Russell and Norvig 2003, p. 32).

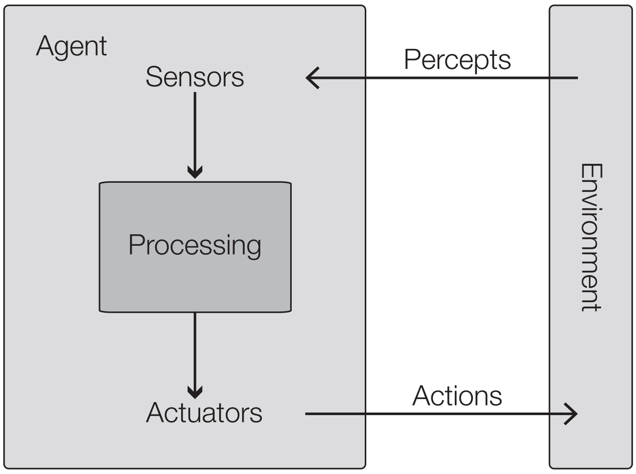


Fig. 2.4.5: A schematic diagram of interaction between agents and environments; redrawn by author based on: Russell and Norvig (2003, p. 33).

In the context of animation and game development, Reynolds (1999) proposes "autonomous characters" to virtually model the autonomous characteristics of real robots. He specified a new type of virtual autonomous agent that includes certain aspects of "situated," "reactive," and "embodied" characteristics (Reynolds 1999, pp. 763-764). Transferring the physical characteristics of real robots to virtual environments signifies the development of a computational model that abstracts the essential features of physical agent modeling. In addition, the agents are self-contained or discrete entities that have boundaries to separate their contents from other external entities (Macal and North 2009, pp. 87-88). The discretization of entities emphasizes the specific performance of agents that are accessible for agents by a set of behaviors, such as motion and interaction (Gilbert 2008, pp. 21-22).

Environment

In agent-based modeling, some kind of environment is needed for agent interaction. Without an environment to situate the agents, they will be incapable of performing the tasks dictated by their individual attributes. In the context of the computational agent, Gilbert (2008) indicates that the virtual world determines the environment. In this case, the environment is either a neutral element or an active participant in the process of modeling and simulation (Gilbert 2008, p. 6). The participation of an environment in the process of modeling establishes the conception of the environment as a static type of agent. In particular, Resnick (1994) proposes the discretization of the environment into square cells as "patches." These patch systems are capable of storing a specific range of information, either permanently or temporarily; for example, one could embed chemical signals at each patch (Resnick 1994, pp. 33-34). Figure 2.4.6 shows the relation between agents and patch systems. Patch systems provide packages of information for agents to improve their behaviors by processing extracted information. Accordingly, agents require sensory mechanisms to facilitate this process.

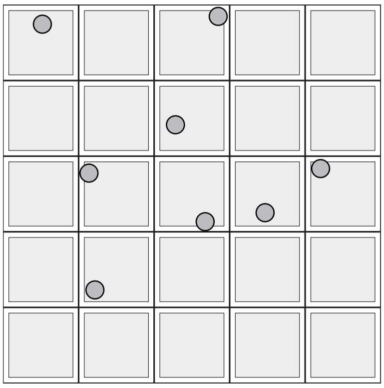


Fig. 2.4.6: Patch systems and distributed agents (turtles).

However, Holland (1995, 2000, 2010) denotes that a main characteristic of the environment is “perpetual novelty” with intrinsic diversity to avoid generating similar states (Holland 2010, p. 23). When autonomous agents search environments to find appropriate answers, the perpetual novelty of the environment necessitates versatile mechanisms that have to be adaptable. The versatility of agents emphasizes the behavioral character of agents that dynamically adapt to unknown situations. Therefore, considering the possible solution spaces within an environment leads agents to explore problem domains with limited knowledge. Agents must use an environment's embedded knowledge to discover solutions to the problem domains.

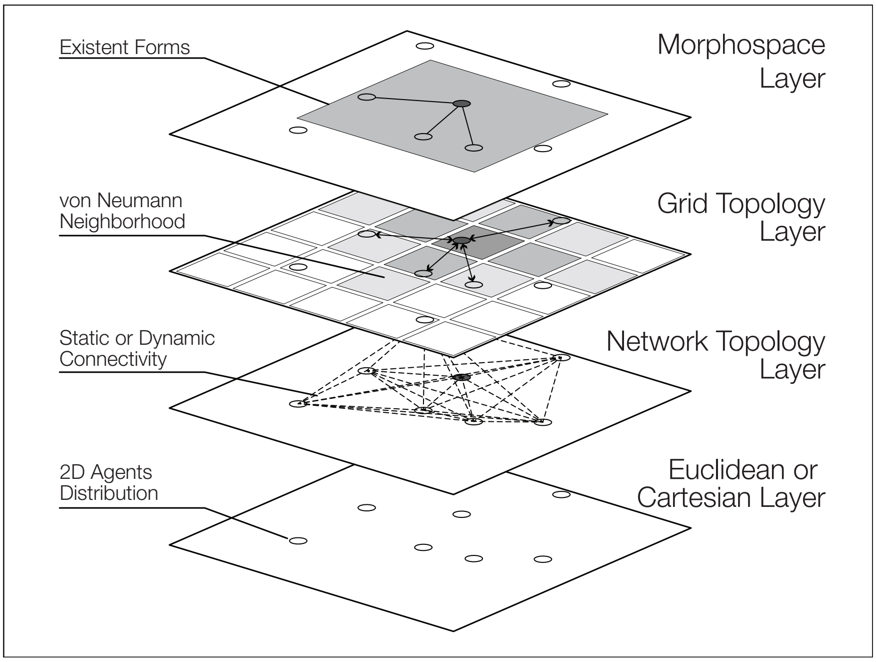


Fig. 2.4.7: Information space including different layers to enhance agents' behaviors. A schematic diagram of overlapping various layers; developed based on: Macal and North (2009, p. 94), Macal and North (2010, p. 152), and Torrens (2010, p. 430).

Extending the environment from a bilayer system to a multilayer system provides agents-based systems with a wide range of knowledge. For example, using the concept of a fabricational morphospace with an environment provides an additional layer of information. Accordingly, by exploring an environment, agents typically require different layers, such as morphospace layers, which enables them to consider different criteria for solving problems.

Figure 2.4.7 depicts an environment as an information space overlaid by various layers, such as topological and Euclidean spaces and morphospaces.

Interaction and communication

In agent-based modeling, the relation between agents and the environment is established through the purpose of the model. A model that consists of a vast number of agents requires an effective structure to determine their interactions and communications. The social ability of the agents permits a number of interactions between them. These interactions can be separated into *direct and indirect communications*. In direct communications, the software agents directly interact with other agents and in turn they exchange information through "communication language standards" to create "interoperable software" (Genesereth and Ketchpel 1994, p. 48).

In agent-based modeling, direct communication is associated with network topologies. Through these topologies, an agent will discover which other agents it will have direct interactions with. Figure 2.4.8 illustrates a different topological connectivity where the agent finds adjacent agents via orthogonal and diagonal visioning mechanisms. Direct communication includes different methods, such as "Euclidean space," "aspatial model," "von Neumann neighborhood," "geological information system (GIS) topology," and "Network topologies" (Macal and North 2009, pp. 93-94). In agent-based modeling, which has no direct communications with other agents, agents interact indirectly.

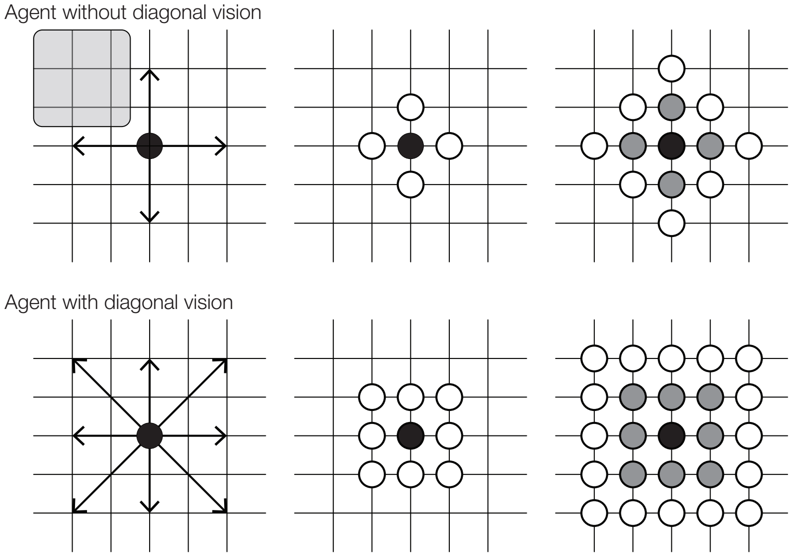


Fig. 2.4.8: Two different mechanisms for finding neighbors; redrawn by author based on: Epstein and Axtell (1996, pp. 24, 39-40).

Indirect communications require a medium, determined as an environment to facilitate interactions among agents. Indirect communications are observable phenomena within the nest building constructions. The communications through a "built structure" or built environment as a medium are called "sematectonic communications" (Wilson 1975, p. 186). Wilson (1975) proposed the term "sematectonic communications" to substitute "stigmergic communications." Stigmergy, introduced by Grassé (1959), as an indirect communication, which coordinates and regulates insect actions (Theraulaz and Bonabeau 1999). According to Bonabeau, Dorigo, and Theraulaz (1999), stigmergic communications are established through an environment that has been altered by an insect. Other insects then perceive the signals embedded in an altered environment and react accordingly (Bonabeau et al. 1999, p. 207). Stigmergic and sematectonic communications facilitate indirect interactions among agents through these embedded signals, but also by modulating environments as mediums. These communication techniques indirectly coordinate the processes of construction and foraging within the social insects (Figure 2.4.9). In addition, this type of communication can develop as learning and memory processes, which avoids the need to develop a memory system in a computational agent.

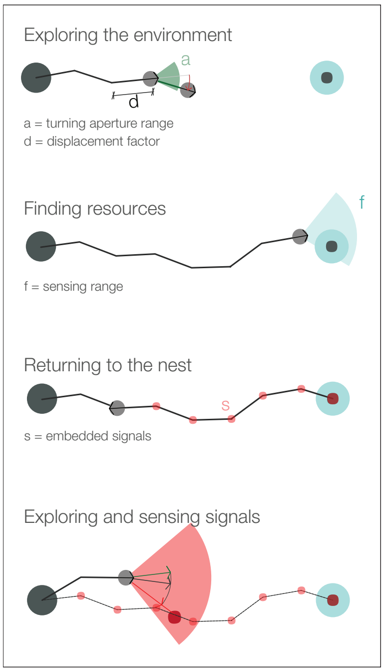


Fig. 2.4.9: A simple algorithm developed by Alvarez and Martinez (2015) represents the process of exploring the environment to find food resources. Embedding signals within the environment is abstracted from the stigmergic communications; source: Alvarez and Martinez (2015).

2.4.2 ABMs: Definition and background

Axelrod (1997) contends that simulation models, in the context of agent-based modeling, "is a third way of doing science." In it, simulation models are similar to conventional deductive and inductive reasoning; the agent-based modeling offers contributions to intuitive approaches (Axelrod 1997, p. 5). From an epistemological perspective, Epstein (1999) argued that the obligation to distinguish agent-based modeling from the "inductive" and "deductive" approaches of science. By referring to "syntactic theory," which was inspired by Chomsky (1965), he emphasizes "generative" approaches to grow "macrostructures" out of "microspecifications" (Epstein and Axtell 1996, p. 177; Epstein 1999, pp. 43-44). Therefore, ABM, as a "generative model," can describe the intuition within the model that is not simply formalized mathematically. Agent-based modeling also represents "unsimplified" behaviors, which are difficult to include in traditional models (Railsback and Grimm 2012, p. 10). ABMs can explain how interactions among individuals can generate nonlinear behaviors that are difficult to predict by scrutinizing individual behaviors (Ball 2007, p. 648). Considering rule-based modeling furthers interrelations among individual agents by demonstrating emergent properties, regardless of any macroscopic presumption (Helbing 2012, p. 29). The rule-based logics embedded within each individual at the microscale facilitate the development of heuristic methods in a computational model. However, the generated outcomes from this model are limited to the computational model, which is initiated with parameters that are abstracted from the real world. Despite these limitations, the outcomes should still satisfy the purpose of the model, while it tries to explain the macrostructures.

First wave

An example of a simple computational model that demonstrates agent-based modeling stems from "the theory of self-reproducing automata" (Von Neumann and Burks 1966). The self-reproducing machine is a universal Turing machine developed by von Neumann through a set of two-dimensional cellular system (Casti 1994, p. 221; Frazer 1995, p. 54). Accordingly, the lattices that the theory is structured on are interconnected networks of cells. The cells within this distributed network are proposed to act automatically and in parallel as cellular automata. Smith III (1976) considered parallel and interconnected automata under the theory of "polyautomata." The focus of this theory was the way in which "microautomata" form "macroautomata" (Smith III 1976, p. 405). In 1951, von Neumann simplified his proposed machine in the way that Ulam suggested, in which the simplified self-replicating automata CA requires 29 states for each cell and 200,000 cell configurations for each state (Wolfram 2002, p. 876).

Casti (1994, p. 223) stated that the main concern with the self-reproducing machine is simplifying the machine in a way that maintains the effectiveness of its self-reproducing mechanisms. Gardner (1970) called the two-dimensional cellular automata developed by John Conway, the game of life. The game included two colors for each cell and three rules of "survivals," "deaths," and "births" (Gardner 1970). The color (state) of each cell is dependent on the current state of the cell, the states of its adjacent cells and the rules of game. Overall, cellular automata consist of cellular lattices, states for each cell, and transition rules that are sensitive to the initial conditions (Frazer 1995, pp. 51-54). In this computational method, each of the cells can be described as an agent that will find their adjacent neighbors through a specific network topology, such as the Moore neighborhood (Macal and North 2009, pp. 89-90).

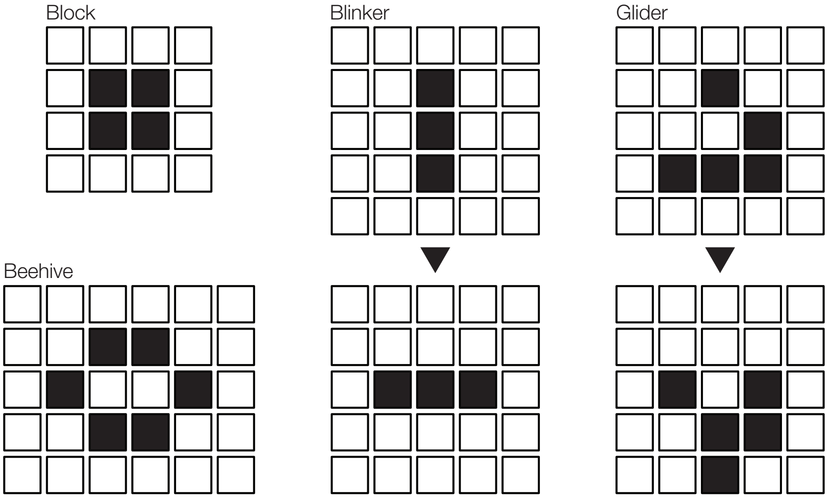


Fig. 2.4.10: The schematic diagrams of Conway's game of life.

The states of each cell are limited to the binary code of 0 and 1. The following states of the cells will evolve out of the transition rules. The main feature of cellular automata (CA), particularly the game of life (GA), is the emergent behaviors that manifest from the simple interactions among cells. Figure 2.4.10 illustrates a few patterns that emerge from Conway's game of life; these include, "block," "beehive," "blinker," and "glider" (Gardner 1970). Though the cellular system is static, the states of the system are dynamic. This is demonstrated by the glider pattern, which wanders across the grid of cells (DeLanda 2010, p. 41). This dynamism induces observers to perceive emergent patterns. Introducing cellular automata (CA) as agent-based modeling (ABM) requires a consideration of each cell as an agent with an isolated location. The generated emergence of a system is limited to the interactions among cells with simplified rules.

Second wave

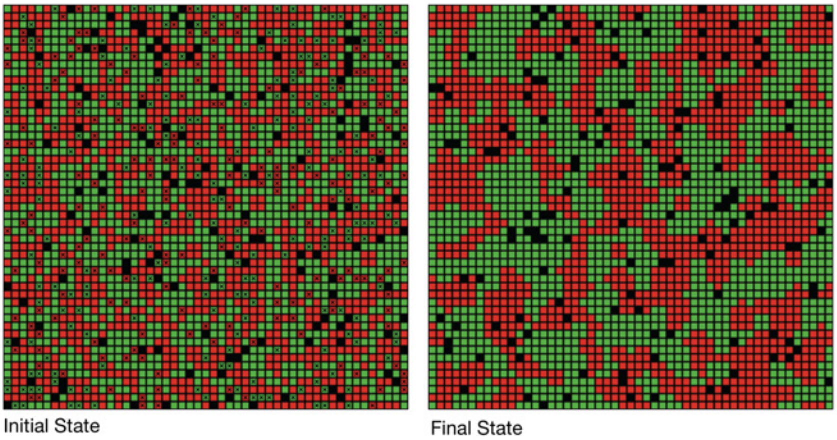


Fig. 2.4.11: A Segregation model; based on the model developed by Wilensky (1999b).

In agent-based modeling, cellular lattices simulate and model the behavioral aspects of different field of science. In particular, social science has a history of applying cellular systems to investigate social behaviors. An early example is the "checkerboard model" (Sakoda Model), which is a computational simulation to investigate "social interactions" (socio-interactions) between "two groups of checkers" on a social field defined as a checkerboard (Sakoda 1971). Along with the checkerboard model, Schelling (1971) developed a dynamic "Segregation model" (Schelling model), see Figure 2.4.11. In this model, the interactive dynamics embedded within the grid system exhibit unexpected phenomena (Schelling 1971, p. 143). Torrens (2010) concludes that both simulation models explore the polarization of "socio-spatial" segregation. Both simulation models also reach emergence behaviors at the "tipping-points" of the model (Torrens 2010, p. 435).

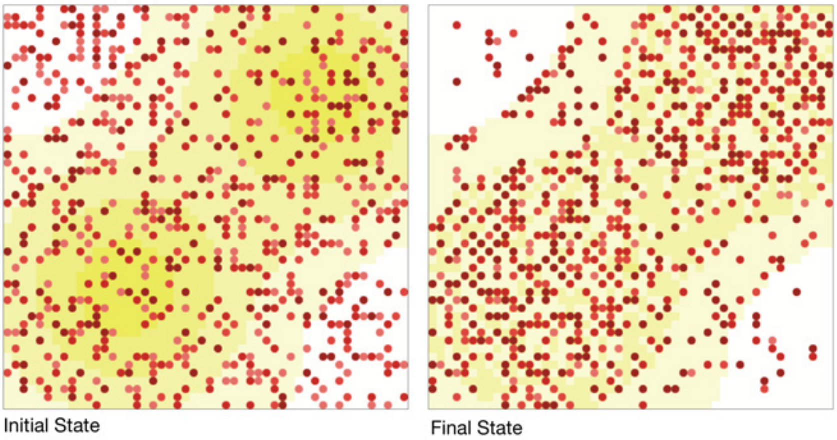


Fig. 2.4.12: A Sugarscape model; based on the model developed by Li and Wilensky (2009).

Epstein and Axtell (1996) developed an agent-based computational model, called the "Sugarscape model" (Figure 2.4.12), to "grow" an artificial society "from the bottom-up." In this model, the emergence of societal patterns evolve from interactions among human behaviors and resources embedded within the environment (Epstein and Axtell 1996, p. 6). The Sugarscape model is an agent-based model that combines agents and cellular automata (Epstein and Axtell 1996, p. 19). Cellular-based models, which define the environment, limit agent-based models to comply with specific topological networks. Examples include von Neumann or Moore neighborhoods. Figure 2.4.13 shows some sequences of finding closest and adjacent agents via a Cartesian method and Topological connectivity. Moore neighborhood algorithms are implemented with different radii to define different levels of agent participation.

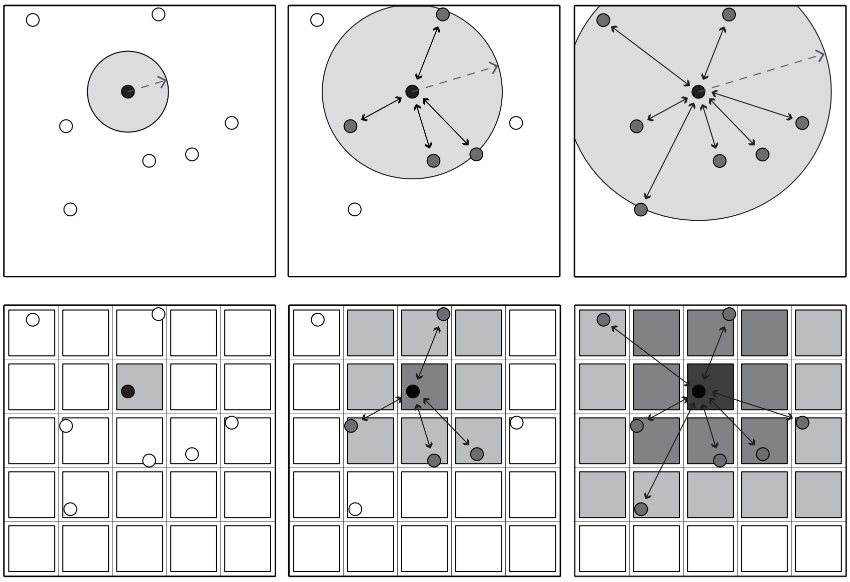


Fig. 2.4.13: Top row: Euclidean space used to find closest agents via a Cartesian coordinate system; bottom row: Topological space used to indicate adjacent agents through diagonal and orthogonal methods of Moore neighborhood.

The transition from cellular lattices to Euclidean space follows the simulation of the collective behaviors of insects and animals, such as a flock of birds or school of fishes. The pioneer of this model is Reynolds (1987, 1999) who developed the boids simulation to avoid individual predefinitions of animated characters, i.e., birds, within graphic animations. Reynolds' (1987) simulation follows the abstraction of locomotive behaviors of birds using motion equations. In the boids simulation, the displacement of each individual follows Newton's second law. The autonomous characters have specific rule-based mechanisms to displace their locations and also to aggregate their motion behaviors (Reynolds 1987). Accordingly, Reynolds (1999) classifies the locomotion behaviors in a hierarchy of “action selection (strategy, goals and planning),” “steering (determining paths),” and “locomotion (animation and articulation).” In this hierarchy, aggregation behaviors, such as “cohesion,” “separation,” and “alignment” (Figure 2.4.14), are more than the predefined rules (Reynolds 1999, p. 788).

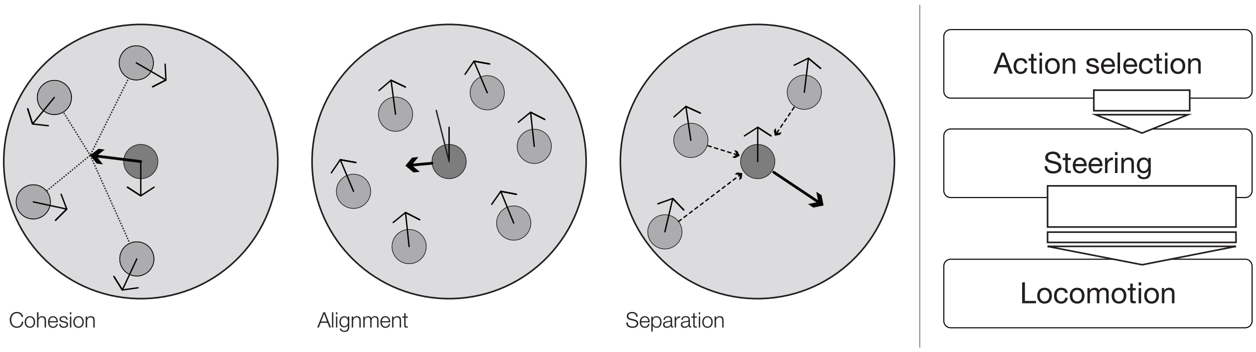


Fig. 2.4.14: Three behaviors of “cohesion,” “alignment,” and “separation” achieved through locomotive behaviors (Reynolds 1999); redrawn by author based on: Reynolds (1999, pp. 783-785).

2.4.3 ABMs: Benefits and ramifications

Features

The characteristic features of ABMs that distinguish these modeling techniques from others are derived from the institutional structure of the modeling, which is inherited from the agents’ aggregation. The interactions provided by the modeling techniques activate embedded rules within the agents to respond to external stimuli. This process characterizes ABMs as unique behavioral techniques. In addition, “stimulus-response” rules, as internal mechanisms, involve major adaptations to the environment. Bonabeau (2002) argues that the emergent phenomena exhibited by the interactions among agents are the main feature of ABMs. It is also important to note that the emergent phenomena feature other ABMs’ characteristics (Bonabeau 2002, p. 7280).

Exhibiting emergent phenomena from simple interactions among agents requires an understanding that local rules cannot lead systems toward emergence. For instance, DeLanda (2010) speculates that emergent behaviors in the game of life are secondary outcomes of cell interactions. Moreover, the collision of two emergent patterns, such as two glider patterns, generates a new emergent patterns that defines another level of emergent behaviors (DeLanda 2010, p. 41). Accordingly, agent-based modeling (ABM) has different approaches to regular modeling techniques, such as “differential equations,” in which ABM considers bilateral negotiations over the system and its individuals (Railsback and Grimm 2012, p. 10).

Considering individuals' behaviors in modeling allows agent-based modeling to unfold behavioral systems from the perspective of the agent (Bonabeau 2002, p. 7281). In this modeling system, the aggregate behaviors of agents' activities establish stable states that are robust enough to withstand the input of some stimuli (Miller and Page 2007, p. 46). These robust features allow ABM to retain its performance even if some of the individuals fail (Bonabeau et al. 1999, p. 7). Furthermore, ABM is adaptable to complex situations and to changing levels of system descriptions (Bonabeau 2002, p. 7281). On the other hand, the adaptation of agents in confrontation to complex situations fits within the generative capacities of agent-based systems. The satisfactory states of the agents rely on generating possibilities to find adaptive states. The adaptive states are the result of the agents' explorations, which are motivated by a desire to obtain a level of regularity. Eventually, the generative capabilities of agent-based systems lead Epstein (1999) to specify five features of agent-based models: "heterogeneity," "autonomy," "explicit space," "local interactions," and "bounded rationality." Through these features, it is possible to obtain "given regularities" from heterogeneous distributed systems (Epstein 1999, pp. 41-42).

2.4.4 ABMs: The basic

Types of agent-based models

Niazi and Hussain (2011) used the term "Agent-based computing" to describe the scientific domain that is covered by the term computational agents. Niazi and Hussain (2011) analyzed the sub-domains of agent-based computing through a "scientometric analysis."[1] In this analysis, agent-based computing is categorized into three sub-domains: "1) Agent-based, Multi-agent based or individual-based modeling, 2) Agent-oriented software engineering, 3) Agents and multi-agent systems in AI" (Niazi and Hussain 2011, p. 482). On the other hand, Torrens (2010) suggested that agent-based modeling (ABM) can be understood as "individual-based modeling" and "multi-agent models." Individual-based modeling scrutinizes the behavior of an individual agent and multi-agent models investigate the collective behaviors of discrete agents (Torrens 2010, p. 431).

---

[1] see Niazi and Hussain (2011, p. 496).

Individual-based modeling is a technique that is applied in ecology, e.g., animals' population (Grimm et al. 1999, p. 276). Individual-based modeling is a bottom-up approach that is established through a set of "unique and discrete entities" (Grimm 1999, p. 130; Grimm and Railsback 2005, pp. 9-10). The multi-agent model benefits from a diversity of tasks for agents (Torrens 2010, p. 431). In multi-agent models, the decision-making processes of individuals are more influenced by other agents. In complex multi-agent systems, agents are structured simply to the extent that they avoid any further complexity within the model (De Wolf and Holvoet 2005, p. 11). In contrast to Complete Complex Agents (CCA), behaviors of agents are dependent on other agents within the multi-agent model, but not vice versa (Bryson 2003, p. 61).

The simplification of agents within multi-agent systems requires a model of a complex system that will exhibit emergent behaviors (De Wolf and Holvoet 2005, p. 11). Therefore, modeling complex multi-agent systems with coherent behaviors necessitates a combination of self-organization and emergent phenomena, in which self-organizations increase the order within the system to exhibit emergent properties (De Wolf and Holvoet 2005, p. 11). Regardless of the type of agents, the interrelation between self-organization and emergent phenomena requires to formulate algorithmic mechanisms to exhibit emergent phenomena.

Building ABMs: Attributes and behaviors

Building an agent-based model requires an understanding of the structure of agents and their interrelations to the environment. Anderson (1999) described a model of agents with schemata that use cognitive structures to indicate an agent's behavior at time t, in response to its perception of the environment (Anderson 1999, p. 219). Accordingly, the states of agents emerge out of a set of rules that are activated by a series of inputs. At each moment, agents receive inputs from their environment. These inputs are used to decide which rules will be used to generate new states for the agents. Torrens (2010) formalized this sequence with a simple ABM, in which the ABM consists of "agent-automata" with a "state-rule-input architecture" as:

$$A \sim \{S, R, I\} \sim \begin{cases} S = S^1, S^2, \ldots, S^k_{i,t} \\ R : \{S_t, I_t\} \to S_{t+1} \end{cases} \tag{2.4}$$

, where A is an automaton, S is the state, I is the input, k_{it} describes attributes of S, and t is time (Torrens 2010, p. 432).

A simple example for illustrating this equation is Conway's Automata (Figure 2.4.15). In this equation, each automaton as a cell-automaton at time *t* has eight inputs as $I = \{I_1,\ I_2,\ \cdots,\ I_8\}$ with two states of $S \sim \{S^1, S^2\} = \{0, 1\}$ and two transition functions as schemata *R* to consider agent's agency (Holland 2000, p. 137).

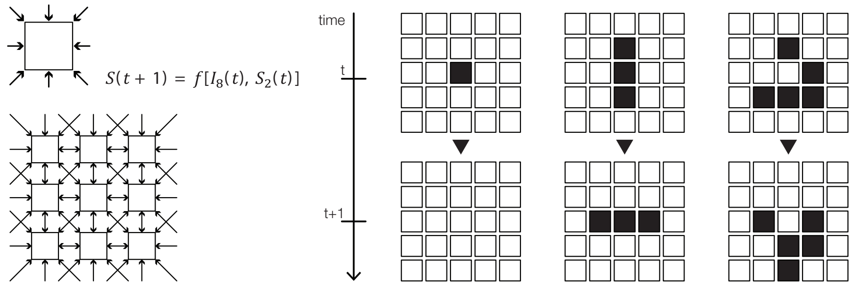


Fig. 2.4.15: An example of formalizing ABM, such as Conway' game of life through CGP; redrawn by author based on: Holland (2000, p. 137).

Towards an agent's agency

An agent's agency at time *t* can be considered by micropatterns that emerge from low-level system modeling. From Minsky's (1988) point of view, the model, at the level of agency, should consider macro-regularities. While, at the level of agent, the model should only consider micro-behaviors with little regard for the configuration of the whole model (Minsky 1988, p. 23). Agency indicates an ultimate purpose for modeling. Agents are required to accomplish this purpose with low-level behaviors. Approaching the level of agency considers two methods of organization: top-down agency (Minsky 1988, p. 23) and bottom-up agency (self-organization), see Figure 2.4.16.

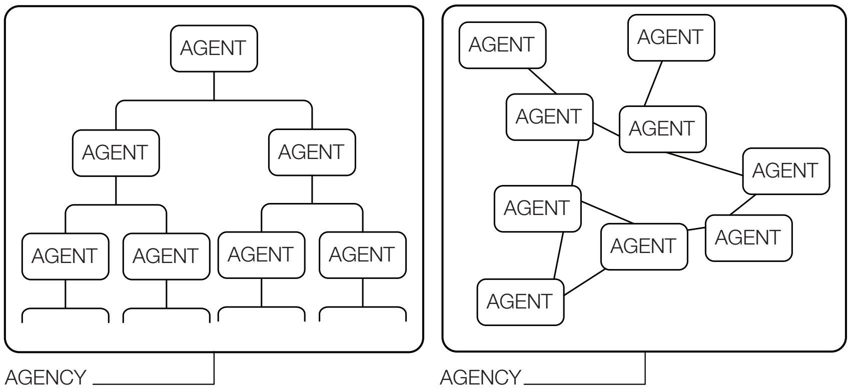


Fig. 2.4.16: Comparing two different agencies: Left diagram represents the top-down organization of an agency; redrawn by author based on: Minsky (1988, p. 23). Right diagram illustrates the bottom-up organization (self-organization) processes among agents to erect agency.

Schelling (1978) considered "the behavior characteristics of the individuals" at the micro-level of aggregation and emerge as macro-phenomena at the higher-level of aggregation (Schelling

1978, pp. 13-14). Epstein (2008) described a model of agents as a generative explanation of macro-phenomena that are exhibited at the macroscopic level (Epstein 2008, para. 1.10). Epstein (2006) argues that "macroscopic phenomena"[1] emerge out of three steps: (a) situating a set of autonomous heterogeneous agents in the environment; (b) preparing interactions among individuals and their environments based on simple rules; (c) letting the system generate regularities at the macroscopic level (Epstein 2006, p. 7). According to Schelling (1978), the macro-specification that is generated is not a simple aggregation of micropatterns. Extrapolating the macropattern is comprised of network of interactions "between individuals and their environment, that is, between individuals and other individuals or between individuals and the collectivity" (Schelling 1978, p. 14).

Squazzoni (2012) considered a generative explanation model using a comparison between the simulated pattern and the empirical patterns. Squazzoni (2012) suggested that a set of individuals or agents with possible states (micro-specifications) of $S_a, S_b, S_c, \cdots$ denotes the state of each agents at time t as $S_a = S_a^1, S_a^2, S_a^3, \cdots, s_{a,t}^k$. In the context of "generative explanation," Squazzoni (2012) explained that if a system has a macropattern of K_r, then there will be a possible combination of micro-specification (states) of agents, such as $K_a = \left\{S_a^2, S_c^1, S_d^3, \cdots, S_n^5\right\}$, which generates a macropattern K_a. Therefore, if the simulated pattern K_a is similar or at most equal to the "empirical pattern" K_r then the model has "sufficient generative conditions" to be considered a "generative explanation" model (Squazzoni 2012, pp. 11-12). Through this model, it would be possible to formalize a relation between the micro-level of agents and the macro-level of the system.

2.4.5 ABMS: Designing and programming

In the context of agent-based modeling, a properly designed model can generate different approaches to solve the modeled problems. A proper model requires appropriate parameters and values from different behavioral factors to establish a promising agent-based system, which will lead designers towards solutions. Networks of procedures, which link initiated agents at the beginning of simulation to equilibrium states at the end of simulation, require insight into the reasoning behind the obtained solutions. Dealing with behavioral systems described by soft parameters outlines the relation between agents' behaviors and stimuli factors. Understanding the rules that

---

[1] Epstein (2006, p. 7) considers "macroscopic explanandum" as "a regularity to be explained."

trigger an agent's reactions requires a development of agent-based systems as "tractable" tools with sufficient details (Bonabeau 2002, p. 7285). According to Bonabeau (2002), the tractability of agent-based modeling enables designers to monitor the behaviors of the models and to comprehend the relation between abstraction levels and exhibited behaviors. For instance, modeling geometrical behaviors relies on the geometrical descriptions of agents.

Gilbert (2008) proposes modeling an agent-based system in the sequence of specific "research questions," "macro-level regularities," and "micro-level behaviors" (Gilbert 2008, pp. 30-31). Associating the purpose of the model with the research question necessitates the comprehensive understanding of the phenomenon being studied. Specifying the research question enables the identification of the regularities of the phenomena that are observable by modelers. A decomposition of the macro-level regularities into simplified elements and rules promotes the development of organized complexities. When a model is developed to address a specific research question, behaviors that are more than macro-level regularities emerge. Two-dimensional cellular automata, for instance, exhibit a variety of emergent patterns. The glider is an example of the emergent noises of agent-based modeling (ABM). Engineering the micro-level behaviors within agents extends the macro-level regularities beyond the research purposes, while answering the research question represented by the potential areas of the model.

The relationship between the macro and micro levels relies on effective factors or, more specifically, the properties of these factors, to lead the model toward the desired regularities. Embedding these factors into an agent-based model commences with the determination of different types of agents. According to Macal and North (2009), these types include decision-maker units and their behaviors. In this context, behaviors could be considered heuristic behaviors (anchoring and adjustment) or formal behaviors (Belief-Desire-Intent (BDI)) (Macal and North 2009, p. 92). BDI emphasizes the role of agent as a decision-making unit. This means that the agents decide which response should be triggered by the environment stimuli and the other agents. The decision-making processes of agents underlies the behaviors of either condition-action rules (Gilbert 2008), or Belief Desire Intentions (BDI) (Rao and Georgeff 1991). Both of these methods determine the behavioral rules by which agents interact with other agents and environments. The agents' interactions enable the model to obtain the desired regularities in collective or individual manners.

The last stage of developing agent-based models is the calibration and validation phases (Epstein 2008; Gilbert 2008; Torrens 2010; Railsback and Grimm 2012). These phases are required for every relational model. Validating a relational model follows an evaluation of simulated results with the generated states found in the real world. A comparison between these two states enables a consideration of the model's validity. It is important to note that once the model is deemed valid, it is extendable to different variables and parameters. Validating an agent-based model is difficult because of the lack of quantitative values that facilitate a comparison between the computational model and the real-world (Torrens 2010, p. 437). For example, Reynolds (1987) expresses that a validation of the flocking algorithm "is difficult to objectively measure," even though the visual similarities between the flocking algorithm and a flock of birds enables observers to recognize the resemblance between the simulated model and the natural model (Reynolds 1987). In the context of social science, validating an agent-based model means that the micro-level behaviors and interactions lead the model toward desired macro-level regularities (Gilbert 2008, p. 31).

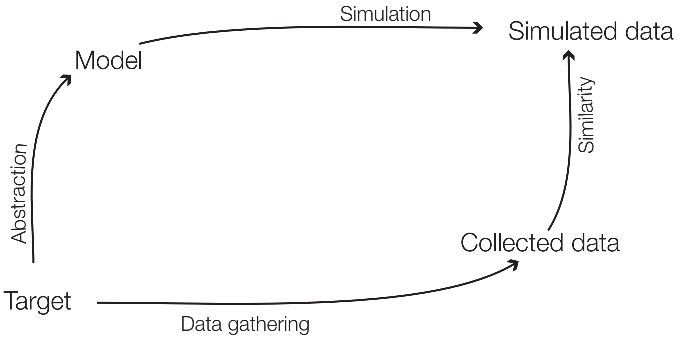


Fig. 2.4.17: A diagram of simulation method; redrawn by author based on: Gilbert and Troitzsch (2005, p. 17).

Scrutinizing the developed agent-based model follows internal (implicit) verification and external (explicit) validation. Gilbert and Troitzsch (2005) note that to verify a model, a program must be proven to work properly (debugging). And, if a model is to be validated, its simulated behaviors must correlate with the desired behaviors, see Figure 2.4.17 (Gilbert and Troitzsch 2005, p. 23). Furthermore, Epstein (2008) argues that validating an explicit model, in contrast to an implicit model requires a calibration of the model and the feasibility of "sensitivity analysis." With the former, the model is tested against a set of existent data or empirical data and with the latter, the values of variables and parameters are replaced with a different set of parameter values to identify the robustness and feasibility of the model (Epstein 2008). In addition, validating an agent-based model requires two levels of relating theory: (a) via sensitivity analysis; (b) via a comparison with the simulated model and empirical data, similar to the regression equation (Gilbert 2008, pp. 44-46). When the output of the model is comparable to the empirical data, the model can be extended to explain the significance of micro-level variables in generating macro-level behaviors.

3 Current State: Agent-Based Systems in Architectural Design

3.1 General Areas of Applications

The development of agent-based modeling (ABM) provides insight into the behavioral aspects of generative systems, where conventional mathematical models are incapable of describing complex behaviors. As mentioned, two waves of ABMs are applied to model and simulate different aspects of behavioral systems, including simulating human behaviors and modeling animals' actions. Helbing (2012) asserts that the study of complex behaviors provides a level of visual representation and a level of virtual simulation, in which the former tries to mimic the behaviors of humans, animals, or insects and the latter tries to simulate their individual and collective behaviors. Considering the different levels of detail in modeling differentiates ABMs into developing computer games and scientific applications, in which the main differences between these two approaches emphasizes realistic visualization and accurate simulations (Helbing 2012, p. 28).

Computer game visualizations insist on apparent details instead of generative interaction among individuals. However, the development of scientific applications considers the generative explanatory aspects of ABMs to explore and explain the emergent properties of phenomena. From a scientific perspective, the classification of agent-based modeling considers "physical models," "economic models," and "sociological models," in which each model reflects the behavioral aspects of individual constituents (Helbing 2012, p. 28).

In the context of behavioral systems, considering ABMs involves a range from social behaviors of human and insects to physical and ecological systems (Macal and North 2009, p. 91). In social science, Gilbert (2008) classified the agent-based modeling applications into "urban models," "opinion dynamics," "consumer behavior," "industrial networks," "supply chain management," "electricity markets," and "participative and companion modeling" (Gilbert 2008, pp. 6-14). In the realm of business and economy, investigating emergent phenomena helps Bonabeau (2002) to prioritize agent-based modeling over other modeling techniques, such as "mathematical modeling techniques" or "statistical analysis." In accordance with emergent behaviors, he extended agent-based modeling to four main categories: "flows," "markets," "organizations," and "diffusions" (Bonabeau 2002, pp. 7281-7283). For example, "flow" is a general term to model and simulate human behaviors in different situations, where the classification of "flows" describes the "evacuations" at the panic moments or the "flow managements" at the traffic jam (Bonabeau 2002, pp. 7281-7282). In each one of these classifications, agent-based modeling considers an agent as an individual, an organization, or even a biological system. The implementation of this model provides to investigate on individual behaviors and collective behaviors. Accordingly, the use of ABMs fosters the development of a behavioral framework to comprehend the effects of communications and behavioral parameters. Therefore, this computational model provides insight into the fragile and stable states of the system. It also recognizes the effective parameters that affect the behavior of the model.

3.2 Categorization

3.2.1 Preamble of agent-based systems in architectural design

In the realm of architectural design, agent-based systems have been investigated to find the behavioral potentials of ABMs' application in design processes. For example, ABM was considered in developing a design tool based on "situated learning" in order to embed the knowledge of experts in design processes (Gero and Nath 1997). The development of agents within situated design tools was extended by implementing a "constructive memory" systems to enhance the design processes (Liew and Gero 2002a,b). The development of agent-based design tools were studied further to consider cognitive sciences and learning mechanisms. For example, Moss et al. (2004) enhanced an existent application (A-Design) with a learning mechanism to examine learning mechanisms within agents to demonstrate that agents can learn from their previous experiences and that they are capable of extending their experienced knowledge to other situations (Moss et al. 2004). This process of adapting agents with different design problems through abstracting and translating designer knowledge to agent-based systems was extended to develop a “situated agent-based design assistant” (Gero and Peng 2004). Accordingly, the development of architectural applications tries to embed designers’ knowledge into agent-based systems.

On the other hand, agent-based applications were used to simulate and model human behaviors in architecture and urban spaces. These modeling techniques were investigated to gain better insight into human behaviors when confronting normal and abnormal conditions, such as pedestrian behaviors (Fatah gen. Schieck, A. et al. 2004; Kitazawa and Batty 2004; Karunakaran 2005; Chen and Chiu 2006; Chen 2009a,b; Shih et al. 2009; Moussaïd et al. 2011) and evacuation behaviors (Chen and Lin 2003; Sharma and Turner 2004; Sun et al. 2007). However, agents-based systems are required to study the context of generative design approaches to cover the process of formation and materialization. The applications of agent-based systems are reviewed based on the proposed design agencies, such as environment effectiveness, fabricational morphogenesis, and performative criteria. Since these agencies categorize agent-based design systems into three main categories: environmental and spatial effectiveness, performative and structural approaches, and fabrication approaches. Each of these categories provides a better description about the significance of agent-based systems in design processes. Categorizing existent examples of agent-based applications fosters appropriate strategies to develop a computational design tool. The computational tool requires an amalgamation of three active agencies to simultaneously materialize the process of formation. Eventually, generative agent-based design computation will blur the existing gaps in design processes with behavioral strategies.

3.2.2 Environment/Spatial effectiveness

Infrastructure layout planning and urban spatial structure

In the context of planning infrastructure, utilizing an agent-based system enables designers to investigate the behavioral influences of spatial structures on human behaviors and vice versa. Investigating bilateral relationships between humans and their relevant environments are essentially effective on the formation of spatial organization. The significance of the environment is defined by internal and external factors in the development of infrastructures and urban spatial structures. Therefore, the effectiveness of the environment is considered through parameters within the formation of spatial structures. Those effective parameters include the structural correlation among humans, their built environments, and the environments. Investigating this effectiveness enables an understanding of the importance of behaviors on emerging forms and spatial organizations.

Modulating the environment through agents' activities employs two methods. In the first method, agents directly change the environment that might be a simple geometry, such as a surface. In this method, agents locally modulate the environment by changing the geometric definition of the surface. The agents could accomplish this by relocating knots of NURBS surfaces or vertices of mesh surfaces. In the second method, agents employ some mechanisms to embed signals or instances on the environment that indirectly facilitates agent systems to adjust their behaviors to the environment. Therefore, agents' behaviors rely on storing data, such as built environments or signals of their activities, and extracting embedded signals from the environment. In this sense, agents modulate the environment to transfer their activities to other agents and themselves via embedding signals. The extraction of these signals from the environment enables an understanding of the main areas of agents' activities. Both of these methods of constructing elements or embedding signals emphasize stigmergic and sematectonic communication approaches. The environmental modulations facilitate mediation between the agent's preceding actions and their upcoming behaviors. In that case, the stigmergic method transfers local modifications to the agent itself and to the other agents. Accordingly, extending this concept to the human activities within urban planning helps designers to configure the spatial circulations.

A series of experiments have been developed to investigate these bilateral relationships between human behaviors and their contextual environments. For example, Eleni et al. (2002) studied the mediation between human behaviors and their contextual environments to investigate mutual influences among them. Eleni et al.'s (2002) research project considered the environment as a series of cubic blocks, in which the behaviors of agents change the surface configuration. In the next iterations, these surface modulations affected the agents' behaviors. From these local modifications, they proposed a stigmergic approach out of indirect communications among agents, where local modifications changed the global form of the environment (Eleni et al. 2002).

Confronted with constructed environments that effectively change human behaviors was translated into generative computational models. Considering human behaviors as generative factors provide topological diagrams for forming spatial structures. In a series of experiments, Ireland (2008, 2009, 2010) investigated a generative process to translate the "diagram of architectural space" to the "diagram of spatial organization," in particular, this model tried to address the problems of circulation in architectural organizations via Swarm Intelligence (SI). Extending the ant foraging algorithm to architectural circulation fostered the development of generative algorithms, in which the proposed model considered interactions between humans and the constructed environments to determine circulation paths, see Figure 3.2.1 (Ireland 2008). Accordingly, Ireland's (2008) generative model offered an approach to generating particular configurations as the diagram of spaces. These configurations emerged from the behavioral interactions between architectural definition of spaces and path connectivity developed by agents' circulations (Ireland 2008, 2009, 2010). The advancement of this translation helps designers to consider the layout planning as the production of agents' negotiation with the constructed environment. This interaction fosters generative models to use architectural properties as generating procedures to produce alternatives. In this sense, Hao and Jia (2010) developed a computation tool for exploring the layout planning based on "bubble diagrams." In Hao and Jia's (2010) project, they linked the floating bubble systems to agent systems (Figure 3.2.2). The behavioral rules relied on architectural specifications described through the topological connectivity of architectural spaces (Hao and Jia 2010).

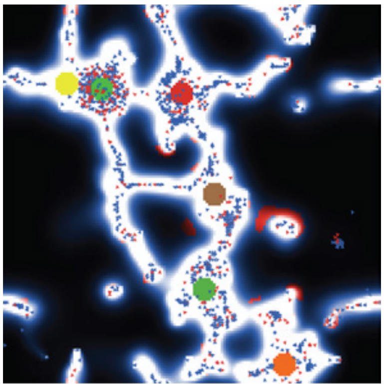


Fig. 3.2.1: The simulation of the diagram of spaces; source: Ireland (2008).

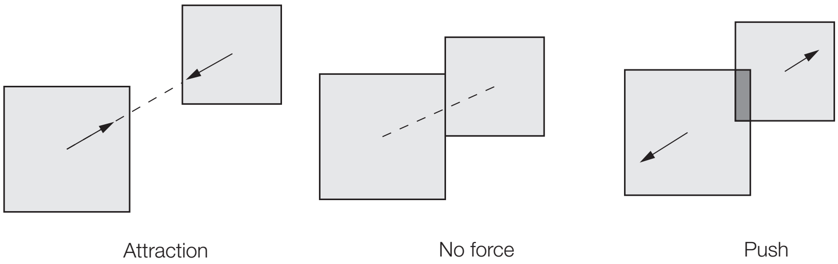


Fig. 3.2.2: The behavioral rules of interacting "bubble diagrams"; redrawn by author based on: Hao and Jia (2010).

Spatial organization and configuration

The design process emphasizes both knowledge-based systems and behavior-based systems. The comparison between these two approaches determines the importance of agent-based systems as behavior-based systems, from which the interactions among agents demonstrates the self-organization process of form generation. The generative agent-based systems, which include different constraining mechanisms to inhibit generated alternatives, might lead behavioral formation towards exhibiting emergent patterns. In this sense, Krause (1996, 1997) described a method to implement a Behavior-based Artificial Intelligence (BBAI) within a computational design tool. Krause's (1996) method considered modulating geometric constituents to generate behavioral forms. Krause's (1996) approach contrasted with the conventional architectural design approach from which the top-down organization of a knowledge-based system was converted to a behavior-based system. In Krause's project, the generative agent-based system was developed through different types of agents, including varieties of behaviors and communications. This project, as one of the early projects, introduced behavior-based approaches through agent-based modeling to the field of architectural design to differentiate this attitude from a knowledge-based system of design (Krause 1996, 1997).

The approach of behavior-based systems followed abstracting physical agents into computational agents. Integrating these processes with an architectural application opened a new chapter in architectural design. Coates and Schmid (1999) elaborated on a series of students works, developed at the CECA[1], with which to utilize different computational techniques, such as cellular automata (CA), Genetic algorithms (GA), and swarm algorithms in architectural and urban design. Coates and Schmid's (1999) investigations included different computational tools, such as StarLogo (Resnick 1994, pp. 31-35) and CAD applications through which they initiated the two-dimensional cellular systems for investigating urban design and three-dimensional CAD agents for representing the interactions between agents and their surrounding environments. Particularly in their proposed CAD agents, the interactions between agents and environment (model) were categorized in three feedback loops "agent-agent," "agent-model," and "model-model" (Coates and Schmid 1999). In one of Coates and Schmid's (1999) works, the agents were facilitated with visionary systems to sense the virtual models, and, in accordance to their rules, the agents modified the NURBS-surfaces of model. In this example, the agents modulated the environment behaviorally to accomplish their embedded tasks, in which emergent properties arose from their behavioral modifications (Coates and Schmid 1999).

---

[1] Center for Evolutionary Computing in Architecture (CECA) at the University of East London School of Architecture.

In other works, the boids algorithm was adopted within architectural applications, such as CAD applications, through which behaviors of agents provided different behavioral patterns. Carranza and Coates (2000) emphasized the flocking algorithm to relate the constituents' element of formation to the agent's behaviors, through which behavioral patterns emerged simultaneously out of translating agents' coordination in three-dimensional space to the geometric constituents of form. In addition, Carranza and Coates (2000) utilized the flocking algorithm in both coupling with environments and coupling with another system (Figure 3.2.3). The process of their linkages was structured with indirect and direct methods of communication with environments (Carranza and Coates 2000). In comparison with morphogenetic movements, the process of behavioral formation was adapted to environmental conditions or other systems, while agents were exploring the possible solution space. In addition to different evolutionary systems, their investigations on the learning approaches extended the adaptability of flocking systems with different conditions (Carranza and Coates 2000).

Fig. 3.2.3: The generated traces based on interactions between agents and the environment; source: Carranza and Coates (2000).

Although coupling with other agents and the environments enhance the agents' behaviors to generate more meaningful forms, the emergence of forms requires more designers' contributions to the process of behavioral formation. Snooks (2012b) elaborated a methodology for developing forms based on instable generative systems. Snooks (2012b) postulated that the instable state of forms, which is vulnerable to stimuli from the environment, necessitates the utilization of a computational method to intelligently adapt the system to new conditions. Accordingly, Snooks (2012b) used the intelligence of swarm systems that relied on multi-agent algorithms to generate forms from which the process of "behavioral formation" was developed by translating architectural aspects into behavioral rules. Therefore, Snooks (2012b) related the behavioral formations to the geometrical constituents within computational design tools, by which the agents' responses to the environment were influenced by architectural specificities. In addition to the behavioral formation, Snooks (2012b) argued that the limitation of local interactions at the micro-level was distinct from the macro-level. He proposed to inform the micro-level with structural properties, which occurred at the macro-level, through "messy computation" or "behavioral structural formation," which commences a negotiation between derived forms and forms' drivers (Snooks 2012b).

3.2.3 Performative and structural approaches

The manifestation of forms is associated with performative criteria, such as structural properties and functional rules. The implementation of different algorithmic rules facilitates the process of form generation with self-organization approaches. The implications of algorithmic rules to self-organize forms indicate simulations of virtual form-finding.

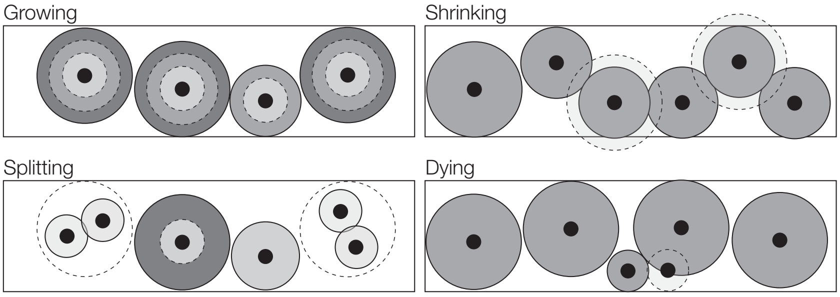


Fig. 3.2.4: The behavioral rules for column-like-agents; redrawn by author based on: Scheurer (2007).

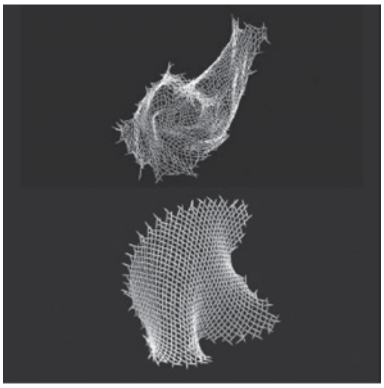


Fig. 3.2.5: The development of an agent-based system to simulate the Zollinger system; source: Tamke et al. (2010b,a).

For example, in Groningen Twister (2003), the construction of a parking areas for bicycles, benefits from the agent-based modeling where the agents were considered the columns (Scheurer 2003). Scheurer (2003, 2005, 2007) indicated that embedding structural and functional properties within the agents fostered a CAD application to self-organize the columns' positions. The negotiations among column-like agents and the environment followed behavioral rules (Figure 3.2.4), such as "growing," "shrinking," "splitting," and "dying" to lead self-organization towards stable states (Scheurer 2003, 2005, 2007). In the context of agent-agent interactions, another example of integrating structural performance is the "lamella flock." Tamke et al. (2010a,b) developed a computational tool based on the traditional wooden Zollinger (lamella) system, where each one of beam elements were determined as an autonomous unit to self-organize a structural system (Figure 3.2.5). Abstracting the principle of lamella systems fostered a set of behavioral rules to sequentially interlock agents (Tamke et al. 2010a,b).

Consequently, agent-agent interactions provide a self-organized condition for structural elements. In addition, agent-environment interactions facilitate another level of integrating performative criteria within generative agent-based systems. Durmazoglu et al. (2008) developed the "DROP" application to analyze the performative criteria of free-form surfaces by considering the rain-flow analysis method, which calculates the "flow of forces on a surface," including "loads on the free-edges," "bending moments," and "drain curves" on the surface. Durmazoglu et al.'s (2008) application associated the agent-based systems with the surface geometry. The agents' responses to the flow of forces altered the form-finding process of generating a free-form surface (Durmazoglu et al. 2008).

Moreover, the interaction with environments can provide agents access to the necessary information about structural performance. In this sense, the environment as a target surface informs the structural properties of the surface to the agents. Therefore, agents' interaction with the structural information requires different behavioral rules to self-organize their actions. For example, Tsiliakos (2010, 2012) developed a multi-agent system, which was based on Swarm Intelligence (SI) to integrate material systems (fibrous materials), structural properties of a target surface, and the environmental factors, such as sun radiation. In Tsiliakos's study (2012), the three inputs of a target surface, principal stresses, and solar radiation were defined as stimuli to trigger the multi-agent system through which agents' responses provided a level of adaptation between the fibrous system and the target surface (Figure 3.2.6). Therefore, the performative criteria, such as environmental effectiveness and structural properties were implemented within a multi-agent system, while the agents' actions on the surface eventually generate trail paths that determined the fiber placement positions (Tsiliakos 2010, 2012).

Fig. 3.2.6: The generation of fibers on the target surface based on performative criteria; source: Tsiliakos (2012).

3.2.4 Fabrication approaches

Developing applications that consider fabrication processes within the process of form generation require an understanding of the correlation between geometric constituents of form with fabrication tools. Therefore, the investigation on the geometric definition of form requires the interpretation of forms into generative algorithms, such as cellular automata (CA), L-systems, and the boids algorithm. On the other hand, the adaptation of these algorithms to agent-based systems fosters a possibility to attribute agents with the necessary knowledge of fabrication tools. For example, Anzalone and Clarke (2003, 2004) developed a series of experiments to integrate the design process with structural and fabrication systems. Anzalone and Clarke's (2003; 2004) design process was developed based on cellular automata (CA) and the boids algorithm to generate a free-form surface (Figure 3.2.7). This process was adopted to the structural properties of a truss system, which were applicable

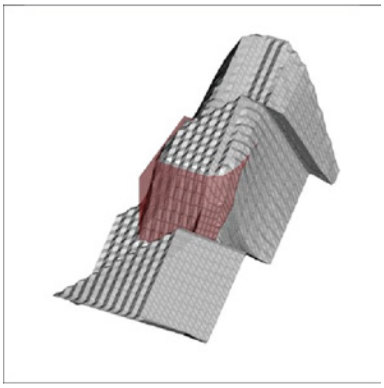


Fig. 3.2.7: The implication of a flocking algorithm and their responses to cellular automata rules; source: Anzalone and Clarke (2004).

for fabrication tools, such as CNC[1] machine with three-axis mill (Anzalone and Clarke 2003, 2004; Clarke and Anzalone 2004).

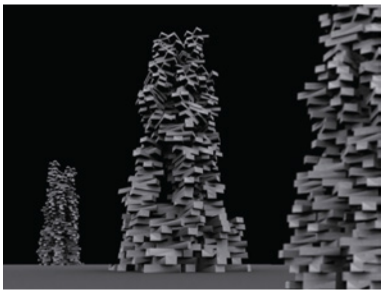


Fig. 3.2.8: The simulation of the adaptive growth based on the robotic constraints; source: Narahara (2008, 2013a,b).

Ultimately, generative algorithms require adapting the process of growing forms with the logic of materializing through fabrication processes. Considering material systems as active factors coordinates the fabrication process towards the geometric specificities of materials. Therefore, the development of generative algorithms is accompanied with the geometric definition of materials and the relationships among them. Narahara (2008, 2013a,b) developed a computational design tool based on simple generative rules to grow an adaptive cluster of unit blocks. In addition, Narahara (2008, 2013a,b) utilized the computational tool with a Genetic Algorithm (GA) for generating new rules and a physic-based algorithm for checking the stacking capacity of the blocks (Figure 3.2.8). Moreover, considering the robotic fabrication processes furthers the hard coding of the robotic tool path with an adaptive generating process through which this tool generates different possible positions for clustering blocks with respect to the robotic constraints (Narahara 2008, 2013a,b). However, both of these examples indirectly integrated the fabrication processes within agents' behaviors. Therefore, integrating fabrication and material aspects within design processes necessitates further studies to gain insight as to whether agents-based system can be adapted to the materialization process in architectural design computation.

---

[1] Computer Numerical Control.

4 Introduction of Experiments and Case Studies

4.1 Reflection of the Adaptive Behavioral Significances on Design Processes

4.1.1 Introduction

In the context of the behavior-based system, this experiment focuses on the relationship between Constrained Generating Procedures (CGPs) and Complex Adaptive Systems (CAS) to trace the hidden lever-points for constraining the simple generative model. The generative model consists of individual units with singular competencies, which include their interaction rules. The behavioral adaptations are allied with the self-organizing properties to maintain the critical states of the system from bottom-up approaches. The bottom-up mechanism proceeds simple tasks that are implemented through the basic rules within agents. For example, agents comply to maintain the internal pressure within a specific range. The overall behaviors of agents automatically cover the whole manifold without any global knowledge about the size of the environment. Adaptations to the tasks arise out of lower-level competencies, wherein the individual agents update the higher-level system with their states of satisfaction. Accordingly, it represents that the different states of agents, which are related to the agents' behavioral definitions, rely on other agents and the contextual environment.

The objective of this experiment is to investigate the behavioral aspects of emergence phenomena, where the system is self-organized at critical states with new arrangements of form and force. Accordingly, critical factors of the system, which originate within the agents' behaviors, such as the displacement vectors of motion behaviors, are analyzed to provide the relation between different parameters with the overall emergent behaviors. The obtained factors establish "IF/THEN" mechanisms within Constrained Generating Procedures (CGPs) to coordinate the system towards a state of self-organization. In addition to the boids algorithm (Reynolds 1987), this experiment considers two other algorithms, which were developed by Walter (1998) and Coates (2004, 2010). The mosaic algorithm emphasizes the parameterization aspects of skin pattern formation where each parameter affects the behavioral properties of the patterns (Walter 1998; Walter et al. 1998). In addition, the study of Coates' (2004; 2010) algorithm provides the translation of simple behavioral procedures within algorithms to the vector-based algorithms. The turtle algorithms are embedded in the NetLogo[1] application; these algorithms represent simple behavioral procedures. Accordingly, these algorithms structure a generative agent-based framework to reflect the adaptive behaviors in design processes. Implementing this generative tool provides insight into the significance of each individual parameter at micro (low)-levels and how these parameters change the order at macro (high)-levels.

4.1.2 Context

According to Holland (1995), the recognition of lever-points within Complex Adaptive Systems (CAS) are necessary to model a complex adaptive system. At these critical points, the behavior of the system drastically transits from one phase to another (change phases). The transition phases within system behaviors result from interactions among lower-level elements, where the system consists of several critical states. Exploring critical states enables modelers to gain a better understanding of system behaviors. The critical moment describes system adaptability; changing parameters in the micro-level alters the state of the system at the macro-level. For example, micro-level interactions cause different emergent phenomena at the macro-level. Eventually, in this experiment, different behaviors emerge from self-organizing at both the micro and macro levels.

Critical states of system behaviors provide essential knowledge about developing "IF/THEN" mechanisms to enhance Constrained Generating Procedures (CGPs). These constraining mechanisms enrich system behaviors by categorizing the effectiveness of each parameter. This categorization determines the singular level of adaptability by identifying a certain range of parameters. Conditional states, which are developed within these "IF/THEN" mechanisms, facilitate switching between different emergent patterns. Exploring the emergence of the system requires

---

[1] see Wilensky (1999a).

a modulation of the level of forces within different states. The self-organization within micro-macro levels adjust unstable levels to return to stable states. This intervention requires mechanisms to provide adaptation within the system. These mechanisms include different procedures to evaluate the generative system through internal conditional criteria.

4.1.3 State-Of-Art

Walter (1998) investigated the Clonal Mosaic discretization on mammalian skins, such as a giraffe to develop pseudo code for simulating their morphogenesis. The simple behavioral system, which was described through a Clonal Mosaic algorithm, consists of different parameters to control the behavior of each individual cell and the overall behaviors of the whole system (Walter 1998; Walter et al. 1998). Implementing Walter's (1998) algorithm for an agent-based system commences with distributing agents on a two-dimensional field through random or stochastic algorithms. And after this random distribution, agents gradually distribute themselves with equal distances over the predefined boundary or environment through behavioral mechanisms. Figure 4.1.1 illustrates these two steps: first, of randomly distributing agents, and second, of employing the behavioral mechanisms proposed by Walter (1998).

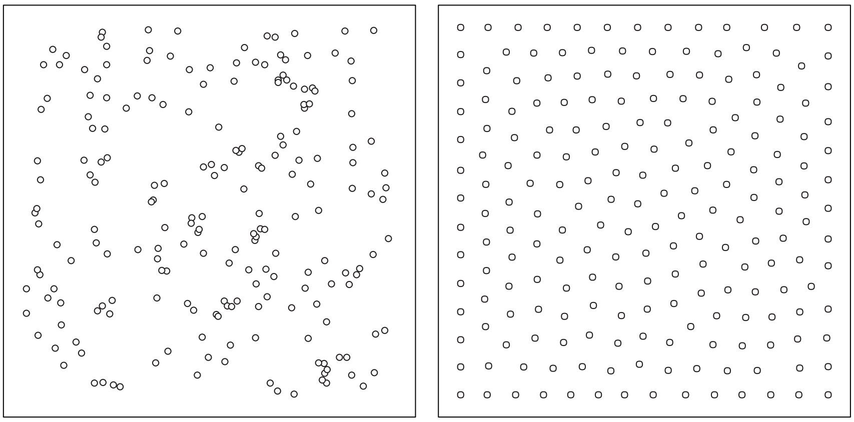


Fig. 4.1.1: Two steps: The random distribution of agents (left image) and the relaxation process of agents (right image).

In addition, Coates (2004, 2010) investigated agent-based systems by differentiating the agents' attributes to demonstrate emergent patterns, such as rings or Voronoi cells. Coates (2004, 2010) tested his algorithm with a simple behavioral definition through the NetLogo application. Conducting experiments based on Coates' algorithm involves emergent properties, which arise out of a simple negotiation between two different types of agent.

The first type of agent, as the main type *A* pushes all other agents with the type *B* off its territory. In addition, type *B* agents have internal interactions to retain their distances to the adjacent agents in a specific range. As Coates (2010) mentioned in his work, modeling these simple behaviors reflects emergent properties that can metaphorically be abstracted to geometrical representations, like circles, or rings around the main type of the agents (Figure 4.1.2).

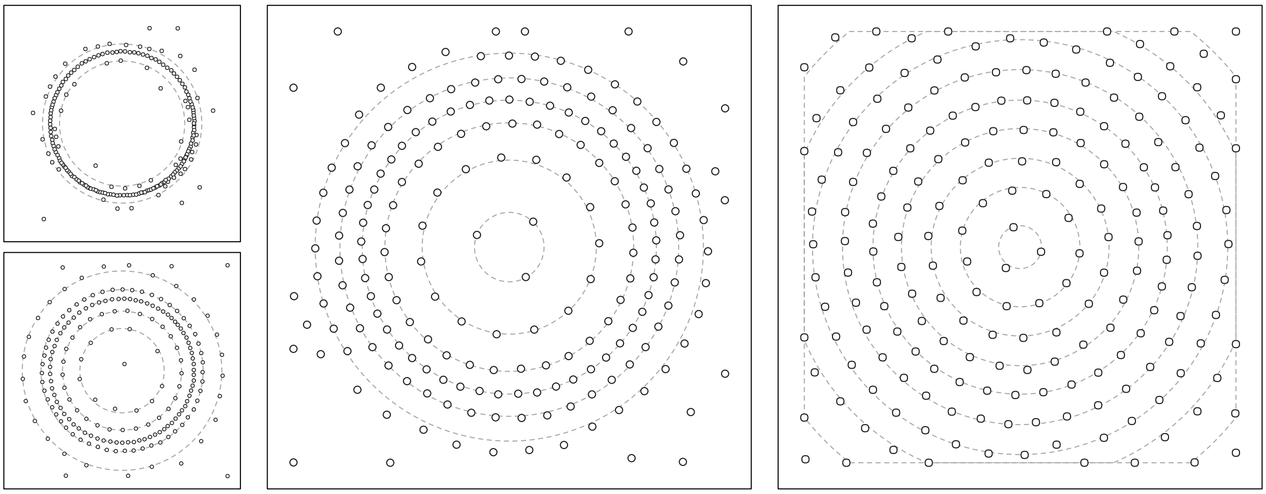


Fig. 4.1.2: Increasing the force values among type *B* agents starts to generate rings around the type *B* agent with isolated force value.

In both of these computational setups, based on Walter (1998) and Coates (2004, 2010) algorithms, increasing the number of type *A* agents starts with colonizing type *B* agents. The overall colonization is comparable to a Voronoi diagram (Figure 4.1.3). Coates (2010) compares the behavioral formation of a Voronoi diagram with computational geometric methods, which require several lines of codes to describe the mathematical relationship between each cell within the overall Voronoi diagram (Coates 2010). However, the Voronoi diagram generated with an agent-based system, which follows behavioral procedures, emphasizes the importance of behavioral procedures (computational behavior) and highlights their advantages to the mathematical techniques (computational geometry). In behavioral modeling, the generated patterns are entirely related to behavioral actions and reactions or, in other words, the negotiations between two different types of agents, which have internal interactions.

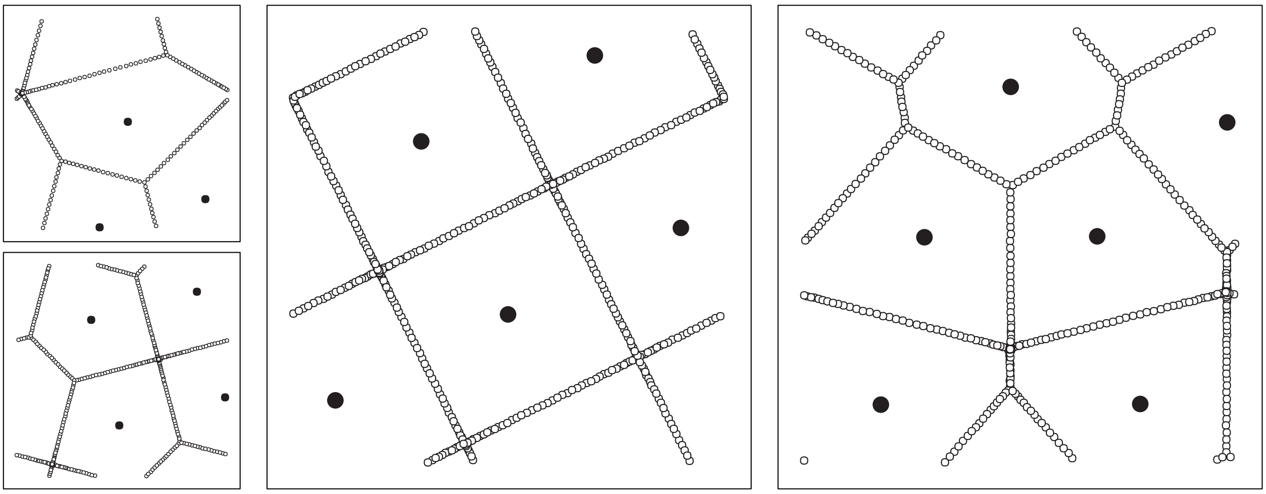


Fig. 4.1.3: Increasing the number of type *A* agents simulates cellular morphologies.

4.1.4 Methods

A computational framework is explored to simulate and model motion behaviors by means of displacement vectors developed within the boids algorithm (Reynolds 1987) and turtle graphic algorithms. Figure 4.1.4 illustrates the differences between an autonomous agent and a turtle agent in coordinating their direction towards a target. The generative agent-based algorithm emerged from both of these algorithms through a vector-based system within the Rhinoceros[1] as a CAD application and NetLogo[2].

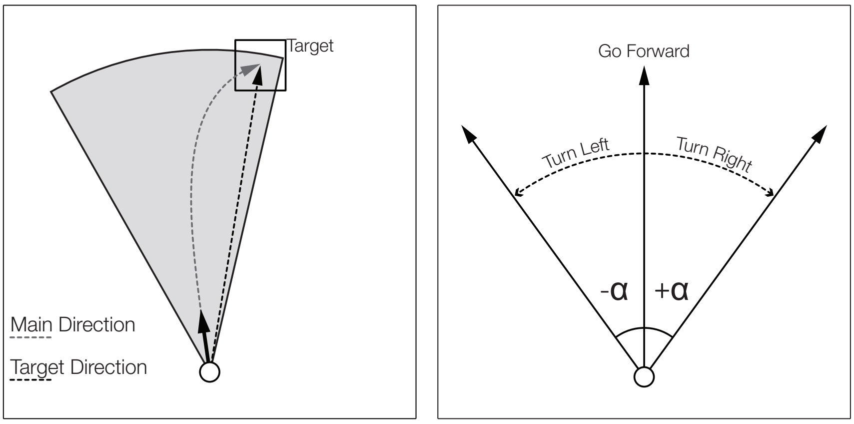


Fig. 4.1.4: Schematic diagrams of an autonomous agent (left image) and a turtle agent (right image).

The algorithmic and behavioral thinking within NetLogo contributes significantly to the development of the script through

---

[1] see Robert McNeel & Associates (2015).

[2] see Wilensky (1999a).

different programming languages, such as Iron Python[1] and Visual basic .Net[2], which are customized for this CAD application. The developed algorithm is explored by embedding parameterization mechanisms to change the quantitative parameters sequentially and to check the effect of this modulation on the quality of the system.

4.1.5 Development

Determining agent properties

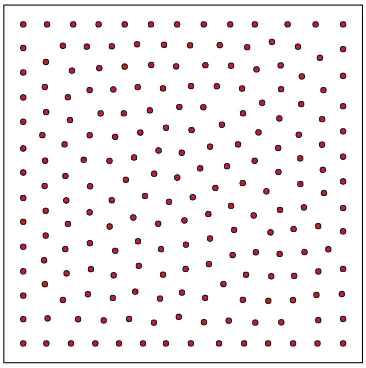


Fig. 4.1.5: Relaxation behaviors of agents.

Types of agents. A generative system was developed to investigate the micro-level behaviors in two steps. Since, this generative system investigates behavioral adaptation in a complex system, the type of agents used was limited to provide better insights into growing complexities through a simple arrangement. In the first step (Figure 4.1.5), the generative system was developed with one type of agent (type A). In this system, agents maintain their distance from adjacent agents. The agents are dynamically related to each other, and at each iteration they check and adjust their distances in response to adjacent agents. This dynamic behavior relies on the topological connectivity of the agents at the micro-level to find the nearest agent. Figure 4.1.6 illustrates this process in which the selected agents find the closest agents in a specific range, and then apply repulsion and attraction behaviors.

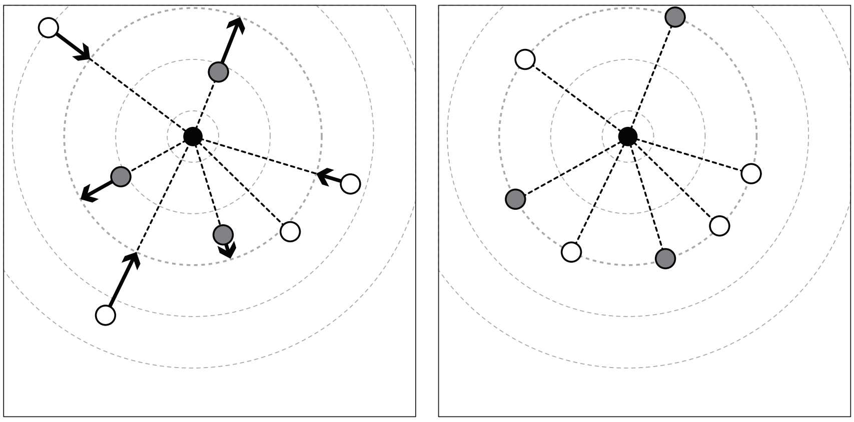


Fig. 4.1.6: Simple interaction between an agent and its adjacent agents.

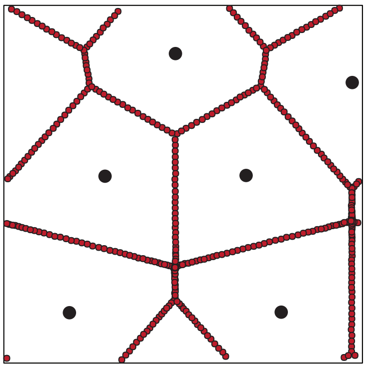


Fig. 4.1.7: Cellular behaviors of agents.

In the second step, two types of agent (types A,B) are introduced into the system from which the type A as a static agent are relatively independent to the type B as active agents (Figure

---

[1] see Dino Viehland (2014).

[2] see Microsoft (2015).

4.1.7). The independent behaviors of static agents are essential to simplify the basis of the generative system. The steady states of the static agents with the invariant locations reduce the complexity of the system at the initial stages and facilitate further study of the second type of agents. Figure 4.1.8 shows the interaction behaviors between type *A* and type *B*. These behaviors consider the attraction and repulsion behaviors of type *A* against type *B*.

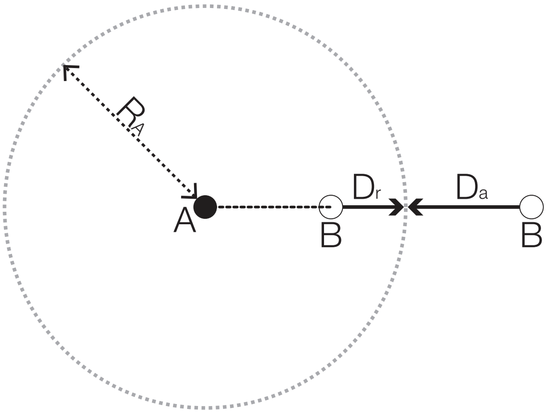


Fig. 4.1.8: Repulsion and attraction forces between types *A* and *B*.

The next experiment considers type *A* agents with dynamic behaviors. Furthermore, type *A* is independent from type *B*, while type *B* relies on the behaviors of type *A* agents. Type *A* agents actively repel and attract type *B* agents. The coherent behaviors among type *B* agents rely on internal mechanisms. Establishing internal mechanisms allows agents to repulse and attract each other simultaneously. In addition to this internal interaction, type *B* agents have to calculate the external repulsion forces imposed from type *A* agents and vice versa (Figure 4.1.9).

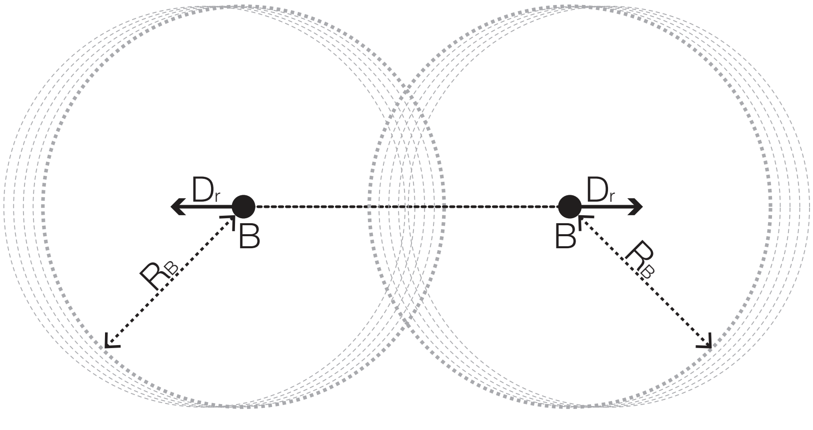


Fig. 4.1.9: Repulsion forces among type *B* agents.

In general, this process is accompanied by checking the distances among agents. The repulsion and attraction behaviors rely on the relation between the circumferences of the agents. When the distance between two agents is greater than the summation of their radii, the attraction behaviors will add displacement factors to each agent (Figure 4.1.10). Simultaneously, employing attraction and repulsion behaviors provides tangencies among agents.

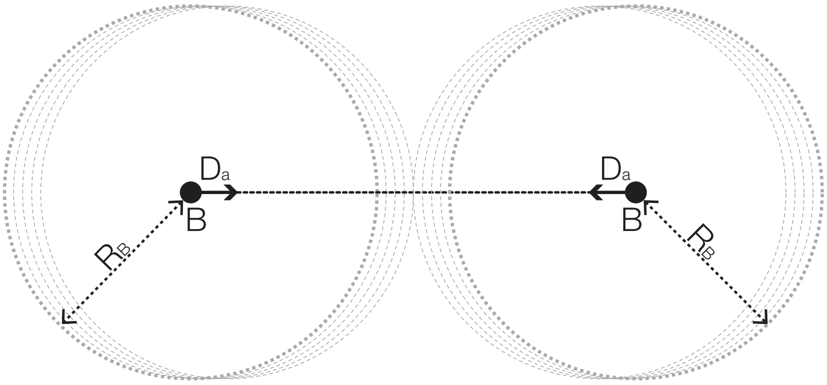


Fig. 4.1.10: Attraction forces among type *B* agents.

Geometric properties. Understanding behavioral adaptations within a complex system necessitates an emphasis on the interactions between one type of agent at first, and then later, between two types of agents. Accordingly, an agent's morphology is reduced to simple geometrical shapes. In this case, the geometrical properties of the agents are determined through a simple point without dimension. In Euclidean space, a simple point is represented by a three-axis coordinate system. This morphological simplification contributes to the study of behaviors that arise from simple interactions among agents. Each point agent has an attribute to maintain its own area that eventually adds a boundary or circumference to the agents.

Behavioral properties. The interactions among agents rely on rule-based techniques. The agents with basic morphologies perceive their surrounding environment. Then, the evaluation of observed data releases appropriate responses through a set of "IF/THEN" mechanisms. Considering agents' typology specifies the basic rules within this experiment. The basic rules are the general behaviors of attraction and repulsion among agents. The generative agent-based system links agents to other agents and environments through three levels of sensing (perceiving), processing, and responding (Figure 4.1.11). Accordingly, behavioral properties are the outcome of agent responses to external and internal attributes and factors, for example, the factors that control the distances among agents and their relations to the relevant environment.

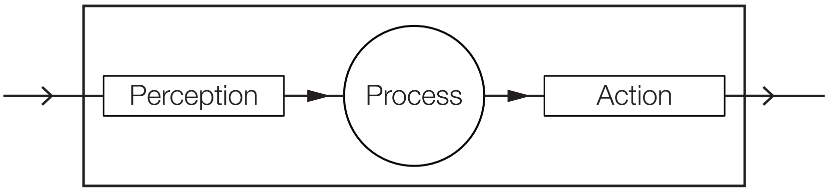


Fig. 4.1.11: A schematic diagram of a behavioral layer.

Determining a contextual environment or field

Modeling a generative agent-based system requires an investigation in the development of a relevant environment, which considers environmental aspects within the agents' interactions that trigger different levels of actions. Accordingly, the computational framework includes environmental aspects. The geometric features of the environment delineate the essential parameters that affect agents' behaviors. A basic environmental setup for this computational framework is a confined area that is projected on a horizontal plane, which provides a simple two-dimensional field of action for agents. Therefore, the environment is fundamentally distinguished into middle and edge areas. The circumference of the environment imposes external forces on the agents' distribution. The circumference is considered an active element within the modeling system. Boundary conditions in mathematical modeling, such as periodic or reflective boundaries, improves the circumference's ability to act as an agent, see Figure 4.1.12.

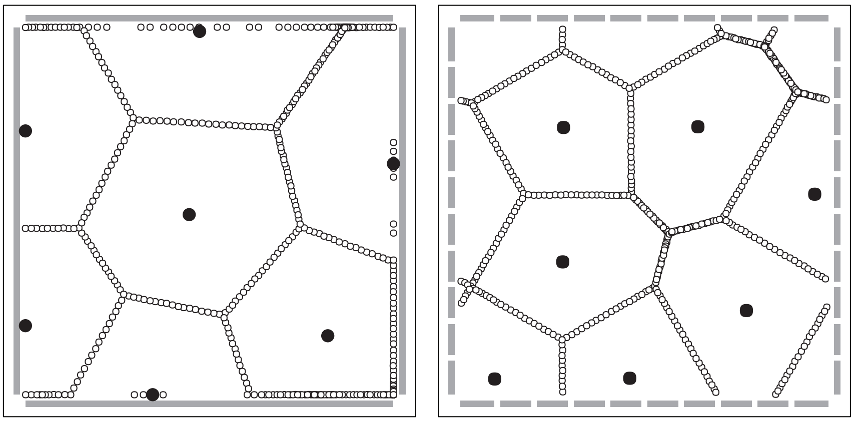


Fig. 4.1.12: The effect of environment circumferences on macro-regularities; left image: The environment with reflecting boundary; right image: The environment with periodic boundary.

The circumference is distinguished by several behaviors, such as repulsion and attraction behaviors. These behaviors use the same logics that are applied for developing a behavioral interplay between the aforementioned agents (Figure 4.1.13). These behavioral characteristics define the environment as an active agency within the process of modeling.

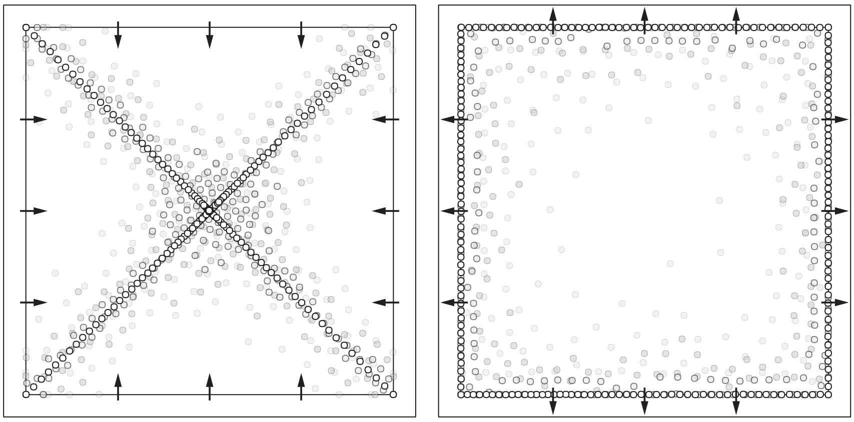


Fig. 4.1.13: The attraction and repulsion behaviors of circumferences; left image: Repulsion behaviors; right image: Attraction behaviors.

Determining interaction behaviors or rules

The rules of interaction between agents are abstracted from three different modeling techniques that are categorized as parametric based modeling (Clonal Mosaic Model), vector-based modeling (Boids algorithm), and a turtle graphics algorithm (NetLogo application), which are all implemented with a vector-based approach.

According to Walter (1998), the Clonal Mosaic Model (CM) is a parametric based modeling technique that simulates cell division and cell-cell interactions. This modeling techniques synthesizes "mammalian coat patterns" (Walter 1998; Walter et al. 1998). In Walter's (1998) algorithm, the initialization procedure of generated patterns lies on a relaxation mechanism that is implemented by two main behaviors: adhesion and repulsion. Each cell maintains its own area by repulsing other neighboring cells; furthermore, each cell uses adhesive forces to preserve the integrity of cellular structures (Walter 1998; Walter et al. 1998).

Coates (2010) developed an algorithm to represent the emergent behaviors. Coates's (2010) algorithm utilizes simple behavioral rules within the NetLogo application. Similar to the Clonal Mosaic algorithm, this algorithm is designed to attract and repel agents from each other. The NetLogo application simplifies the implementation of these behaviors through a set of commands that ask agents to do specific tasks. For example, one command includes a request from an agent (turtle) to move toward the nearest neighbor but, also, to preserve its distance from the chosen neighbor (Coates 2010).

The flocking algorithm, developed by Reynolds (1987), is a collective behavioral modeling technique that simulates the locomotion behaviors of a flock of birds. This model utilizes three main behaviors: “collision avoidance,” “velocity matching,” and “flock centering” (Reynolds 1987). Further development of the flocking algorithm has evolved to support different behaviors, such as “flee,” “seek,” and “path following” (Reynolds 1999). However, in the context of this experiment, three main behaviors of “cohesion,” “separation,” and “alignment” (Reynolds 1999) are investigated to cooperate with vector-based methods to give internal cohesions to the closest agents and maintain separation between them.

All three algorithms are investigated to generalize the essential behaviors that self-organize the complexity of interactions among embedded elements within a system. The strength of each algorithm is evaluated to develop a catalog of behaviors. The catalog consists of general considerations of behaviors and specific details of those behaviors, which, in this experiment, are adhesion and repulsion. The interconnections among agents are established through different types of connectivity. Agents explore their neighbors with topological or topographical connectivity (CECA 2004). Topological connectivity links agents regardless of the distance of their separation, in contrast to topographical connectivity, which is the agents’ connectivity within Euclidean space (Figure 4.1.14). Therefore, the closest agents in Euclidean space might differ from the nearest agents in Topological space.

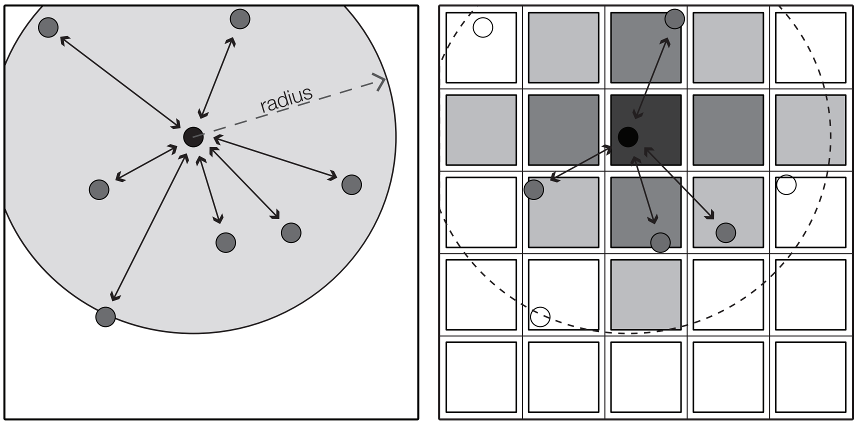


Fig. 4.1.14: Topographical and topological connectivity; left image: Euclidean space; right image: Topological space.

In Euclidean space, the connectivity among agents is effective in generating different collective behaviors. For example, Figure 4.1.15 illustrates that utilizing the generative agent-based tool on random distributed agents on a two-dimensional manifold exhibits two different types of emergence. Considering the connectivity between an agent and its closest neighbor leads the system to equally distribute agents throughout the environment (Figure 4.1.15 left). On the other hand, changing the number of interconnections between agents and their closest neighbor drastically change their interaction behaviors. The collection of their interactions cluster agents around a central core (Figure 4.1.15 right).

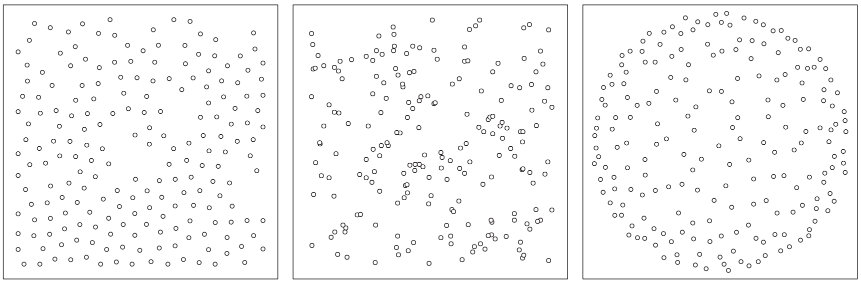


Fig. 4.1.15: Connectivity effect on emergence behaviors; left image: Interaction between agents and their closest neighbor; middle image: Random distribution of agents; right image: Interaction between agents and all other agents.

Interrelating agents, under Euclidean space, employ simple controlling mechanisms to sort the closest agents into an array list. The sorted list demonstrates the access of main agents to adjacent agents via their indices or their distances (Figure 4.1.16). These two ways provide different levels of access to the agents at the micro-levels, which directly affects the emergent behaviors at the macro-levels.

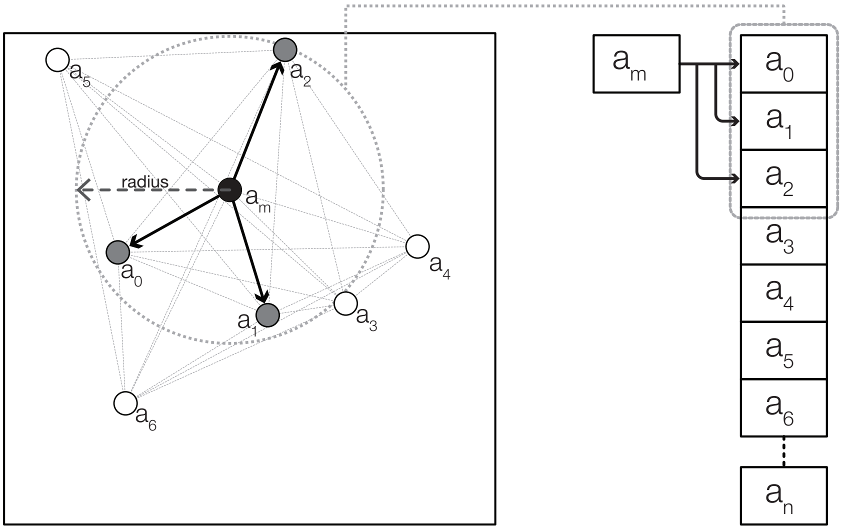


Fig. 4.1.16: The sorting mechanism that finds the closest agents within the specific radius.

Determining the inhibitory mechanism

The inhibitory mechanism, which is established within the first phase of the experiment, self-organizes the micro-macro effects in the generative agent-based system. This process is followed by measuring the agent's displacement vector, and calculating the summation of all agents' displacements[1]. The relations between these two values define a specific domain in which deviation from the domain destabilizes agents and, accordingly, the whole system. In other words, the inhibitory mechanisms measure the distances to the perceived closest agents and if the distance has a large deviation from the accepted tolerance then the inhibitory mechanism triggers the related behaviors. In this experiment, the "IF/THEN" mechanism examines these two aforementioned aspects to analyze any deviations from the mean average displacement factors. The inhibitory mechanism self-organizes the system by varying the number of agents, when displacement factors have any differences with average mean displacements (Figure 4.1.18). Applying this inhibitory mechanism enables the generative agent-based system to fill the empty environment by comparing collected individual behaviors as macro orders with the single behavior of individual agents at the micro-level (Figure 4.1.17).

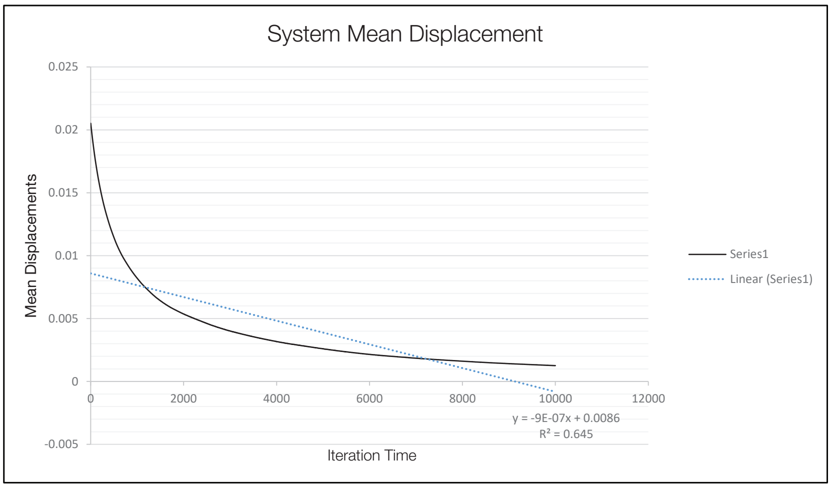


Fig. 4.1.18: The diagram of mean displacements that are analyzed over time.

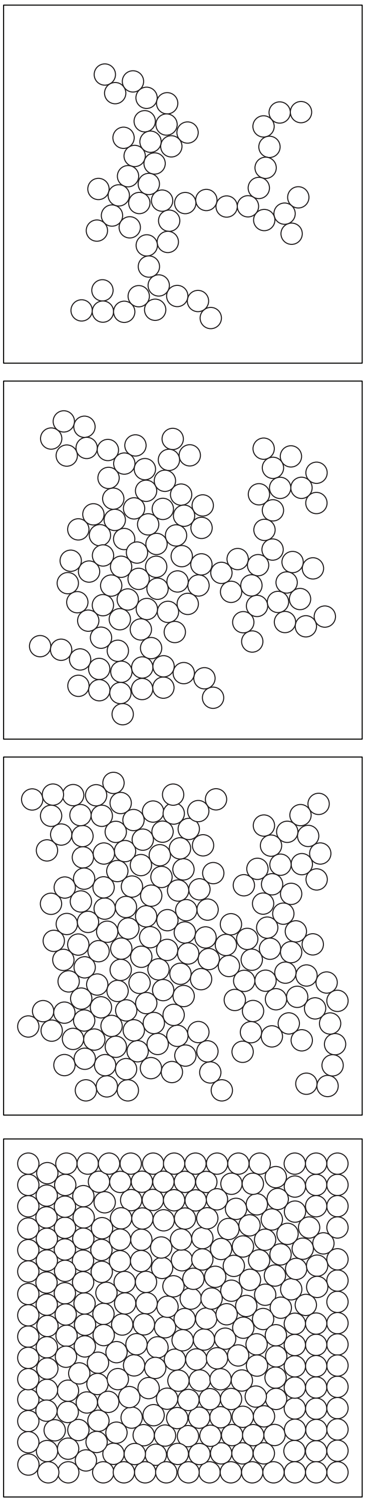


Fig. 4.1.17: The development of self-organization at the micro and the macro levels. At the micro-level, agents avoid any overlapping boundaries. At the macro-level, agents fill the manifold with the agents.

Determining the coordination mechanism

The general attributes of agents, which are developed in this experiment, are their coordinates on the manifold or environment

---

[1] Reeves (2013) develops a sphere packing mechanism that considers altering the number of spheres when the average pressure of all spheres is less or greater than the predefined threshold. The author's experiment has a different approach to investigate micro-macro effects within a Complex Adaptive System (CAS). Determining the lever-points through statistic analyzing automatically determines the tolerance at the macro-level and agents dynamically adjust their local behaviors with this tolerance.

and their displacement vectors. Each agent utilizes a mechanism of assembling behaviors to compute the position and displacement vector throughout each iteration. Similar to the steering mechanisms in the flocking algorithm, the displacement vector is a summation of behavioral responses that are translated into vector-based algorithms. After each iteration, the new position is calculated by adding the displacement vector to the current agent's coordinate position. Computing displacement vectors relies on inhibitory mechanisms to selectively activate the layers of actions. These layers provide two levels of interaction, one level provides the intrinsic behaviors, such as a repulsion or adhesion behavior, and another level determines the extrinsic behaviors that consider the agents' interaction with environmental aspects. A classifier system prioritizes agents' responses for each action. The summation of these prioritized values indicates the displacement factors for next step. Figure 4.1.19 highlights the behavioral assembly procedures.

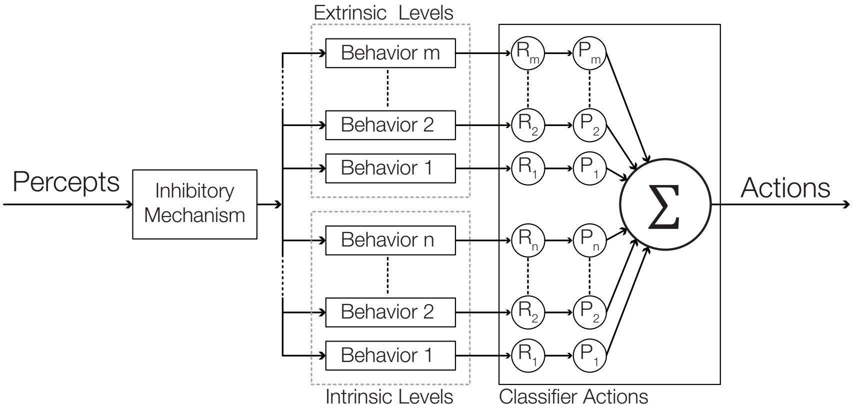


Fig. 4.1.19: Assembling behaviors, including inhibitory mechanisms, intrinsic and extrinsic behavioral layers, and classifier actions.

The classifier mechanism coordinates the summation of responses. It prioritizes the behaviors by weighting the agent's responses. In this experiment, since agents have limited behavioral mechanisms that are confined to repulsion and adhesion behaviors, the prioritization, as an overall allocator, is limited to internal variables for parameterizing these behaviors (Figure 4.1.20).

Determining the parameterizing mechanism

Development of this generative agent-based tool is monitored by parameterizing the agents' attributes. Monitoring agents' behaviors to fix one variable while the other variable changes frequently. This monitoring mechanism is customized to change the variable' value automatically or by the user's interventions. Accordingly, the adhesion variable is parameterized to alter the

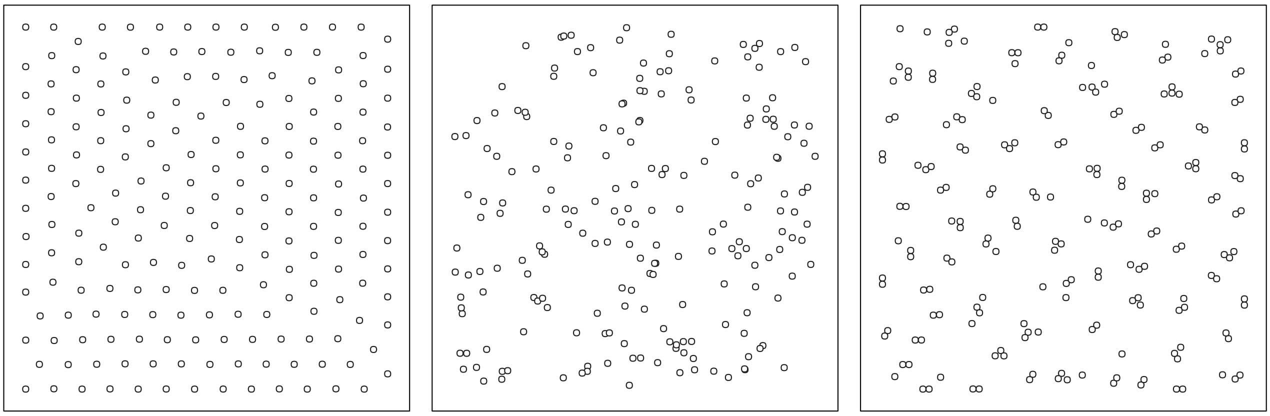


Fig. 4.1.20: Parameterizing repulsion and adhesion behaviors; left image: Increasing repulsion behaviors; middle image: Random distribution of agents; right image: Increasing adhesion behaviors.

adhesive behaviors among agents. The correlation between different variables changes the overall behavior of the system, which is the consequence of interactions between internal variables and external forces. Equipping the computational framework with monitoring mechanisms enables the framework to investigate the behavioral transition phases from one level of self-organization to other levels. Understanding the location of the lever points within the complex system requires the continuous monitoring of one variable by constantly increasing its value and then observing the overall system's behaviors. For example, the adhesion variable of agent type *A* is parameterized within the monitoring mechanisms through which the value of this variable is automatically incremented with the tolerance value defined within the system operator. In parallel, the other agent's attributes, such as iteration times, velocities, and locations, are stored to analyze the agent's behavior in both the micro and the macro levels.

4.1.6 Discussion

Understanding the behaviors of the system requires an investigation of both the micro and macro levels. The bottom-up approach of this investigation introduces the significant effects of micro-level interactions onto the overall behavior of the system. For example, the inhibitory mechanism sequentially adds agents to the system. This process allows designers to study interactions at the micro-level and to observe pattern formation, which is determined by the sum of agents' micro-behaviors. In this specific set up, the emergence of pattern covers a range of different patterns: from branching patterns to square and hexagonal packing. These patterns are generated from overall agents that interact with environments. The negotiation between agents and the environment informs the lower-level agents to change their position, which subsequently changes the connectivity network between other agents. This means that the individual agents not only sense the whole system behaviors, but also change the behavior of the whole system. When a new agent is added to this dynamic system, the agents' behaviors expand radially through the embedded agents. The wave-like pattern that represents the transfer of information among agents through direct interactions also explores the behavioral-based approaches within the system.

4.2 Agent-Based Digital Morphogenesis for Plate Shell Systems

4.2.1 Introduction

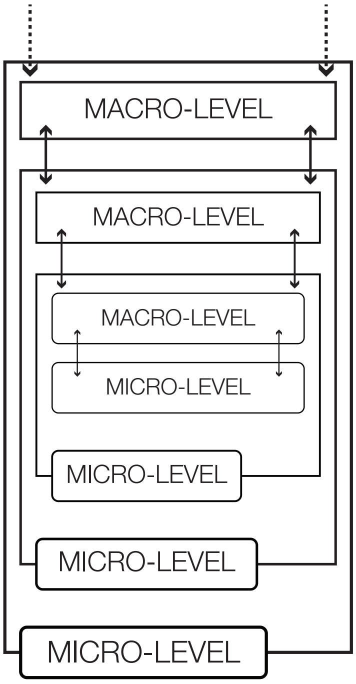


Fig. 4.2.1: A schematic diagram of intertwined micro-macro levels.

Generative agent-based systems enables a better understanding of micro-level effectiveness when determining the purpose of the model. The inter-relations among parameters require an investigation of their effects on interactions at different levels. Extending the concept of micro-macro effects to the different levels of interaction enhances the generative agent-based system with micro-generating mechanisms. Micro-generating mechanisms are responsible for producing macro-stable states. In this case, macro-effects at one level become micro-generating modules at higher-levels (Figure 4.2.1). Accordingly, each level is the aggregation of various micro-generating systems, which gradually determine the purpose of the model.

This experiment focuses on self-organizing cellular structures to produce complex surfaces. It investigates the bottom-up principles of shell structures in which the geometric features of materials are abstracted into the agents' morphology. The agent's morphology benefits from the aggregation technique that is described as an essential property of a Complex Adaptive System (CAS) (Holland 1995, pp. 10-12). Accordingly, the morphological structure of the agent considers the aggregation of different constructor agents. This aggregation follows the morphological consistency of agents that are found in interactions both with other agents and environmental factors. In relation to their interactions, agents are required to adapt themselves to changing situations, so they can perform the embedded tasks of assembly. The aggregation of agents generates a macro-level structure with self-organizing microelements. In addition, adaptation with the environment correlates macro-level regularities with different levels of emergent complexities. Moreover, regulating the collective behaviors of agents relies on the internal mechanisms within agents and the external influences of the environment.

The preliminary findings of this experiment were published in Baharlou and Menges (2015).

4.2.2 Context

Generally, both bottom-up and top-down approaches are widely applied to construct shell structures. The structural properties of shell structures are investigated to rationalize a predefined geometry. In this context, *differential geometrical algorithms*, such as *triangular meshes*, *planar quad-lateral meshes (PQ meshes)* are developed to discretize free-form surfaces (Pottmann et al. 2007, p. 671). The mathematical approaches of discretizing free-form surfaces follow integrative methods, such as “fabrication aware-design” to consider fabrication criteria within the rationalization processes (Pottmann 2013). The main challenge in developing shell structures will arise when designers require an integration of performative aspects and fabrication principles within the process of rationalizations.

From bottom-up approaches, the free-form surfaces arise from combinatorial interrelations among material structures and generated forms. Accordingly, the self-organization of materials is investigated to construct shell structures using form-finding techniques. A notable example of this technique is Mannheim Multihalle by Frei Otto, Carlfried Mutschler, and Ove Arup and Partners (1975), in which the lattices and cells are modulated to integrate material behaviors with structural behaviors. The result of this modulation is a free-form structure, which demonstrates the importance of utilizing the intrinsic properties of a material. Encapsulating these intrinsic properties into micro specifications offers a potential to generate a shell structure at the macro-level. The micro-macro linkages provide insight into the bottom-up organization of material elements and their effects on generating free-form surfaces.

4.2.3 State-Of-Art

An early example of a computational tool for generating structural geometry is the “eifForm” application, which investigates form generation from a structural perspective; a triangulated structure is studied through structural elements and their connectivity (Shea 2004, p. 93). A recent project in the context of behavioral organization is “Lamella Flock,” which investigates the generation of self-organized tectonic structures (Tamke et al. 2010a,b). In this project, the structural properties of lamella systems are abstracted into agent systems. The agents in this system are in constant interactions with each other. Their interactions are utilized through a sematectonic communication to self-organize agents’ connectivity within a three-dimensional space (Tamke et al. 2010a). Then, the abstracted reciprocal structures are connected to generate a free-form surface. The self-organization behaviors of this free-form structure are constrained to the fixed topological mapping. The hybrid relation between the bottom-up connectivity of cells are coordinated from the top-down approach of fixed topological connectivity. The concept of the "lamella flock" is further investigated through the release of agents' interrelations from a predefined topological connectivity. Agents with a dynamic network connectivity are initiated sequentially within the three-dimensional environment (Parascho et al. 2013). In this example, communication between agents, as triangular structures, enables the development of a triangulated free-form structure, which regulates the micro-level interplay with macro-level coordination.

4.2.4 Methods

In this experiment, the development of generative agent-based computational tools relies on VB.Net and IronPython. Both of these programming languages are accessible within the Rhinoceros CAD application. The material properties of shell elements are abstracted into the geometric elements of a polygon, such as nodes and inter-nodes. The bottom-up organization of a shell structure requires an aggregating technique to categorize the common geometric properties as various classes of elements and behaviors. For each class, the development of specific rules and procedures is essential to link micro-specifications to macrostructures. The aggregation process furthers generative agent-based systems at three different levels: the level of "agent," the level of "meta-agent," and the level of "meta-meta-agent" (Holland 1995, pp. 6-15). Transitioning from one level to another requires the development of "IF/THEN" mechanisms and tagging mechanisms. Accordingly, the simultaneous utilization of these two methods within an inhibitory mechanism fosters the development of generative agent-based systems.

4.2.5 Development

Determining agents attributes

Types of agents. This experiment consists of three types of entities that are determined by their level of aggregation. Aggregation techniques commence by developing a group of entities with a set of common behaviors. The ubiquitous properties between the entities provide possibilities to form another level of aggregation. For example, Holland's (1995, p. 15) classification proposes three levels of aggregations: "agents," "meta-agents," and "meta-meta-agents" (Figure 4.2.2).

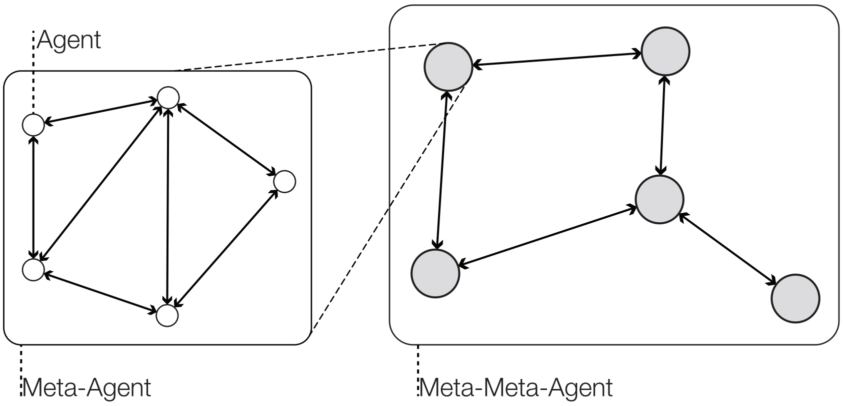


Fig. 4.2.2: A simple representation of interrelations among agents, meta-agents, and meta-meta-agents.

Furthermore, this study provides both micro and macro levels to facilitate aggregation techniques for self-organizing agent behavior. At each level of aggregation, the interplay between entities is located at the micro-level, where the new level of aggregation (meta-agent) arises from interactions among lower-level entities (agent). In relation to the micro-level structure, the new aggregation level generates macro-level properties (Figure 4.2.3). The transition from one level to another level, from micro to macro, requires different class of behaviors to coordinate the linkage between these two levels. In the behavior-based context, entities with similar tasks aggregate with each other to form a new level of entities with common purposes. Accordingly, the new aggregation types are manifested from interactions between entities with the same type of behaviors.

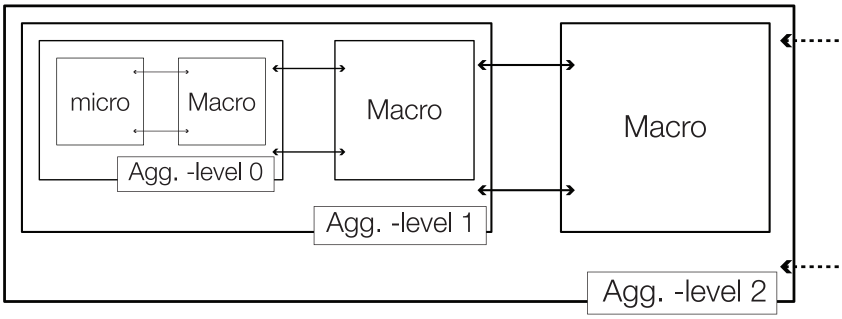


Fig. 4.2.3: A conceptual diagram comparing micro-macro levels with different level of aggregation.

These three types of agents are linked together through micro-generating systems that organize dynamic structures with different versatilities. Accordingly, the higher-level of aggregation is generated by lower-level synergies. Interaction between lower-level elements emerges in higher-level entities (Figure 4.2.4). The agent and the meta-agent are theoretically interrelated; however, meta-agents are the emergent properties of agents' interactions, at least conceptually. The meta-agents' morphologies are the result of agent aggregation. Meta-agents also inherit some characteristic features from the agents. This inheritance allows meta-agents to benefit from the agents' properties and behaviors; moreover, meta-agents utilize a different class of behaviors to accomplish their own tasks. Assigning tasks to different levels of aggregation necessitate the definition of diverse rules and behaviors for agents and meta-agents. These rules must be in proportion to the morphologies of agents and meta-agents; otherwise, the entities are incapable of performing their tasks.

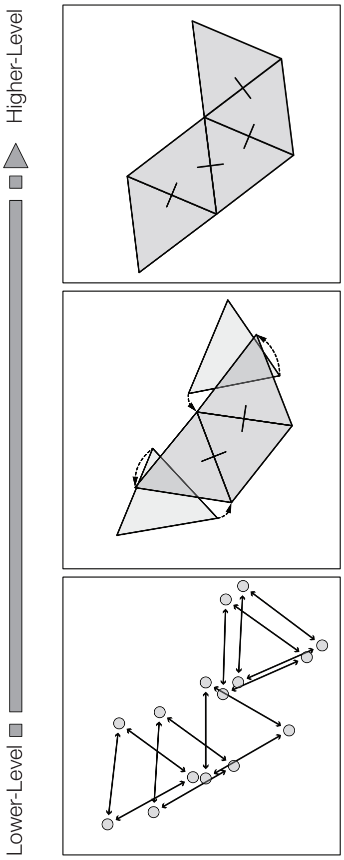


Fig. 4.2.4: Conducting an aggregation technique via the transition of lower-level elements to higher-level orders.

Geometric properties. The aggregation technique provides a method for constructing the geometric elements of polygonal shapes. Abstracting polygonal structures requires a study of the geometrical capacities of each structural element. Polygonal structures consist of n-gons that are defined by a set of nodes and inter-nodes (Figure 4.2.5). An aggregation of these nodes makes one polygon, in accordance to the agents and meta-agents, each of these nodes is abstracted to a subset of agent as a valence.

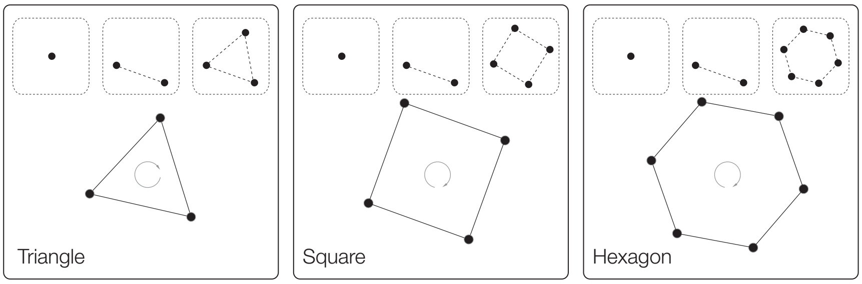


Fig. 4.2.5: Three types of n-gons with different configurations of nodes and inter-nodes.

Generating a polygon is necessary to develop appropriate rules for collecting sub-agents, which are used to generate an agent (i.e., a polygon). One essential attribute of this set of rules is the number of valences at one level of aggregation in which the number of valences defines the geometrical capacities of agents (Figure 4.2.6). For example, a triangle has three valences, a square has four valences, and a hexagon has six valences. Sub-agents with similar behaviors are collected within one type of agent. Each valence requires a mechanism to count the number of agents, which limits agents' connectivity on the higher-level.

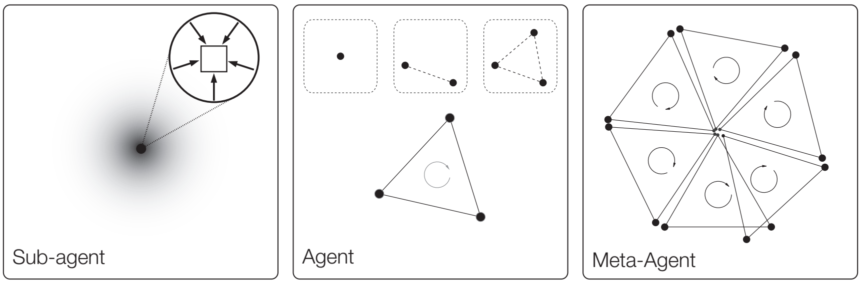


Fig. 4.2.6: An aggregation process of a hexagon, which includes sub-agents, agents, and meta-agents.

The number of sub-agents, which are collected within agents, is a minor parameter without any effect on defining the behavioral rules. However, this secondary parameter in the micro-level changes the agents' morphology. A minor parameter at micro-levels is converted to a major parameter at macro-levels. Figure 4.2.10 illustrates the importance of the number of sub-agents that impose the valences connectivity numbers. Considering the mathematical relationship between these two factors enables aggregating agents to assemble singular types of meta-agents.

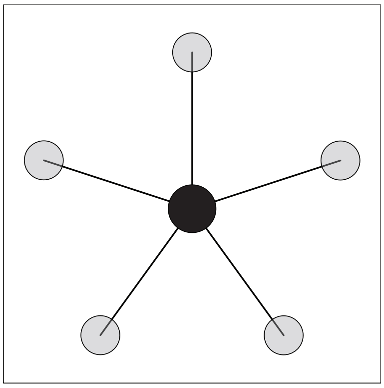


Fig. 4.2.7: A schematic Diagram of a valence with five connections.

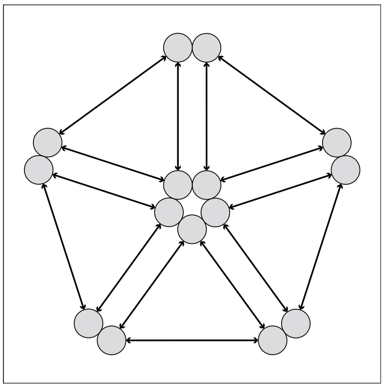


Fig. 4.2.8: Agent-Agent interactions at shared valences and available edges.

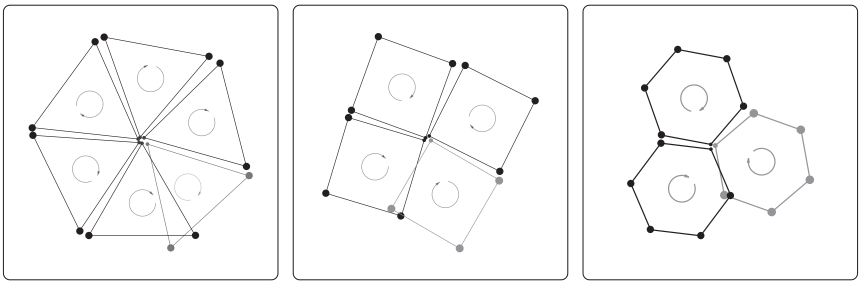


Fig. 4.2.10: Different assemblies of agents and meta- agents.

At another level of aggregation, when the agents are collected to assemble meta-agents, the inherited properties of sub-agents determine the degree of agent connectivity at one node, see Figure 4.2.7. This process follows the agent properties, where the naked edges of agents modulate the assemblage of meta-agents. Accordingly, the morphology of a meta-agent is not only related to the sub-agents' capacities, it is also related to the availability of the agent's edges (Figure 4.2.8). The correlations between geometrical capacities at the macro-level are interrelated to secondary parameters at the micro-levels. The secondary parameters that have minor effects on their aggregation level lead to the

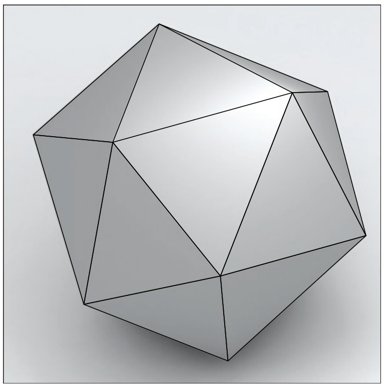


Fig. 4.2.9: A morphology of one meta-agent.

fundamental behavioral changes at the higher-level of aggregation (Figure 4.2.9). Figure 4.2.11 illustrates two setups of sub-agents, agents, and meta-agents, in which the orders of self-organization demonstrate the emergence of two different complex surfaces. The regulated degree of connectivity provides insight into the effect of valences' connectivity and edges availabilities.

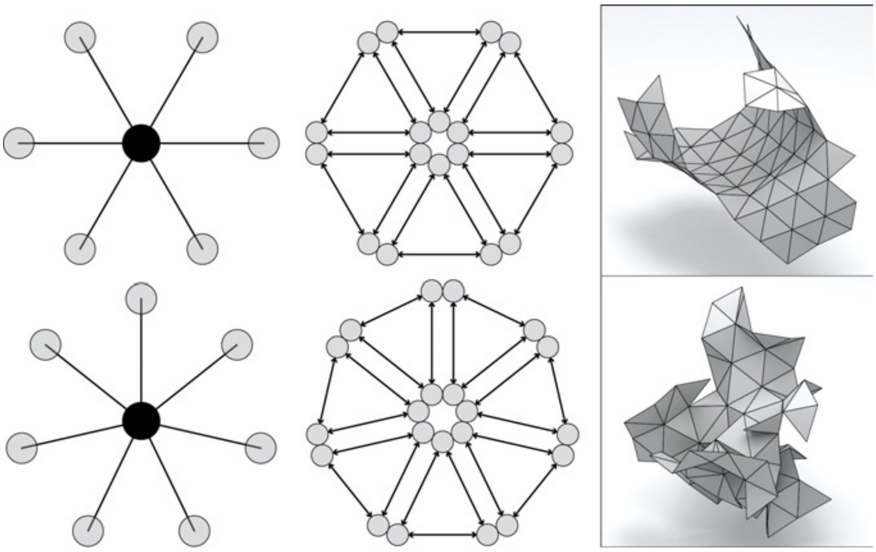


Fig. 4.2.11: The comparison between two different capacities of valences.

This process also considers the irregular capacity of valences through which it increases the level of complexity. Figure 4.2.12 shows this complexity by varying the capacity of valences between six and seven.

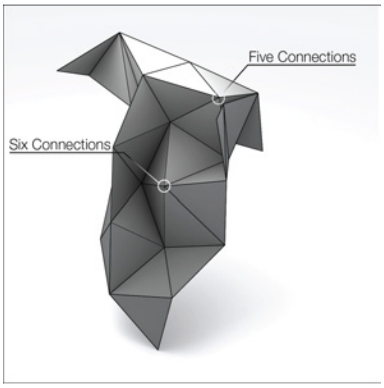


Fig. 4.2.12: Combination of two different connections in generating one meta-agent.

Behavioral properties. The behavioral properties of agents are organized at both the micro and the macro level. Micro-level behaviors self-organize the modeling system toward macro-level behaviors, with the possibility of exhibiting emergent phenomena. This concept necessitates an organization of behavioral rules in the micro-macro categorizations. Similar to the concept of homeostasis in biology, these two levels use self-organization to maintain the consistency of internal properties, and to adapt overall organism to the external factors. Figure 4.2.13 schematically represents a behavioral layer, which considers micro-macro effects when adapting and regulating external and internal pressures. Each level considers self-organization procedures. The consistency of an agent's morphology is defined within the micro-level of the generative system, wherein stability arises from the synergy among elements. The elements with informed tasks apply several mechanisms to adjust their behaviors. This adjustment is required to assemble elements and to preserve the cohesion of their accumulation.

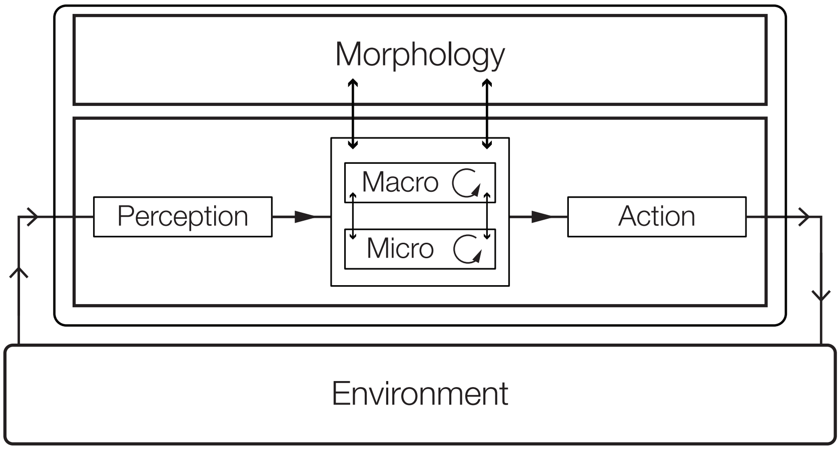


Fig. 4.2.13: Indicating micro-macro levels within a behavioral layer.

The micro-level behaviors rely on generating mechanisms that are constrained by specific rules. The implementation of these constraints requires feedback loops with "IF/THEN" mechanisms. These mechanisms impose the conditional states of rules to regulate the behaviors of each element. The micro-level behaviors are in bilateral relation with the macro-level behaviors. The macro-level behaviors, which are associated with external factors, indirectly coordinate micro-behaviors by informing them about outer features. At the macro-level, elements have the tasks of adapting systems to external influences and informing microelements of external stimuli. Accordingly, low-level elements will release appropriate responses to adjust the system to new external conditions. Hence, at the micro-level, the cohesive behaviors are elaborated further by negotiating higher-level signals.

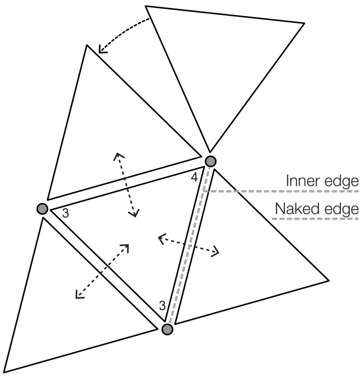


Fig. 4.2.14: Assembly procedures demonstrate the importance of edges and the valences.

The interrelation between agents and meta-agents corresponds to the basic attraction and repulsion behaviors that are associated with direct communication between agents and meta-agents. Interaction among agents, which are situated at the same level of aggregation, arranges at a new level as meta-agent. This aggregation requires topological and topographical networks to coordinate the agents' assembly. At the micro-level, agents explore the environment to find the closest set of agents, and then communicate with the closest set to provide information about the target set of agents. The agent's tag signals the potential for accepting other agents for aggregation. This potential is associated with examining the topology of the closest agent for detecting empty valences and naked edges. Figure 4.2.14 schematically illustrates the relation between an active agent and a target agent. Through assembly, two agents will change the state of the edges from naked sides to inner sides.

The "IF/THEN" mechanisms execute two possible directions to iterate the aforementioned exploration mechanisms or run the assembly mechanisms to connect with the selected agent. Defining the cyclic repetition of these two conditional rules enables the explorer agents to maintain the level of aggregation while successfully executing the "IF/THEN" mechanisms that break the cyclic repetition, and attains a new level of aggregation. The "IF/THEN" mechanisms aids in the transition from one level to another by tagging the different states to the agent systems. In this experiment, these states include *active agents*, *dynamic agents*, and *connected agents* (Figure 4.2.15).

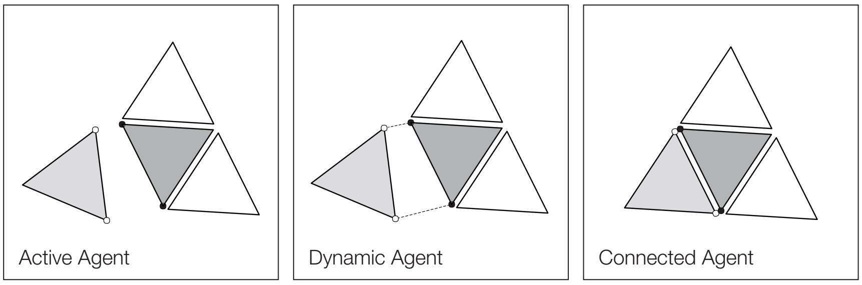


Fig. 4.2.15: A schematic representation of the assembly process.

Active agents wander across the environment to find available target agents, including unoccupied pair valences with free sides between them. When an agent finds its target, its tag changes from "active" to "dynamic." This process is accompanied by a change in the course of agents' actions toward the target agents. Approaching the targets activate the assembly process by which one mechanism changes the agents tag from "dynamic" to "connected." A successful assembly of connected agents send signals to other agents that, as target agents, are ready for further assemblies. Accordingly, each of these states steers the micro-level of agents' behaviors toward the desired macrostructures (macro-level of regularities).

Determining a contextual environment or field

The contextual environment or field in this experiment is determined by three categories: the neutral field, the passive field, and the active field. Each of these fields is elaborated with the purpose of investigating the adaptation processes within the macro-level interactions. The neutral field considers an empty field with an anchor point (Figure 4.2.16). The anchor point initiates a cluster of agents to examine the meta-agents' aggregation while the meta-agents execute different mechanisms, such as the hinging and pivoting mechanisms, to improve the possibility of connectivity by avoiding self-intersection.

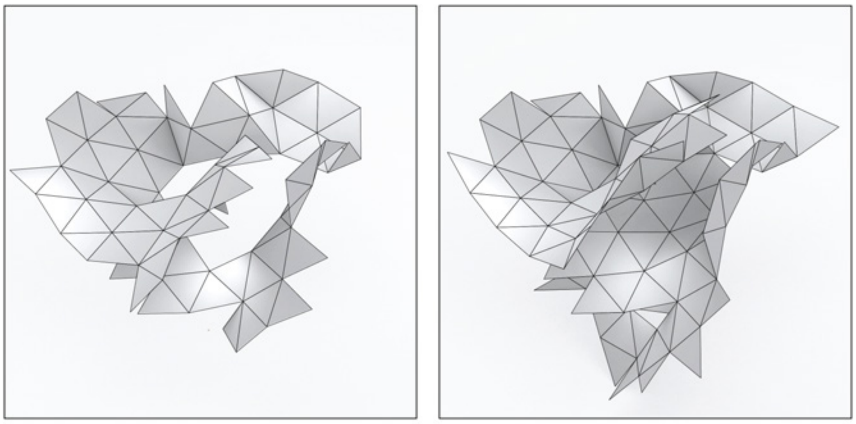


Fig. 4.2.16: Results of the self-assembly process within an environment with an anchor position.

After appraising the internal mechanism of aggregation, the active fields that consist of several attractor points are established with a three-dimensional vector field. The external forces that are imposed on the active field modulate the meta-agents' formations. The meta-agents include these forces as internal mechanisms to reflect their morphogenetic movements. Initiating meta-agents through this field steers their behaviors toward self-organizing and self-forming complex surfaces. In contrast, the passive field, which includes surfaces with different curvatures, imposes its structural complexity on the agents and meta-agents (Figure 4.2.17). The agents are forced to follow the surface curvatures; however, they self-organize on the surface without emergent properties. Due to the macro-level definition of formation, agents only self-organize their assembly process, and then adjust their behaviors to follow environmental factors.

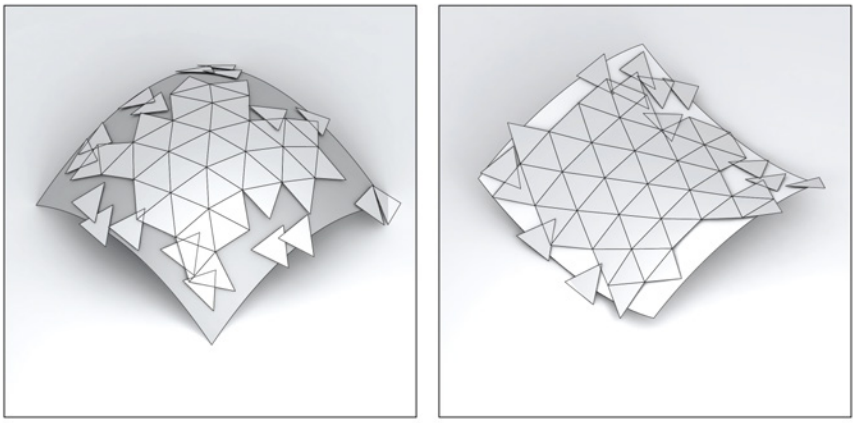


Fig. 4.2.17: Assembly processes on target surfaces (synclastic and anticlastic).

Determining interaction behaviors or rules

The behavioral mechanisms of agents and meta-agents, which are developed within the micro and macro behaviors, facilitate internal stabilizations and external adaptations. The stimulus-response mechanisms trigger behavioral mechanisms embedded within agents. Following aggregation behaviors, the stimulus includes several aspects that are directly related to the agents' assembly mechanisms and meta-agents' formations. In corresponding to the agents' assembly, the inhibitory systems direct the agents' perceptions with several conditional executions. Along with "IF/THEN" mechanisms, an inhibitory mechanism requires tagging procedures to revise and update the agents' conditions. Updating the agents' tags provide necessary signals for other agents to indicate their availability for exploring assembly processes. These two processes narrow the agents' responses down to trigger specific behaviors, which gradually prepare the agents to assemble and form meta-agents. The formation process at the level of meta-agents is accompanied by coordination systems that weigh all the responses to assign rated displacement vectors to every sub-agent, agent, and meta-agent. Figure 4.2.18 schematically illustrates the process of assembling behaviors from perceptions to actions.

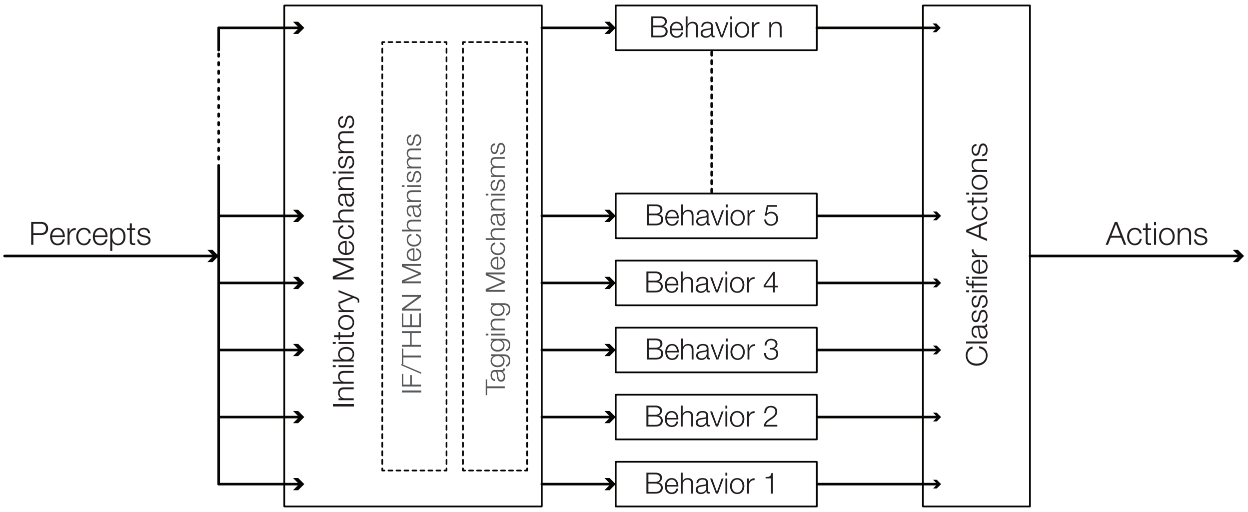


Fig. 4.2.18: Assembly behaviors utilized with inhibitory mechanisms.

Determining the inhibitory mechanism

Inhibitory mechanisms are established to facilitate self-organizations within micro-level structures. The inhibitory mechanism stabilizes the interrelations among a singular level of aggregation. Establishing a different level of micro-behaviors within agents, meta-agents, and meta-meta-agents relies on micro-constructors, such as valences and their topological connections. For example, at the level of agent, the macro-level structure of the meta-agents is related to the attributes of sub-agents and their connectivity. Accordingly, the process of inhibiting agents emphasizes the “IF/THEN” mechanism to check the internal properties, which are derived from attributes of valences, such as the availability of valences (nodes) and the possible permutations of connected agents at their adjacent edges (inter-nodes). This mechanism collects the topological descriptions of valences and edges to compute the state of the agents (Figure 4.2.19).

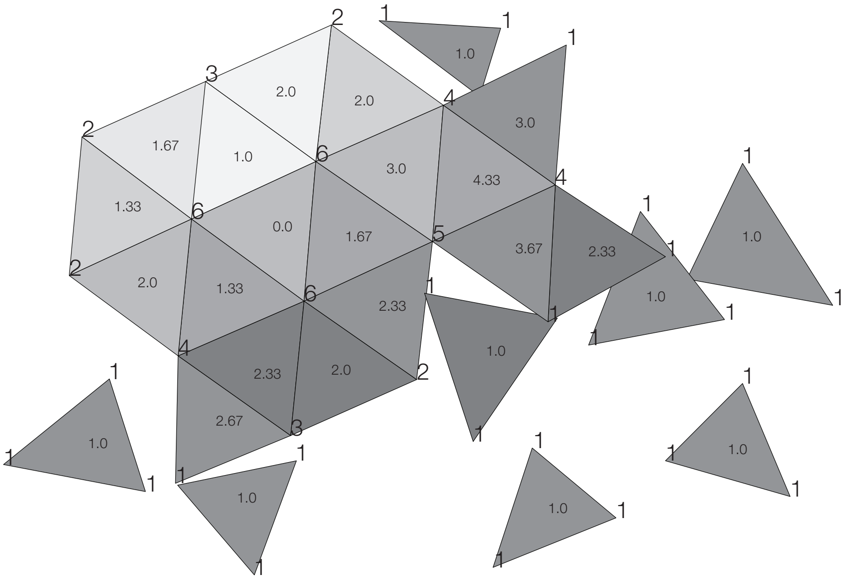


Fig. 4.2.19: Calculating permutations and probabilities for assembly sequences.

For meta-agents, inhibitory mechanisms have two phases; first, the phase of checking the current state of all the connected agents; second, the phase of comparing the evaluated results with the meta-agents’ tasks. This comparison flags the state of their constituents’ agents. These two processes are accompanied with procedures that calculate the assembly permutations and probabilities. These calculations require the ratio of assembled agents to the overall meta-agent. In addition to inhibitory mechanisms, tagging mechanisms dynamically flag agents and meta-agents with their availability ratio to avoid any wrong connection and to maintain the level of adaptation within the agent system. The activation of tagging mechanisms relies on “IF/THEN” mechanisms to check the state of the agents. Furthermore, the inhibitory mechanism commences the self-organization of the whole agent system to exhibit macro-level regularities.

Determining the coordination mechanism

Coordination systems enable the micro-generating system to synchronize the motion behaviors of each agent within the environment. The coordination system calculates displacement vectors and then transmits the relocation values to the agents' coordinate systems. In addition, different types of agents have their own classifiers to manage the agents' actions within the system. The coordinating systems that are implemented within all the dynamic elements of the agents and the meta-agents, seek to synchronize their behaviors with the identified flows. The coordination of new positions is associated with parallel computation, which set all agents on the same stage to compete.

Each agent has equal opportunity to explore the environment. This equality promotes competition among the agents, as the search for available valences and edges. Agents that are competing for the same available position exhibit convergence, which is an emergent behavior. Agents that converge towards the same target manifests high correlations among all coordination systems. In addition, agents' convergence toward an available target causes competition among agents, to the extent that one agent updates its tag to the connected state. This means that the winning agent is within a certain radial distance of the target. Subsequently, the state tags of target agents and the winner agents will be changed to "occupied" and "connected." Together with this process, when the target position is engaged with one agent, other competitors change their course to the next unoccupied target.

Determining the parameterizing mechanism

The parameterization of aggregation techniques is determined by n-gon structures. In general, parameterization mechanisms are associated with aggregation types of agent systems. The number of gons, the topological connectivity of nodes, and the length of each inter-node within the agent are all described by parameterizing procedures. Accordingly, nodes and inter-nodes (gons) are articulated with static and dynamic variables. The variables that define the n-gons' connectivity are instantiated with a number of connections at each node. In the context of graph theory, the nodes conceptually correspond to valences. The variable connectivity of each valence elaborates the type of aggregation.

The valences' attributes define the possible permutations of agents for assembling the meta-agents. This number is circumscribed to the agents' morphology. The agents' structure imposes geometric constraints that are abstracted from material properties. When agents have a certain number of inter-nodes, then the permutation values of valences should support the agents' morphology. For example, agents with four inter-nodes require valences to consider a value to support four connections. Otherwise, the meta-agents' morphology will deform at the inner angles of n-gons, and it will eventually collapse. Although, the agents' inhibitory mechanism will prevent any drastic deformations by stabilizing the distances and the angles between the valences of each agent. The geometrical definition of connectivity among elements also requires consideration.

4.2.6 Discussion

This experiment used coordination systems and inhibitory mechanisms to organize connectivity among the different types of agents. The controlling mechanisms within this experiment suggest a bottom-up shell structure. A complex surface is achievable both by specifying an exact number of connections at each valence and by modulating agents through the environment to coordinates agents' behaviors. In addition, aggregation techniques provide behavioral classes for each level of aggregation. The interactions among different levels of aggregation, such as agents and meta-agents require mechanisms to adjust their behaviors for assembling the free-form surface. In this sense, approaching stable states within each group avoids flaws within the assembly process.

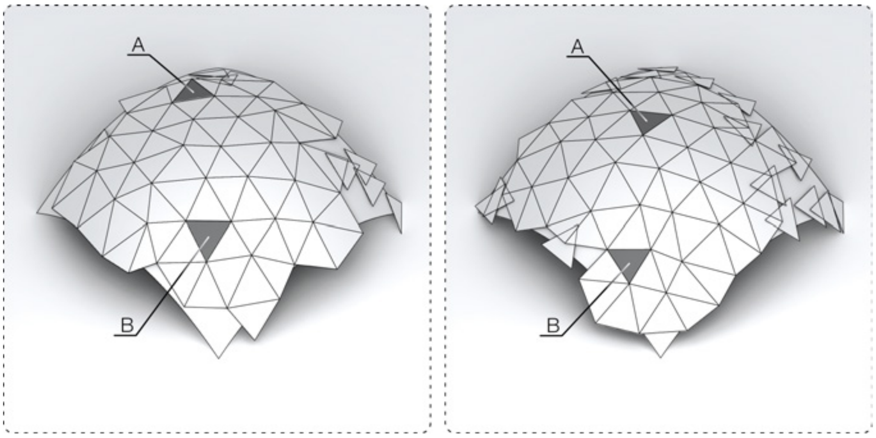


Fig. 4.2.20: Agent's adaptation to the user intervention: i.e., user can control agents assembly by fixing individual agents (A/B).

The synchronization within this process leads agent-based systems towards self-organizing n-gon agents. The adaptation process advances agents to fit within the solution space, where macro-regularities determine the purpose of the agents' assembly.

This process necessitates the imposition of some order on the agents' interactions, such as determining the target surface through which agents only self-organize their behaviors to adequately assemble. This process can be improved further through users' interventions to directly control agents' exploration, see Figure 4.2.20.

4.2.7 Acknowledgments

An agent-based approach for generating free-form surfaces was taught by the author during his time at the Institute for Computational Design and Construction (ICD), University of Stuttgart. Specifically, this approach was investigated by Mark Baur, Boyan Mihaylov, Djordje Stanojevic, Stefana Parascho, and Julian Wengzinek during their seminar, diploma, and master thesis projects.

The role of the author included tutoring and supervising the development of an agent-based computational framework for generating free-form surfaces.

4.3 Fabricational Morphogenesis for Panelizing Free-Form Surfaces

4.3.1 Introduction

This experiment investigates the panelization of free-form surfaces with respect to the constraints of material and fabrication systems. This investigation follows inclusive computational design to include fabricational morphogenesis in the process of panelization. In this experiment, fabricational morphogenesis considers material systems and fabrication tools. These separate drivers are abstracted to develop a hyper-dimensional morphospace, which is defined by a restricted space for negotiating the geometric capacities of a material and the potential of a fabrication system. The implementation of the theoretical morphospace within a generative agent-based system requires inhibitory mechanisms and coordination systems. This experiment proposes to embed material properties into the agents' morphology. This process provides direct participation of agents in the morphogenetic movements. Accordingly, the establishment of behavioral negotiations between agents and fabrication morphospaces modulates the agents' morphology with materials and fabrication constraints, while agents are adapted to environmental factors.

The structure of agents as an individual and a collective system modulates inclusive design computation. In this sense, regularities that arise from collective micro-behaviors gradually evolve within the theoretical morphospaces. The multi-dimensionality of a morphospace provides a cloud of solutions that cover the possible solutions of each dimension. This cloud extends with little tolerance to form a multi-criteria solution space that might include optimal solutions for each dimension. The extended multi-criteria optimization space facilitates the agent's exploration of optimal solutions. The decision-making units of each agent require this optimization space to determine optimal solutions simultaneously. The decision-making unit is also accompanied with inhibitory mechanisms and coordination systems to regulate the behavioral decisions of each agent. Moreover, the decision-making units convert the behavioral objectives of agents into a vector-based system that map behaviors onto coordinating systems.

The preliminary findings of this experiment were published in Baharlou and Menges (2013a,b).

4.3.2 Context

An investigation on the discretization of geometric surfaces provides several approaches to discretizing surfaces with planar faces, which consider the fabricational capacities of flat sheet materials at the early stages of design. These approaches include *triangular meshes*, *quad mesh algorithms*, *PQ-mesh algorithms*, and *planar hexagonal mesh algorithms* (Wang et al. 2008; Wang and Liu 2010), *advancing mesh frontiers* (Atmosukarto et al. 2001; Zhou et al. 2002), *tangent plane intersection (TPI)* (Luping 2001; Troche 2008; Stavric et al. 2010; Stavric and Wiltsche 2011; Manahl et al. 2012), and *variational tangent plane intersection (VTPI)* (Zimmer et al. 2013). In the context of fabricational morphogenesis, tangent plane intersection (TPI) retains the planarity of each component. Moreover, the complex interactions among these individual elements work to panelize the target surface. The correlation between planarity and panelization suggests the use of this method for integrating material and fabrication systems in the early stages of design.

Tangent plane intersection algorithms require underlying mechanisms to distribute point clouds on a surface, then relaxation mechanisms are required (Turk 1992; Walter et al. 1998; Cutler and Whiting 2007) to equalize the distances between distributed points. This process is similar to implementing circle packing algorithms (Beardon et al. 1994; Höbinger 2009; Schiftner et al. 2009) on a surface, which provide a basis for triangulating the surface. The triangulated surface describes the topological connectivity among the distributed plane geometries on the surface.

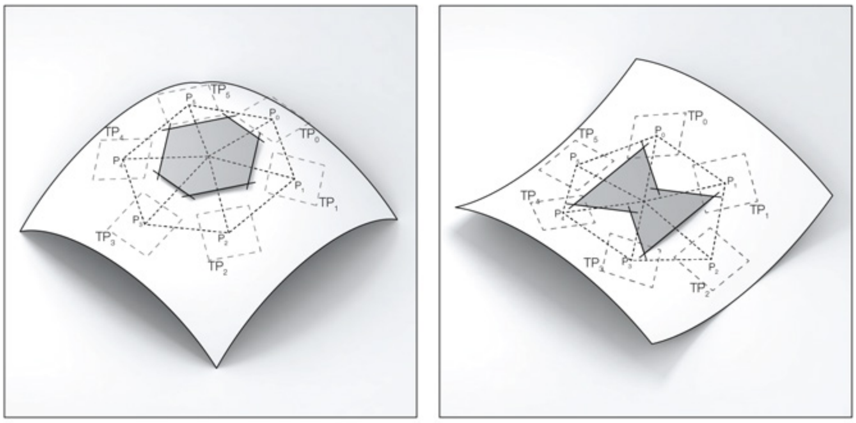


Fig. 4.3.1: Conducting the tangent plane intersection algorithm on synclastic surface (left image) and anticlastic surface (right image).

Associating the plane geometry to the circles, whose centers are tangential to the surface, develops the tangent plane geometry. The locations of this tangency overlap the vertices of each triangle.

Accordingly, the triangulation network enables tangential planes to recognize adjacent planes. This recognition provides a sequential intersection between the main plane geometry and the surrounding planes to gradually polygonize the desired plane geometry. Figure 4.3.1 schematically illustrates the tangent plane intersection of one such plane, where $P = \{P_0,\ P_1,\ P_2,\ \cdots,\ P_5\}$ are the locations of surrounding tangent planes $TP = \{TP_0,\ TP_1,\ TP_2,\ \cdots, TP_5\}$.

The TPI benefits from the Dupin duality algorithm (Wang et al. 2008) to evaluate the intersection vertices of the generated polygon. In accordance with the Dupin duality, Wang et al. (2008) stipulates that the intersection vertices need to be located within a circle, which is obtained from the original coordination of the three intersecting planes, which are coincident at the location of the triangle's vertices. Figure 4.3.2 explains the Dupin duality algorithm where the intersection among three tangent planes $TP_i,\ TP_{i+1},\ TP_{i+2}$ denote the U_i, where $V_i,\ V_{i+1},\ V_{i+2}$ as the locations of tangent planes, denote the red circle. The Dupin duality algorithm studies the relation between the generated circle and the location of U_i.

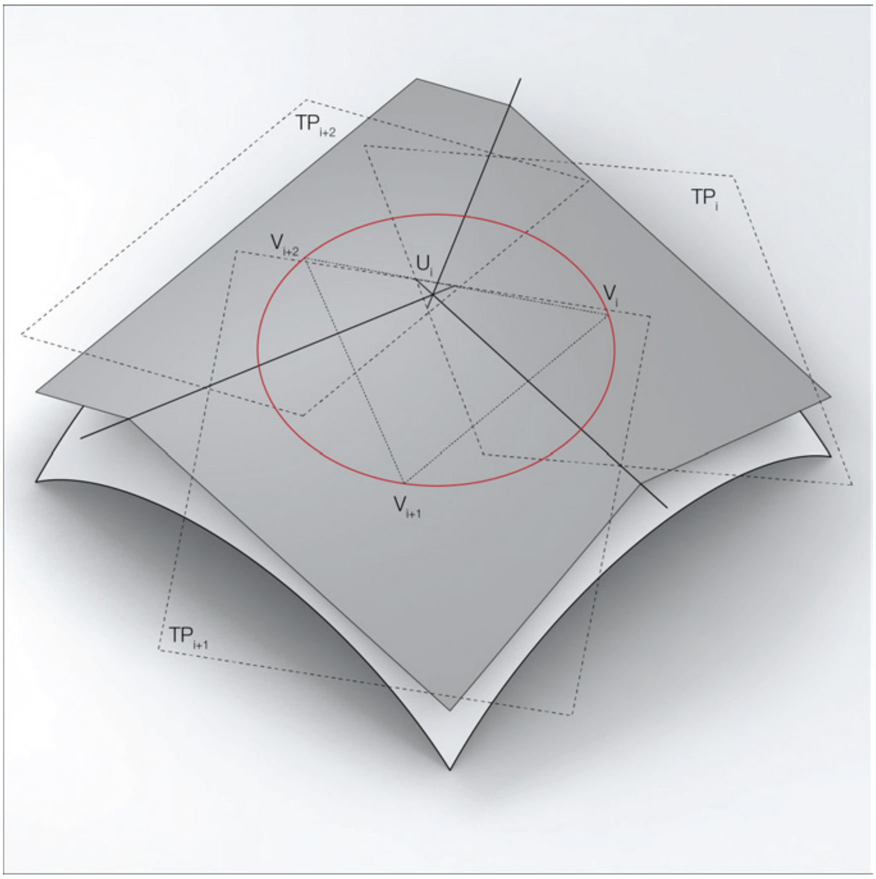


Fig. 4.3.2: The Dupin duality for three intersecting planes.

In a tangent plane intersection (TPI), the planarization process considers the planarity of a material system, such as flat sheet materials. This property is then embedded within the geometric description of the planar surfaces. Other panelization algorithms, such as the quad-mesh algorithm, utilize planarization to approximately planarize each quad mesh face with acceptable tolerance (Liu et al. 2006). The PQ-mesh algorithm, which is developed using discrete differential geometry, is accompanied with a fabricational awareness that considers the constructible properties of fabrication elements (Pottmann 2013).

4.3.3 State-Of-Art

Developing a method that includes material properties and fabrication tools requires a study of natural morphology, where the materials and structures evolve and grow together as inclusive organisms. For example, the plate structures of sea urchins have been widely studied by biologists, engineers, and architects (Cummings 1985, 1989; Trotter and Koob 1989; Wester 1989; Cummings 1990; Bletzinger and Reitinger 1991; Philippi and Nachtigall 1991; Telford 1991; Spirov 1993; Ellers 1993; Seilacher and Gishlick 2014) to gain insight into the morphogenetic movements of this organism, specifically, the growth pattern of plate structures and the interrelation between different plates, as well as, their relation to external factors. Developmental approaches use morphogenetic simulation and materialization to study the organism. Simulation and modeling of the sea urchin not only considers the knowledge of the plates' geometry and structural properties, they also provide a novel insight into design processes.

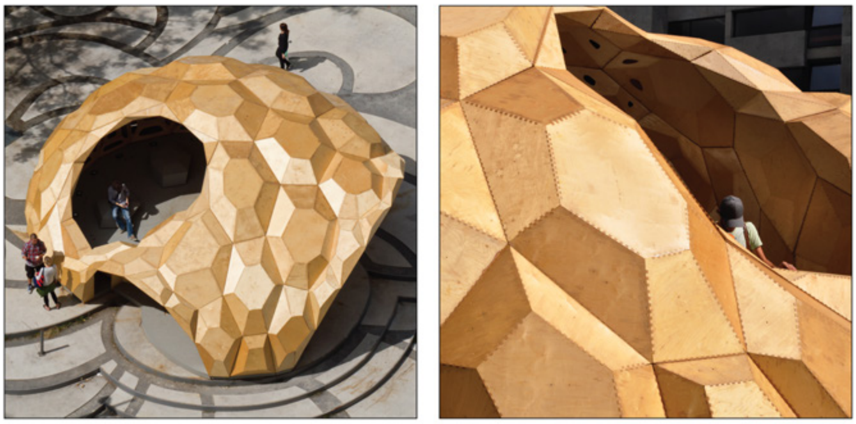


Fig. 4.3.3: ICD/ITKE research pavilion 2011; source: ICD/ITKE University of Stuttgart.

At the research pavilion ICD/ITKE 2011 (Figure 4.3.3), the scalability of the sea urchin as a biological role model was investigated through a consideration of material organization and fabrication tools. Concurrent with the wave of integrative architectural design, this investigation attempted to integrate the geometric properties of flat sheet materials with robotic fabrication tools. This linear computational integration led to the concept of a “machinic morphospace” (Schwinn et al. 2012). The concept of a “machinic morphospace” is used to investigate the limitations and constraints imposed by material systems and fabrication tools (Menges 2013). Theoretically, this concept suggests the development of a solution space (Figure 4.3.4) through an analysis of the relation between the geometric capacities of plywood systems and robotic fabrication environments.

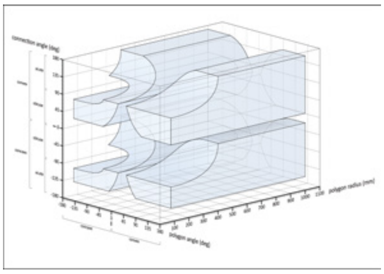


Fig. 4.3.4: A theoretical morphospace based on plywood systems and fabrication tools; source: Institute for Computational Design and Construction (ICD), University of Stuttgart.

4.3.4 Methods

A computational framework was developed with different algorithms similar to circle-packing and simple two-dimensional Voronoi diagrams, such as a clipping algorithm[1] and the tangent plane intersection (TPI) algorithm. The fabrication setup consists of a KUKA KR125/2 robot with a KUKA KPF1-V500V1 turntable along with a HSD ES 350 spindle. The milling tools, and their limitations, were calculated for cutting wood with a thickness of 6.5 mm (Figure 4.3.5). This initial setup was utilized to develop an analytical morphospace that considers material properties and fabricational limitations within the generative process of panelization. However, parameterizing this computational framework required the system to consider various analytical morphospaces from similar material systems and fabrication tools.

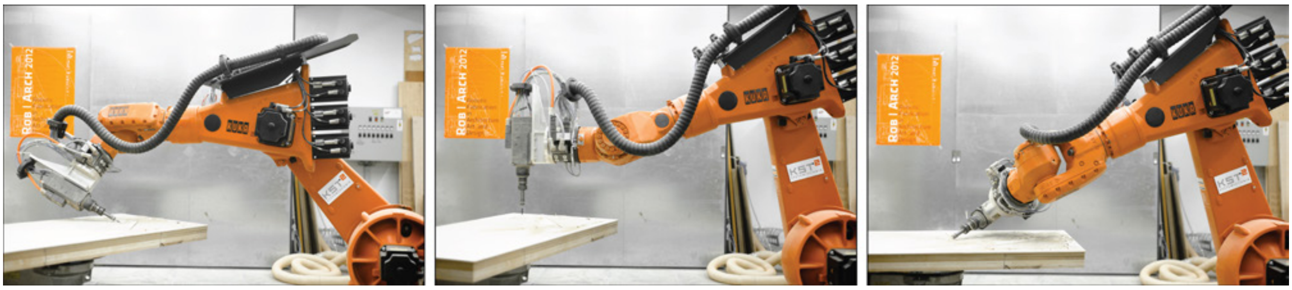


Fig. 4.3.5: The fabrication setup: Milling process of timber plates; source: Institute for Computational Design and Construction (ICD), University of Stuttgart.

---

[1] The clipping algorithm is an extended version of Rutten’s (2005) algorithm, where he developed an algorithm for constructing individual two-dimensional Voronoi cell. For more detail, see Rutten (2005).

4.3.5 Developments

Determining agent properties

Types of agents. This experiment includes a collection of active agents that dynamically contribute to the development of polygonal patterns. The type of agent is limited to the particular system for evolving planar polygonal faces. Accordingly, conceptualizing a generative agent-based system via this type of agent requires a consideration of the multi-agent system. However, the decision-making process of each agent is independent from other agents, while its behavior is under the influence of surrounding agents. Agent behaviors reflect the collection of responses to internal and external influences. In addition, each agent has predefined tasks that it must accomplish using predefined rules. These predefined tasks relate to the geometrical definition of agents, in which dynamic interactions among agents and the environment are correlated with agent morphology. This dynamic morphology is established based on the plane geometry that facilitates evolvements of polygonal faces from the multi-agent interstitial interactions.

Geometric properties. The geometric description of agent morphology is related to the material properties of flat sheets that necessitate the consideration of planar polygonal faces. A planar polygonal face is determined by a plane geometry circumscribed with polygon edges. The transition of these principles into an agent's morphology relies on the consideration of plane geometry as the center of the agent system. The geometrical features of agents arise out of interactions among them. An investigation of the intersections between different plane geometries is required to understand the geometric relations among agents. The results of these intersections generate bounding lines and vertices that demonstrate the geometric forms of agents. The agents' geometry, as polygonal faces, are formed by collecting separated lines that originate from intersections of one agent and its neighbors. Figure 4.3.6 shows the relationship between the plane geometry and the polygonal geometry of each agent. The intersection among the main tangent plane TP_c and other surrounding planes TP_0, TP_1, TP_2, $\cdots$, TP_5 generate polygons P_c and P_0, P_1, P_2, $\cdots$,P_5, where these agents are tangent to the surface at locations V_c and V_0, V_1, V_2, $\cdots$, V_5.

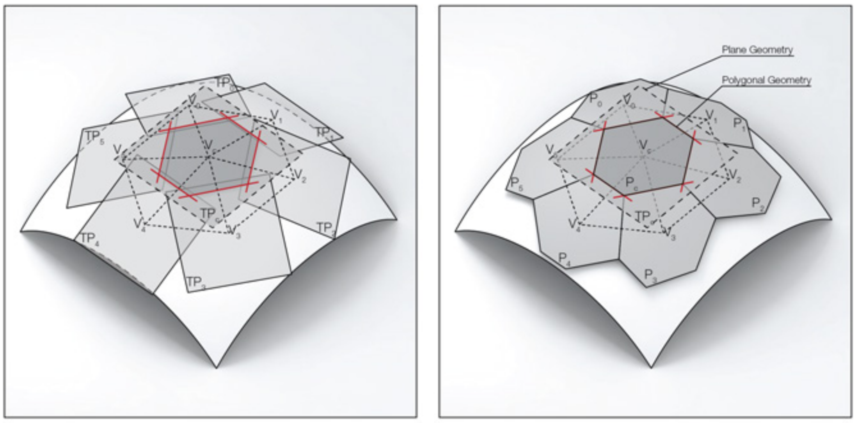


Fig. 4.3.6: Geometrical definition of agents at the level of plane geometry and the level of polygonal geometry.

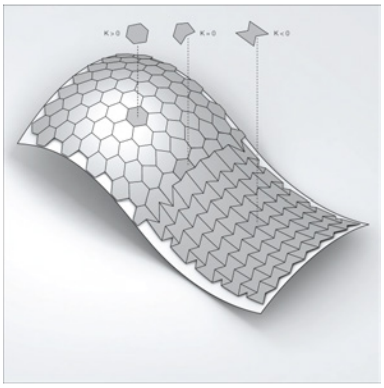


Fig. 4.3.7: Different types of polygonal geometry based on Gaussian curvature.

Agent morphology is structured at the level of the determined and the level of the undetermined. A determined level is the basis for an agent's morphology, which is described through plane geometry. In contrast, the undetermined level is a dynamic description of the agent's geometry, which arises from the interplay among agents. The undetermined geometry of an agent is designated as a polygonal base that limits the outer perimeter of the agent. The polygonal base can take both convex and non-convex forms, such as "bow-ties" and "butterflies," which rely on the curvature of the target surface (Troche 2008). Therefore, the environment, as a target surface with principal curvatures, indirectly establishes the undetermined level of an agent's morphology (Figure 4.3.7). For example, plate-like agents on a synclastic surface will produce a convex polygonal structure. The bilateral negotiations between determined and undetermined states of an agent's geometry conduct the gestalt of agents (formation). This gestalt introduces a dynamic manifestation of the agent's form through behavioral processes and the interactions among agents and environment, which are associated with different embedded tasks within an agent's structures.

Behavioral properties. In this generative agent-based system, behavioral characteristics are inherited from a geometrical constituent of an agent's morphology, which is associated with tasks, and determined by fabricational morphogenetic principles. The fabricational movements of agents are developed through a theoretical morphospace, which includes material capacities and fabrication constraints. The abstraction of these two factors provides simple geometrical descriptions and multi mathematical constraints that are represented within a theoretical morphospace. Accordingly, the establishment of an analytical morphospace requires several "IF/THEN" procedures to examine the state of agents with the fabricational movement. Any deviations from this movement generate a signal to maintain the states of the agents. The morphogenetic movement of the agents determine the relationship between agent and agent, at one level, and the relationship between agent and environment, at another level. In these relationships, agent behavior is characterized by geometric relations and negotiations with the environment or field. The geometric relations of agents consider intersections between plane geometry through the clipping algorithm (synclastic surfaces) and the tangent plane intersection (TPI) algorithm (anticlastic surfaces) to generate the vertices of polygonal faces. In addition to these algorithms, the Dupin duality provides the development of "IF/THEN" procedures within the inhibitory mechanism to examine the generated vertices, so they will fit in the acceptable tolerances. The interstitial interactions of agents use the adhesion and repulsion behaviors that were developed in the previous experiments. Figure 4.3.8 illustrates a simulation of these behaviors on plate-like agents.

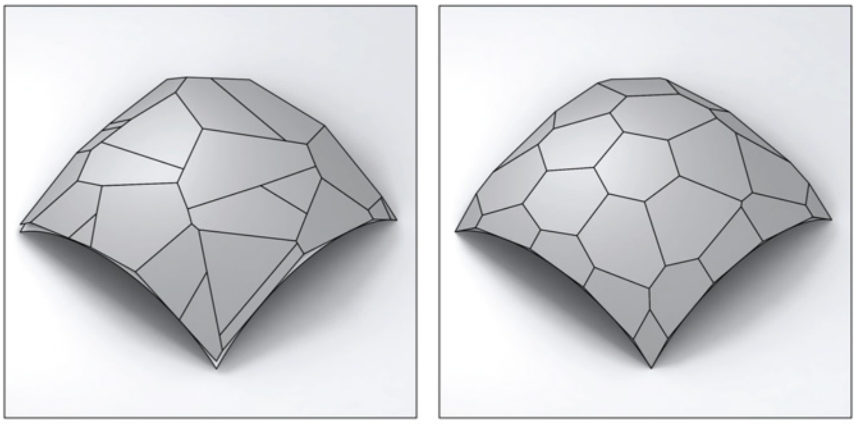


Fig. 4.3.8: Agent-Agent interactions; left image: Adhesion behaviors; right image: Repulsion behaviors.

In the context of morphodynamics, the agents' negotiations with the environment use intrinsic and extrinsic drivers, which also affect the agent's gestalt. The environment, as an internal driver, coordinates the behavioral states of agents on the target surface. The geometric description of agents is tangent to the surface. Within this context, the tangency of agents determines the normal of the plane geometry is parallel to a surface normal vector, which is generated at the contact coordination. The principal curvatures of the surface provide internal drivers for tangential agents to coordinate their morphological behaviors. The external features of the environment directly change the course of the agents' actions. The geometrical and mathematical features of the surface determine the extrinsic influences over the agents. Extrinsic effects, such as surface edges, define the field of actions for agents. For example, the inner and outer loops of the target surface bind agents to the topological definition of the surface. In addition, the geometrical behaviors of the agents follow the topological connectivity of agents, which corresponds to the topology of the surface. Moreover, the surface edges benefit from adhesive and repulsive behaviors, which impose forces on the agents (Figure 4.3.9).

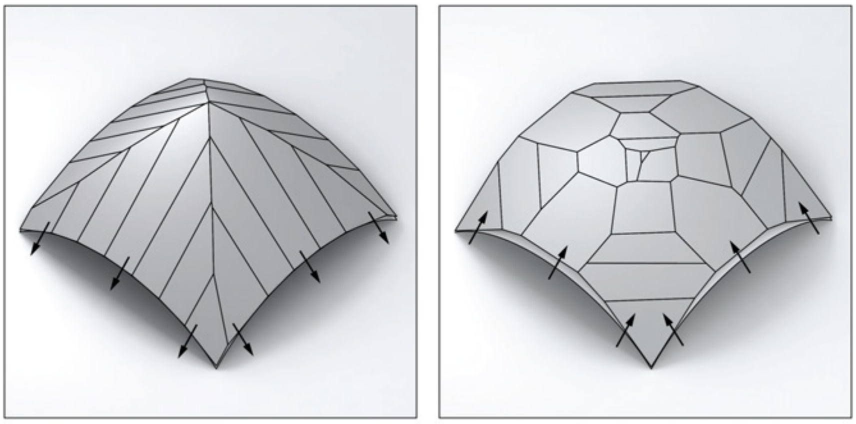


Fig. 4.3.9: Agent-Environment interactions; left image: Adhesion behaviors; right image: Repulsion behaviors.

Determining a contextual environment or field

In this generative agent-based system, an environment that is considered a target surface is a passive field. This field uses agents' explorations to cover the whole area of the field. The agents control their distances from other agents to maintain their areas. In the context of the differential geometry of surfaces, the passive field is considered a mathematical surface that provides a geometrical description at any given point. This description could be a normal vector or principal surface curvatures. The normal vector, which is perpendicular to the field, defines one agent attribute that is used to develop a plane geometry. An agent's behavioral variation is informed by principal surface curvatures, which also provide information on the intersections among plate-like agents. Checking the curvature at an agent's location on the target surface and comparing it with neighboring agent curvatures determines the type of agent morphology (Figure 4.3.10).

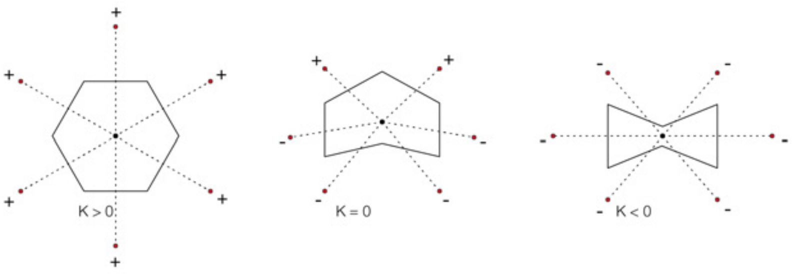


Fig. 4.3.10: A conceptual relation among agents' morphology and surrounding agents' curvatures, where *K* represents Gaussian curvature.

The topological features of the surface, such as the inner and outer loop, form the agent's connectivity. Agents require these topological features to recalculate their connectivity, when they are close to the edges. Synchronization between the topological connectivity of agents and the manifold categorizes interactions among agents. For example, one group of agents, close to the edge area, is tagged as naked agents. This group of agents requires an artificial bounding area to accomplish the intersections between plate-like agents by enclosing the polygonal face (Figure 4.3.11). Therefore, the agents' tags correlate to the types of agent interactions used for computing their morphological behaviors.

Fig. 4.3.11: The intersection of naked agents with surrounding plate-like agents TP_a, TP_b, TP_c and the bounding plane *BPa*.

Determining the interaction behaviors or rules

The behavioral structure of this generative agent-based system focuses on both micro and macro levels of interaction. These two levels are indicated for specific tasks. At the micro-level, the interrelations among agents are investigated through behavioral negotiations over different panelizing algorithms that describe the geometric relationships between agents. Panelizing algorithms, such as the clipping algorithm and the tangent plane intersection (TPI) algorithm, determine the geometric representation of agents that is continuously transformed within the polygonal structures during the process of modeling. Figure 4.3.12 illustrates the clipping algorithm.

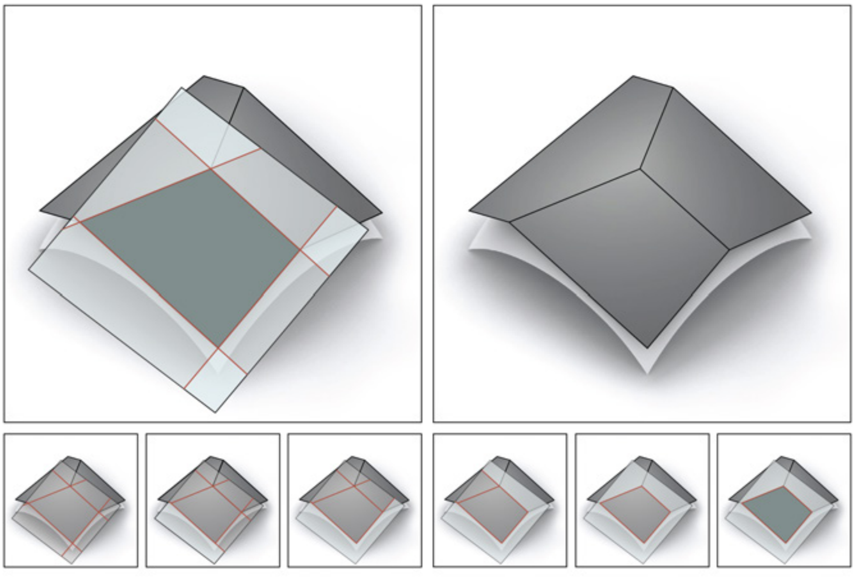


Fig. 4.3.12: The clipping algorithm; bottom row represents the sequence of clipping the base geometry to develop the final form determined by the agents.

A comparison between these two algorithms reveal similar methods for computing the vertices of polygon geometry. The difference between the two is found in their approach to finding adjacent agents. The clipping algorithm considers a local approach to finding adjacent agents. This approach to finding neighboring agents limits the applicability of the clipping algorithm to synclastic surfaces. In contrast, the TPI algorithm interplays between local and global levels. At the global level, triangulation algorithms determine the topological connectivity of agents. At the local level, the topological connectivity of agents determines the intersection between agents to develop their polygonal geometry. In this sense, the TPI algorithm has comprehensive knowledge of agent connectivity. This method of finding adjacent agents fosters the applicability of the TPI algorithm for both synclastic and anticlastic surfaces.

At the micro-level, both algorithms require controlling mechanisms to maintain the internal cohesion of the agent's morphology. The proper algorithms are embedded in the micro-level avoiding disruptions to the system's cohesion. One of these algorithms is responsible for avoiding self-intersection among polygonal structures. To accomplish this, the algorithm sorts the sequence of vertices in a clockwise or counter-clockwise order. Figure 4.3.13 compares two different orders of intersected points $P_0, P_1, P_2, \cdots, P_5$. Employing the (counter-) clockwise algorithm alters undesirable agent morphologies–ones with self-intersected sides–to desired morphologies.

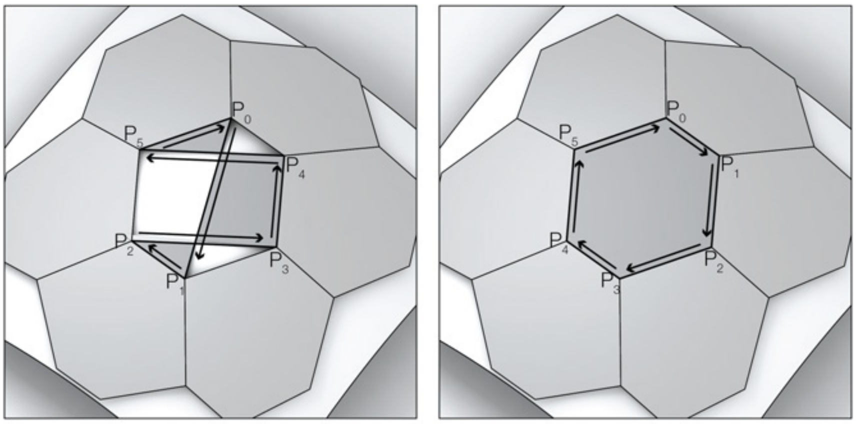


Fig. 4.3.13: The implementation of a clockwise-ordered algorithm to sort intersected points.

The enhancement of an agent's intersections requires the Dupin duality algorithm, which checks the intersection between three plane-like agents and controls the cohesion of the agent morphology. This algorithm determines whether the intersected vertices are adequate for circumscribing the plane-like agents. Any failure of the vertices within the agents' morphology release predicted behaviors, such as eliminating the agents, applying forces to change the agents' location, or activating different intersection mechanisms to bound the agents' morphology (Figure 4.3.14). For example, in the latter case, the generated circle between agents might determine an artificial intersection position. By iterating this mechanism for all of the three plane-like agents the polygonal morphology will be generated. However, applying this mechanism will generate tolerances at the shared edges between the agents. Decreasing this deviation might benefit from finding the average coordination between shared vertices. In addition, relocating the original vertex towards the mean average location might change the planarity of plate-like agents, but it will also decrease the deviation among the shared edges of agents.

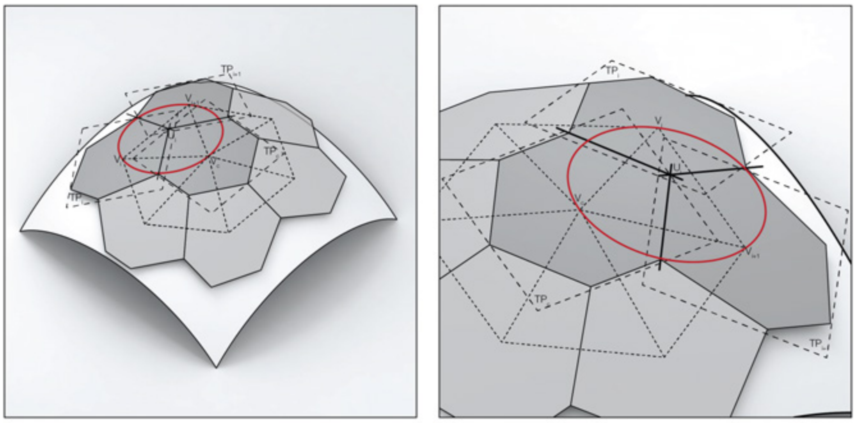


Fig. 4.3.14: The implementation of the Dupin duality algorithm to control intersected point U.

At the macro-level, agent behavior relies on the internal and external influences of the environment. The internal factors regulate the behaviors at the micro-level with normal vectors and principal surface curvatures. The normal vectors at the surface are developed within an algorithm that maintains the tangential properties of the agents within a specific distance to the target surface. Moreover, the algorithm retains the normal of the plane geometry, which remains perpendicular to the surface through identified angles. However, the external factors that change the agents' behavior bound the agents' activity to the surface. The surface edges affect the agents at the level of topography and the level of topology to change their behavior and morphology. The topography of the surface alters the behavior of agents by driving them away or attracting them to the surface edges, while the surface topology reconfigures the agents' connectivity network. Accordingly, tagging mechanisms within the inhibitory mechanism dynamically update the topological relation among agents by differentiating the inner agents from the outer agents or naked agents.

Determining the inhibitory mechanism

Fabricational morphogenesis requires mechanisms to inhibit the generative potential of systems toward constructible elements with specific tools and materials. In relation to the planar polygonal elements, the constraints of materials and fabrication tools are investigated to develop a theoretical morphospace. The theoretical morphospace is structured with the geometrical limitations of the material system and the related constraints of the robotic fabrication tools, for example, plywood materials and a set of robotic fabrication tool (Figure 4.3.15).

Fig. 4.3.15: The robotic milling process: An interplay between material and fabrication setups; source: Institute for Computational Design and Construction (ICD), University of Stuttgart.

In the context of this experiment, the development of a theoretical morphospace, which considers these two factors, is determined by three main constraints: "connection angles," "polygonal radii," and "polygonal internal angles," see Figure 4.3.16 (Menges 2013, p. 42). Each of these constraints indicate one dimension of the hyper-dimensional morphospace that can be followed by other mechanisms to inhibit the intersection problems and to negotiate the principal curvatures of the free-form surface.

In particular, the fabricational morphogenesis includes a theoretical morphospace of robotic fabrication. Each of the constraints relies on the direct relations between geometric capacities of the materials and robotic fabrication tools. The plate-like agents interact directly with the target surface and the robotic fabrication system. In addition, the geometric features of the materials impose their properties on the fabrication processes. The

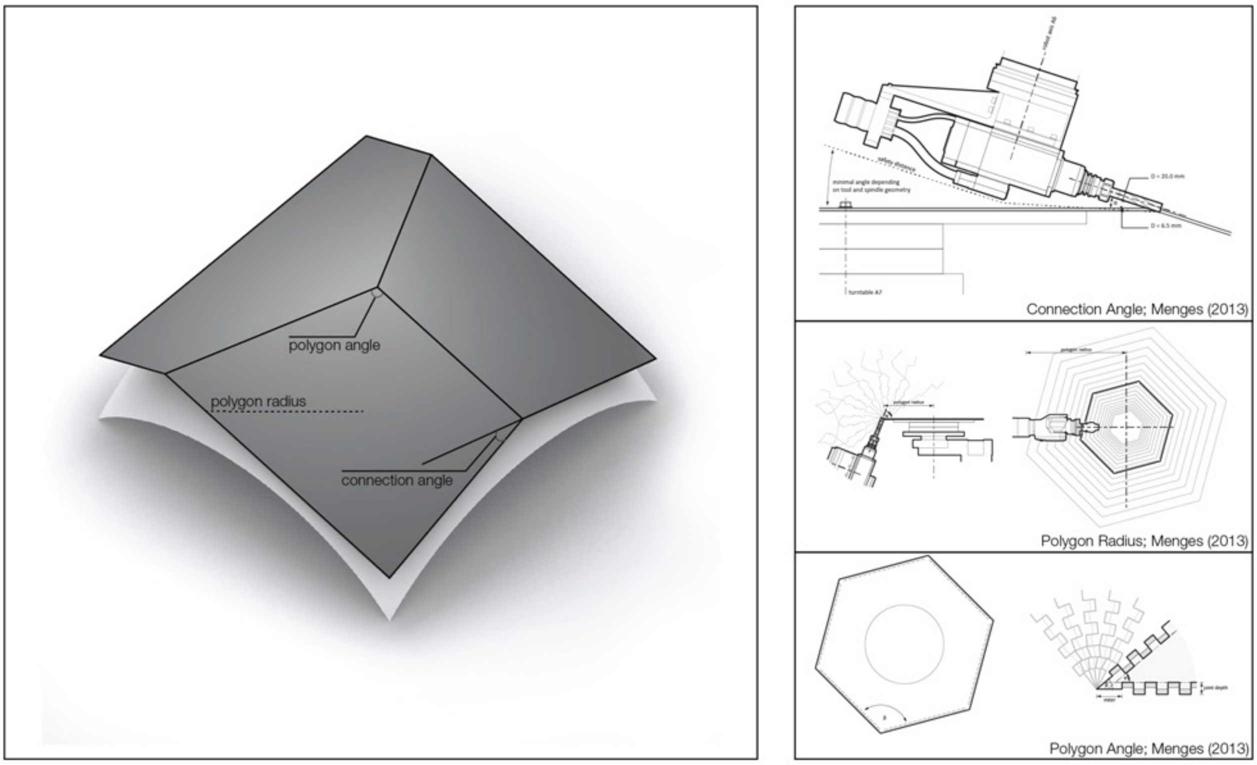


Fig. 4.3.16: Fabrication and material constraints: "Polygon angle," "polygon radius," and "connection angle" (Menges 2013); source: Institute for Computational Design and Construction (ICD), University of Stuttgart.

correlations between plate-like agents are required to address these constraints to prepare the constructible elements. Accordingly, the inhibitory mechanism considers the relationships among plate-like agents, the target surface, and the robotic fabrication.

These mechanisms consist of several "IF/THEN" procedures to examine the relation between plate-like agents and the target surface. These conditional procedures coordinate the flow of agents towards the constructible area of the theoretical morphospace, which also considers almost all of the material and fabrication conditions. The establishment of these conditions relies on the development of an analytical morphospace through an experimental empirical study of different materials and fabrication setups. Furthermore, the inhibitory mechanism benefits from tagging procedures to flag the interrelation among agents and the target surface. For example, the topological edges of the target surface affect the relationships among agents by categorizing agents as either outer agents or inner agents. This categorization requires appropriate procedures to flag each class of agent for further implementation of intersection mechanisms.

Limiting agent behavior with selected experimental constraints, such as connection angles, polygonal radii, and polygonal internal angles, necessitate the development of behavioral rules that maintain agents within an acceptable range of the theoretical morphospace. In this experiment, a synclastic surface with the Gaussian curvature minimum 3.901683797*E*-7 and maximum 4.725764133*E*-7 is investigated to test different behavioral layers for solving these problems (Figure 4.3.17). If these constraints are not solved, the morphospace will need to be redesigned. These layers of action are developed similar to the mechanisms that are used to avoid obstacles, and will prevent agents from reaching the impossible area of the morphospace.

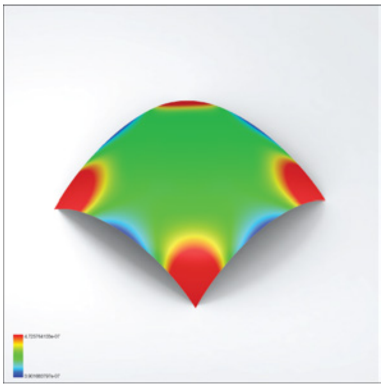


Fig. 4.3.17: An analysis of surface curvature.

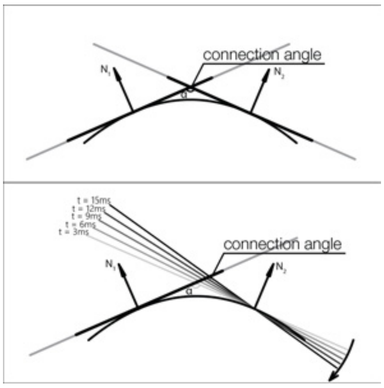


Fig. 4.3.18: The controlling mechanism of connection angles.

[i] Controlling the angle between a plate-like agent and the neighboring agents require an examination of the angle of agents with their adjacent agents. Angles that deviate from the acceptable range require agents to rotate towards the constructible areas of the morphospace. This process considers two similar methods for rotating plate-like agents along the shared edges that are tangential to the surface (Figure 4.3.18) and relocating agents on the target surface. The latter method relies on the target surface curvature, in which the tangent agents try to find locations on the surface that generate appropriate connection angles. In this sense, the rotation angles must be converted into the transformation vectors. The generated vector is the output of one behavioral layer, assembled with other behavioral layers (Figure 4.3.19). However, the exploration of adequate angles relies on the curvature of target surface. Agents that are tangential to the surface might not be able to find the right spot to solve the connection angle problems.

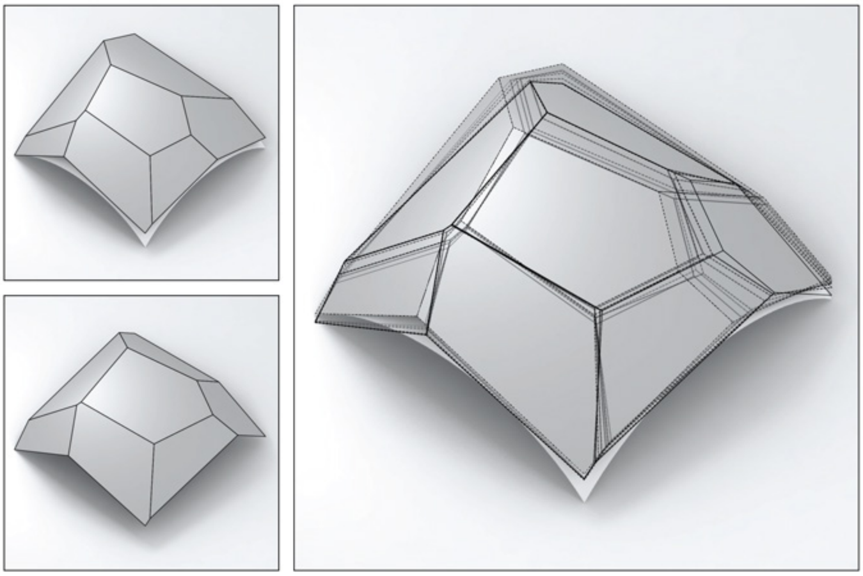


Fig. 4.3.19: The simulation of a connection angle mechanism; left-top image: The initial state; left-bottom image: The final result; right image: The process of approaching to the appropriate angle.

[ii] The polygonal radii of the agents rely on the area of the polygonal face, where the areas of the geometric agents are calculated at each iteration. When “IF/THEN” mechanisms determine that the agents’ areas are not within the acceptable range then, from the longest edge of the polygonal face, the agent’s cell is divided into two agents. The new generated agents inherit all the characteristics of their ancestor of the same type (Figure 4.3.20).

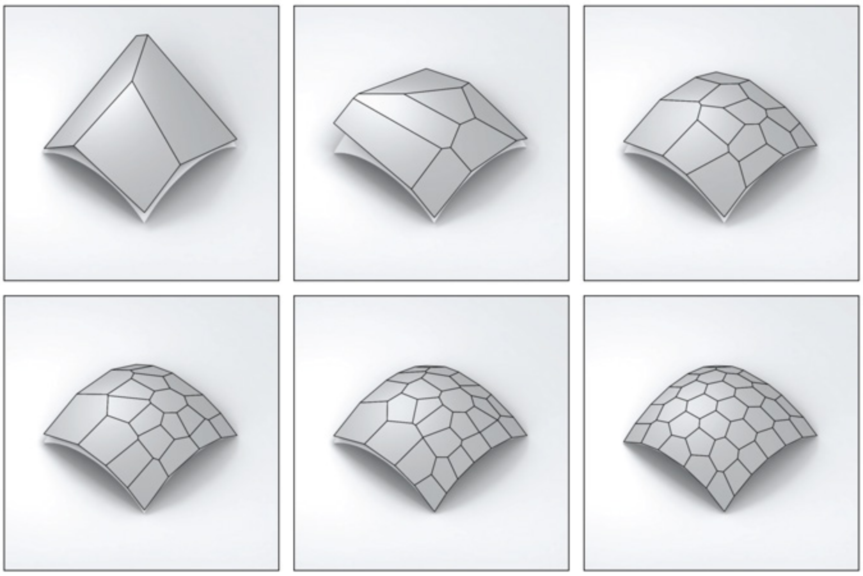


Fig. 4.3.20: A simulation of the polygonal radii mechanism via the agent division mechanism.

Figure 4.3.21 illustrates the results of applying this mechanism. It also shows the inheritance among generated cells and divided cells. The time delay between relaxing agents on the surface and applying the controlling polygonal radii mechanism provides enough time to adjust the distances between adjacent agents. Furthermore, the interaction behaviors among agents, such as repulsion and attraction behaviors, increase the time-frame to achieve equilateral states. Integrating a time-frame within the process of controlling radii facilitates the process of self-organizing to erect different arrangement of cell divisions.

[iii] During the process of cell division, the topology of agent connectivity dynamically adjusts itself to the elimination of one agent and the originating two new agents. The dynamic interrelations between agents demonstrate that the topological network of the system can adapt itself to variations in the number of agents. This adaptability requires a mechanism to control the polygonal internal angles by changing the number of adjacent agents. There is a direct relation between the number of agent connections and the polygonal internal angles. Reducing the number of connections will decrease the number of angles and vice versa. This description is based on

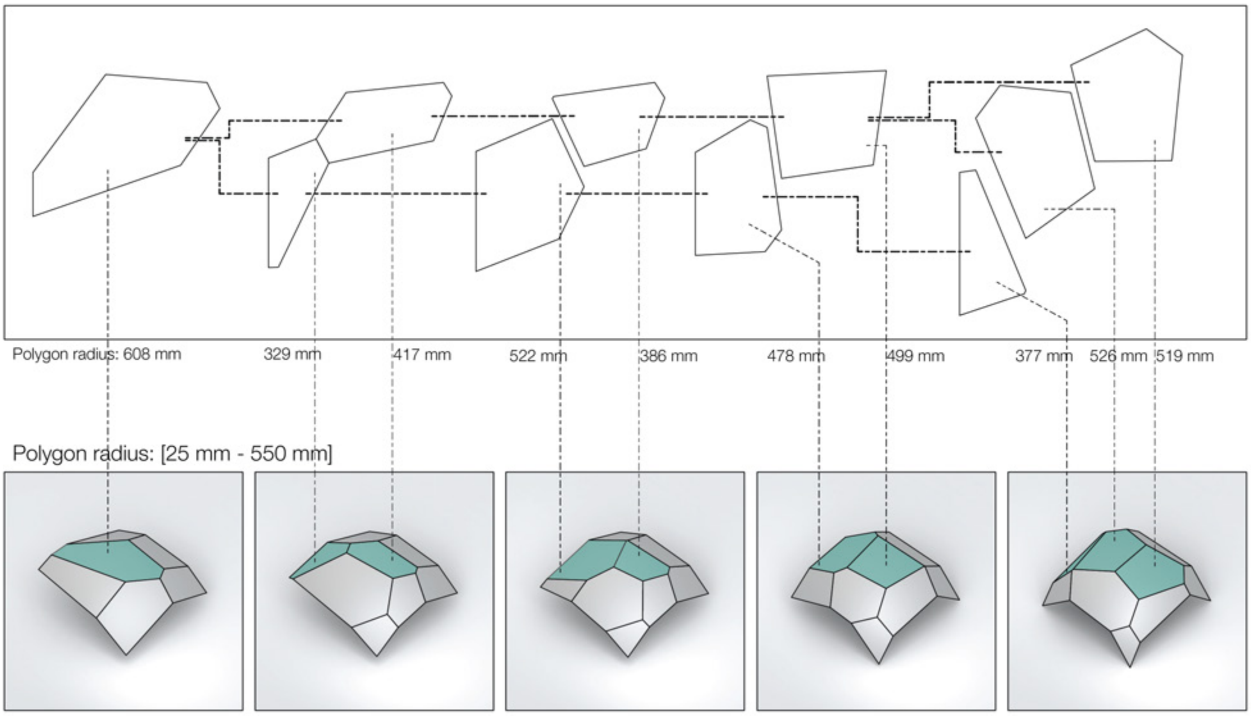


Fig. 4.3.21: The analyses of implementing the polygonal radii mechanism.

the equation used to find the internal angle of a regular polygon, $(n-2) \times 180°/n$, where n is the number of vertices or sides. Thus, the number of sides is equivalent to the number of adjacent agents. This layer of action is under the influence of other mechanisms, such as primitive layers of repulsion and adhesion behaviors. The assembly of all the layers of behavior dynamically alters the number of adjacent connections. Figure 4.3.22 illustrates the application of all of these behavioral layers in which the generative model tries to self-organize with one morphospace configuration.

Agents are allocated to the level of topology and the level of topography. The topological space determines the relationship between agents together and the topological characteristics of the contextual environment. These two spaces are mapped through a coordinate system to facilitate communication between the topological and topographical agents. At the topological level, the coordination system adapts agents to a topological definition of the mathematical surfaces. At the topographical level, the coordination systems maintain agents within the defined boundary of the environment.

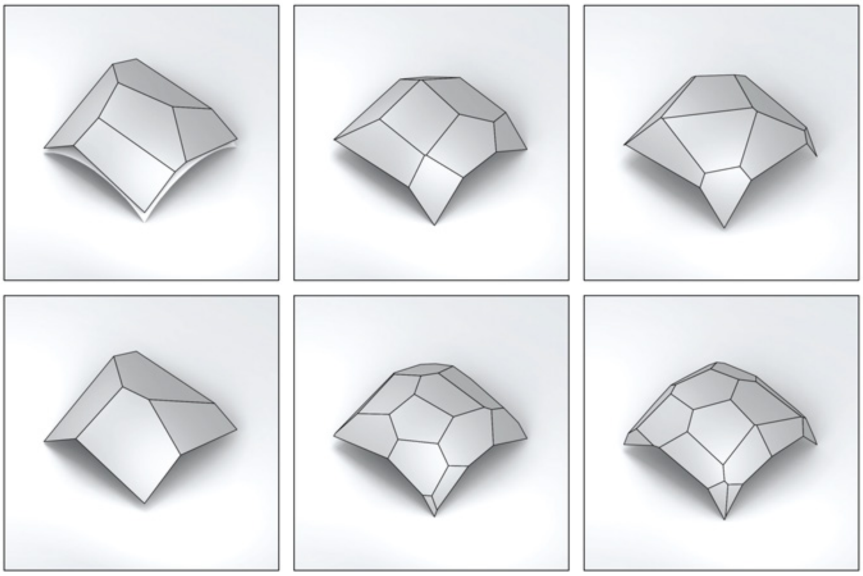


Fig. 4.3.22: Generative agents-based design computation for plate-like agents; top row images: Initiated with five agents, connection angle $[40^\circ - 140^\circ]$ and polygonal radii $[25\ units - 550\ units]$; bottom row images: Initiated with four agents, connection angles $[30^\circ - 150^\circ]$ and polygonal radii $[50\ units - 350\ units]$.

Correlations between these two levels are adjusted by mapping mechanisms that relate the topology of a manifold to the surface topography. For example, the relation between the inner and the outer loops are investigated at the topological level and mapped to the topographical space. In addition, each agent recognizes adjacent neighbors. Accordingly, agents determine that they are surrounded with other agents or that they are located at the edges' border (Figure 4.3.23). In the topographical space, the classifier system utilizes a translation mechanism that adopts the topological networks to the topographical system, which is determined by a free-form surface.

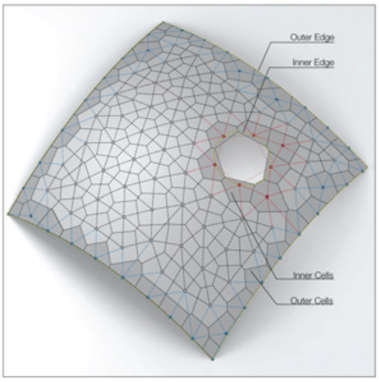


Fig. 4.3.23: Overlapping topological and topographical spaces on tagging procedures.

Determining the coordination mechanism

The main agencies of the inclusive design computation, such as fabricational morphogenetic and environmental factors, are developed within the coordinating system and the inhibitory mechanism. The generative agent-based system requires mechanisms to adapt itself to the gradual changes of inclusive drivers. The environment, as a target surface, is a parameterized factor that can be altered dynamically during the design process. However, different mechanisms are developed to control the range of possible changes and avoid drastic alterations. For example, the coordinating system adapts agents to transit from synclastic surfaces to anticlastic surfaces, which require adaptations to normal vectors and surface curvatures.

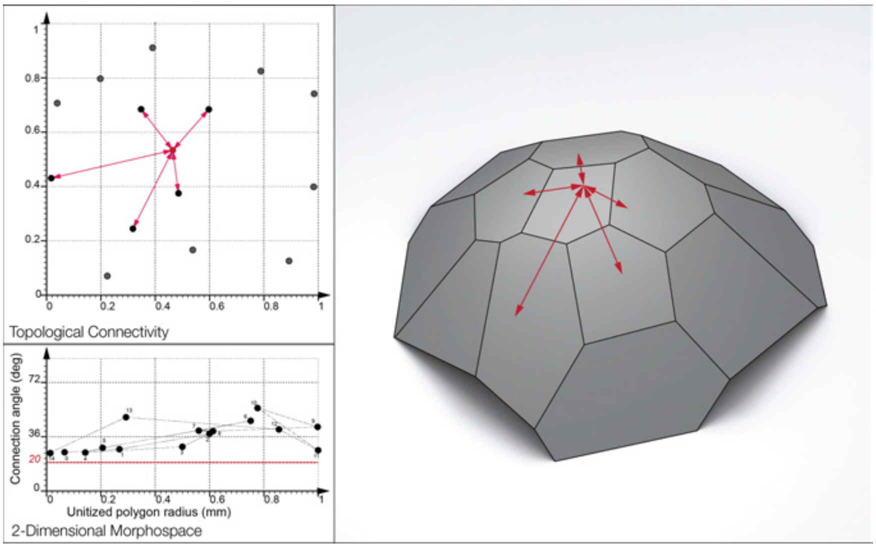


Fig. 4.3.24: Conducting an inhibitory mechanism and a coordination system to interrelate topological connectivity (left-top image), morphospace or parametric space (left-bottom image), and Euclidean space (right image).

Another level of the coordination system relates fabricational morphogenetic aspects to the inhibitory mechanism. Fabricational effects of the inclusive design computation are translated into a theoretical morphospace, which provides analytical procedures within the inhibitory mechanism. The parameterization of the theoretical morphospace is correlated with differentiating the conditional states of "IF/THEN" procedures through which agent exploration corresponds to the fabricational movement. In this sense, the development of a theoretical morphospace begins with the geometrical principle of material organization, which correlates to constraints within the process of robotic fabrication. The inhibitory mechanism considers the parametrization of the constraints that coordinate agent behavior. It is important to note that the underlying principles of theoretical morphospace are fixed but the articulations of its dimensions are modulated (Figure 4.3.24). For example, the possible range of angles between plate structures can be parameterized to explore different panelization patterns.

4.3.6 Results and examples of implementation

Example 01: Rob|Arch 2012

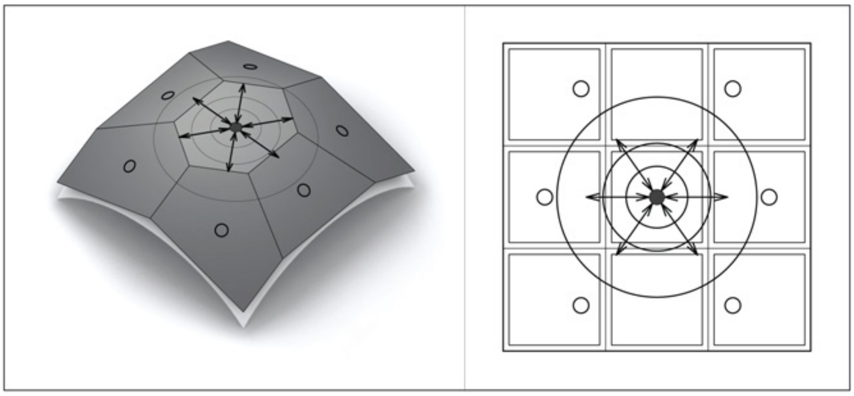


Fig. 4.3.25: A schematic illustration of agent-agent and agent-environment interactions via two mechanisms of motion behaviors and geometrical mechanisms.

In Rob|Arch 2012, a generative agent-based design computation was investigated to blur the distinction between the design intention (panelizing a free-form surface) and the construction process (robotic fabrication). The agent design tool benefited from two main libraries, which dealt with the agents' morphologies and their motion behaviors (Figure 4.3.25). In this sense, the agents' morphology, which was determined through a geometrical mechanism, relied on clipping algorithms. This algorithm constrained the design process to synclastic surfaces. Figure 4.3.26 illustrates the computational model of the prototype, in which generated plate-like agents are fabricated for construction.

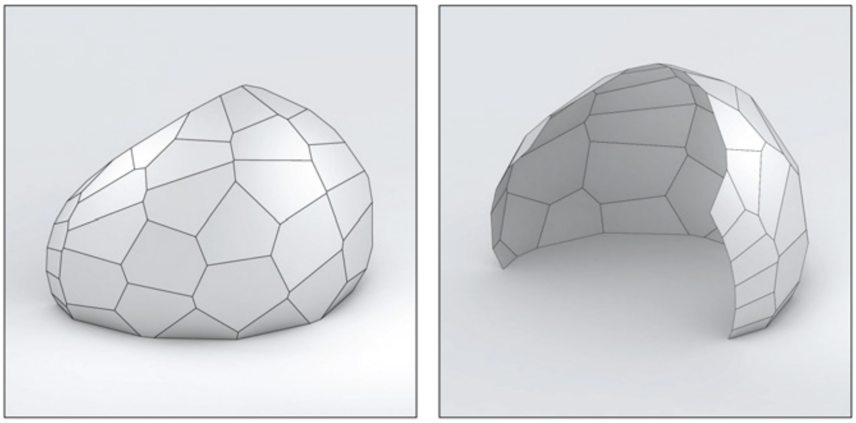


Fig. 4.3.26: The simulation model of the prototype.

The goal of this tool was to panelize synclastic surfaces to generate fabricable components with plywood plates. The agent design tool was accompanied with fabrication computational tools to determine the robotic milling paths. Both of these sequences were accomplished using the *Echinoid*[1] Add-on, which was developed for the Grasshopper[2] plug-in. Figure 4.3.27 shows the details of the physical prototype, which was developed using three steps of design: plate-like agents, tool-paths generating, and fabrication processes. In the agent design tool, producibility of generated plate-like structures was evaluated by mechanisms that determine fabrication and material constraints. If the agent-based system was unable to find the right configuration, then a redesign or new implementation of fabrication tools became essential. This modification furthered the specific criteria for the theoretical morphospace, and the possible solution space was expanded.

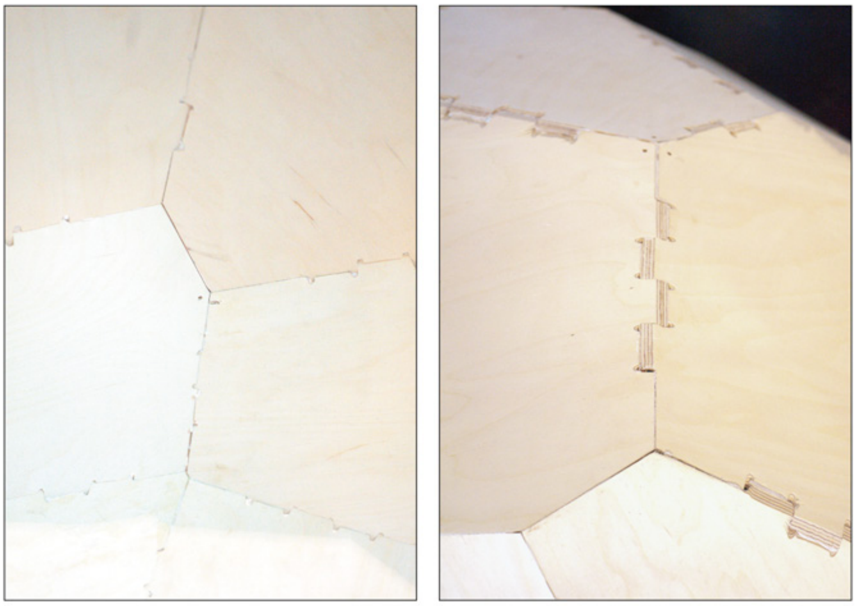


Fig. 4.3.27: Details of the physical prototype; source: Institute for Computational Design and Construction (ICD), University of Stuttgart.

The generative agent-based system was associated with the motion behavioral mechanism. This mechanism was developed as a primitive behavior to explore the overlay of topological and Euclidean environments. Parameterizing the behavioral mechanisms gave users the ability to alter agent behavior on a target surface. The agent design tool benefited from the integration of Constrained Generating Procedures (CGPs) and behavior-based systems. The developed generative agent-based system was analyzed with fabrication and material constraints. Figure 4.3.28 represents the built physical prototype, where the assembled plates are consistent with computational plate-like agents. This correlation proved that the generative agent-based system could be used to panelize surfaces. However, this agent-design tool requires an enhancement

---

[1] see Subsection 4.3.8.

[2] Grasshopper is a visual programming interface for Rhinoceros.

of geometrical mechanisms that can also panelize anticlastic surfaces.

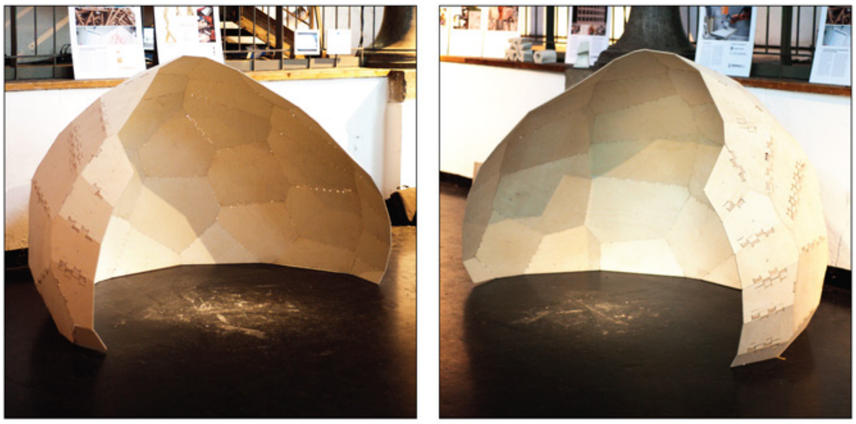


Fig. 4.3.28: The final physical prototype; source: Institute for Computational Design and Construction (ICD), University of Stuttgart.

Example02: Landesgartenschau Exhibition Hall 2014

Fig. 4.3.29: Landesgartenschau Exhibition Hall 2014; source: ICD/ITKE/IIGS University of Stuttgart.

The Landesgartenschau exhibition hall (Figure 4.3.29) was constructed to demonstrate the potential of integrating design computation and fabrication processes. Similar to rob|arch 2012, the agent design application, the computational tool that was developed for this project benefited from the elaboration of agent-based systems. This computational tool utilized a tangent plane intersection (TPI) algorithm to panelize a free-form surface. Moreover, this computation tool implemented a machinic morphospace to adjust agent behavior with fabrication and material constraints. The computational tool was strategically developed to deal with transitions from synclastic to anticlastic surface areas. According to Schwinn et al. (2014), a vector field was used to avoid areas with zero curvatures. This vector field would lead agents to areas of increasing curvature (Schwinn et al. 2014, p. 118). Similar to plate-like agents, this generative tool distributed agents on the target surface and after several iterations the agents converged towards their relaxation states.

4.3.7 Discussion

Agents' morphology is determined by the geometrical definition of the material organization. The agents' behavior is partially determined by their geometry. Interactions among agents are controlled by intersections between two planes. When agents are surrounded by their neighbors, their interactions are sequentially described through the clipping algorithm and the tangent plane intersection (TPI) algorithm from which each agent computes its intersection with surrounding agents sequentially in a clockwise or counter clockwise order. The result of these interactions is a set of points that are collected in the data-structure of the main agent. From these points, an agent is formed into a convex or non-convex polygonal cell. The behavioral cell formations are accompanied with other inclusive design aspects. Agent behavior also follows the fabrication constraints that are abstracted from specific fabrication tools. Accordingly, the development of a theoretical morphospace is related to environmental factors, such as the inner and outer edges of the surface and the surface principal curvatures. These parameters are embedded within the surface in which they are computed from the second derivative of the surface equations. These parameters are available to agents when they explore the surface. This environmental information is distinct from the agents, but is still fully associated with the agents as internal properties.

The theoretical morphospace that defines a possible solution space requires behavioral mechanisms to lead agents toward constructible areas. The use of different rules proved the possibility of developing a behavioral mechanism that can produce fabricable elements. However, retaining the tangency of an agent to the target surface demonstrates the limitation of morphospace. Therefore, releasing agents from the surface is similar to drastically redesigning the surface to change the possible solution space. This example demonstrates that the theoretical morphospace only considers fabricational morphogenesis, which is required to consider the effectiveness of the environments. Due to the agents' dependency on the environment, it is necessary to consider the environmental layers in parallel to the morphospace layers.

Expanding the machinic morphospace would require the utilization of different types of materials or fabrication tools. For example, the connection angles are directly related to the depth of the milling tools. Increasing the depth of the milling tools will also increase the connection angles of the morphospace. The application of these methods consequently modifies the acceptable range of connection angles. It is notable that this process is applicable for two other problems.

4.3.8 Acknowledgments

The Echinoid add-on was developed within the Rob|Arch 2012 workshop. Its authors were Ehsan Baharlou, Tobias Schwinn, and Oliver D. Krieg.

The role of author in this project included the development of a generative agent-based design tool that was used to panelize a surface.

4.4 Agent-Based Morphodynamic Generation of Fibrous System

4.4.1 Introduction

This experiment focuses on an inclusive design computation to integrate robotic fabrication processes into the early stage of composite design. In this study, the generative agent-based system computes the robotic tool-paths to lay fibers on a target surface. The generative agent-based system is facilitated by three main agencies: fabricational morphogenesis, environmental factors, and the performative aspects of design. This experiment investigates a methodological approach of extracting the principles of natural fiber composites, for example, the water spider (*Argyroneta aquatica*), which makes its home by spinning a silk enclosure underwater. The main objective of this experiment is to develop a computational tool for generating the tool-path of fiber placements, which is applicable for robotic fabrication in the context of architectural design. The agent-based system is developed to represent the robotic end-effectors. The structural and mechanical properties of fiber systems, as well as, the principles of fabricational morphogenesis were used to generate the robotic tool-paths. This inclusive process interacts constantly with environmental factors. The environment is a formwork that requires fibers for stabilization and reinforcement. Accordingly, the formwork is modulated by agents, whose behavior is modulated, in turn, by the environment.

The preliminary findings of this experiment were published in Vasey et al. (2015), which was written for ICD/ITKE research pavilion 2014-15.

4.4.2 Context

The versatility of fibrous composites (Kuo et al. 1988, p. 1004; Hensel et al. 2010, p. 86) allows for the integration of fabrication methods in the early stage of the structural design process. The advantage of a fiber machinery system, such as robotic fiber placement, is that it facilitates this integration by advancing composite fibers to obtain high-performance. The customized composite is associated with parameters, such as “path geometries,” “steering orientation angles,” and “stacking sequences,” in which altering these parameters will significantly change the mechanical and structural properties of the fiber composite (Van Campen et al. 2011, p. 2; Brampton and Kim 2013, p. 1). Developing a method that considers the performative aspects of customized fiber composites requires a number of different mechanisms embedded within a computational tool. The sequential interactions between laid fibers inform the whole composite in terms of the strengths and weaknesses of the system. This awareness coordinates path orientation and the sequence of stacking fibers on previously laid fibers. Consequently, computational techniques for fiber placement are critical to integrating the robotic fiber placement with the performative behaviors of a fiber shell structure.

4.4.3 State-Of-Art

In nature, fiber-forming polymers are limited due to the versatility and metabolically expensive materials; nevertheless, the designing potential of composite fibers is almost limitless (Jeronimidis 2000, 2004, 2008). In aerospace and sailing industries, automated processes of fiber layering enhances fiber composite manufacturing techniques, via a series of methods, such as Resin Transfer Molding (RTM), Automated Tape Laying (ATL), and Automated Fibre Placement (AFP) (Debout et al. 2011, p. 122). In architectural practice, fiber composites are investigated to develop an adaptive composite that integrates structural and functional properties (Doumpioti 2008). In other works, the process of fiber layering derives from the interaction of structural and ornamental properties through a swarming algorithm (Snooks 2012a). This algorithm benefits from informing agents about structural data that is embedded within the design field (Tsiliakos 2012). In addition, the performative criteria of solar radiation can enhance the design performance (Gerber et al. 2014). An Investigation of the performative aspects of fiber layering without a consideration of fabricational movements would rely too heavily on assumptions about the high accuracy of fabrication tools and the homogeneous properties of standardized materials. Moreover, these assumptions would be accompanied by the over-design the composite structure to cover unpredictable nonlinear failures.

4.4.4 Methods

The development of a generative agent-based system relies on individual-based modeling. The controlling mechanisms of this tool consider behavior-based robotic fabrication, which reflects computational agents in physical industrial agents. As a case study, ICD/ITKE research pavilion 2014-15 elaborates the development of this tool with respect to a setup that uses a singular morphospace. Furthermore, this generative agent-based design computation benefits from algorithms and tools that were explored in previous experiments.

4.4.5 Developments

Determining agent properties

Types of agents. In this experiment, individual-based modeling requires focus on one type of agent to provide insight into the effects of individual agents on the environment and the built environments. The agent's effects include trail paths that are generated from the agent's behavior on the environment. The generated trails correlate to material systems and fabrication tools, which are established by individual agents. Individual agents integrate two separate systems that are described by fabricational morphogenesis. Accordingly, agents correspond to the chosen fabrication system, in this case, a single industrial robot that lays synthetic fibers on a formwork. The behavior of the agent, in the virtual world, defines the tool-paths of the robotic end-effector. This coordination between the virtual world and the physical world required an abstraction of the material system, the environment, and the fabrication tool. The agent's behavior is determined by all three criteria. The consideration of these criteria is fostered within a class library of design data. The initialization of this class provides necessary details about environmental, material, and fabricational agencies.

In this experiment, the agent communicates with the built environment, instead of interacting with other agents. The built environment is developed through interactions between the material system with the environment. An agent must be equipped with mechanisms that gather data from the environment, and then coordinate that data within the fabrication process. Thus, individual agents require specific perception mechanisms to exploit the field and to collect data from that field for further explorations (Figure 4.4.1). Moreover, collaborations between agents and environments facilitate agent with less computation to analyze the built environment and to act upon that. In the context of a behavior-based system, the visioning system senses the environmental parameters and identify the observed elements via "IF/THEN" mechanisms. These elements trigger the agent's responses. Accordingly, an agent without a memory system observes the environment and reacts directly to it using a simple recognition mechanism.

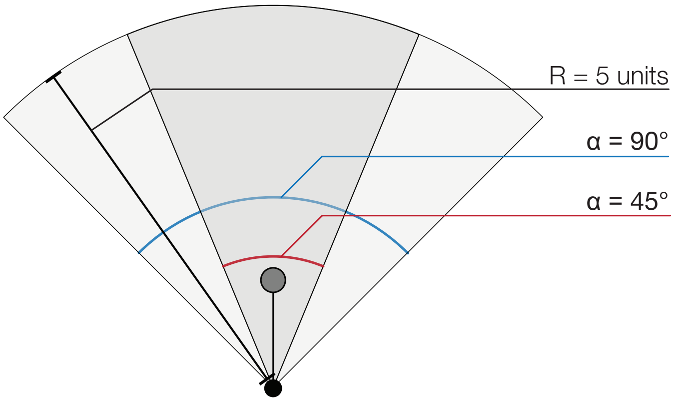


Fig. 4.4.1: A schematic diagram of an agent with a parametrized field vision and a sensory field tentacle.

Geometric properties. The individual agent comprises of a collection of data that is represented with a basic geometry of a vector. This type of agent, without a specific morphology, has the potential to modulate the environment. Additively layering the agent trails on the environment modulates the environment. Therefore, the agent's morphology has negligible effects on the process of pattern formations. In the context of agents with negligible morphology, the emphasis on the interactions among agents shifts to the interactions between agents and their environment. In relation to the structure of the environment, the interactions between an agent and an environment require communications between agent and environment-like agent. Accordingly, the behaviors generated by these communications exhibit a behavioral arrangement that includes fabricational morphogenesis within the environmental system. The manifested structure provides a behavioral consideration of morphodynamics, constructed by an agent with basic morphology.

Sensing properties. Perception mechanisms underlie agent communication structures, at least, in the context of agent communication with the environment. These mechanisms are directly related to conducting environment-like agents. This type of environment provides individual agents access to embedded information for further exploitation. The main factors of the environment as agent are initialized within the class of design data, which include the target surface, external structural elements, and anchoring positions. In addition, during the process of simulation, it provides access to other information, such as laid trail paths. Perceiving the environmental stimuli is delivered via two perception mechanisms: a vision field detector and a sensory fixed tentacle

(Figure 4.4.2). In the context of computational modeling, each of these detectors is developed to probe information that is embedded within the environment.

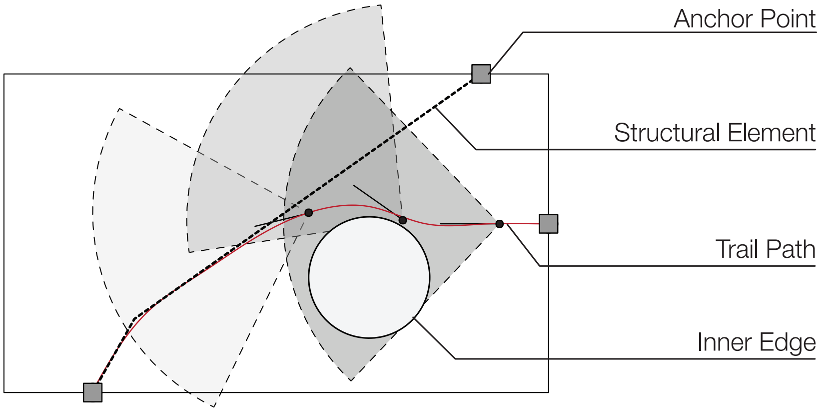


Fig. 4.4.2: The simulation of an agent's perception mechanisms.

The vision field detectors are developed to control the number of patches that an agent can exploit. The agent's displacement vector identifies the vision direction, which also limits the angle and depth of vision. This narrows the field of vision to the specific number of patches. Therefore, limiting the number of patches facilitates communication between individual agents and patch systems. The patch systems collect the information regarding laid trails along with other necessary information, such as the structural and mechanical properties of the environment. Reducing the number of patches controls the amount of interactions among agents, the built environment, and the environment. These limited interactions avoid unnecessary computations. In addition, this mechanism is enhanced at the level of topography and the level of topology. At the topography level, the vision field utilizes a Cartesian distance calculation, and at the topology level, the surface considers mesh connectivity to find related patches in the vision field. Figure 4.4.3 illustrates the combination of these two levels to explore the environment. In this model, the von Neumann neighborhood algorithm calculates the connectivity among patches.

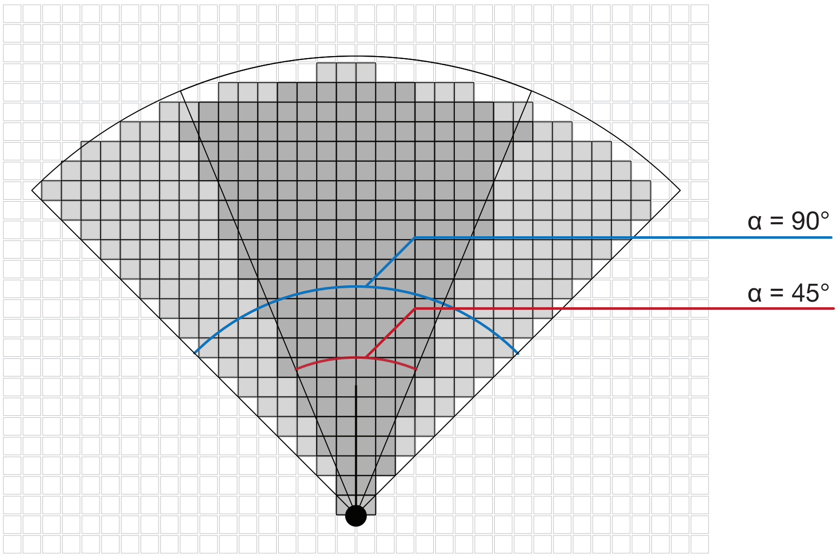


Fig. 4.4.3: Overlapping Topological and Topographical levels to find related patches.

The fixed sensory tentacle is developed to exploit the agent's location on the target surface or the formwork. This mechanism detects any anomalous areas on the formwork, which consist of the inner and outer edges of the surface (Figure 4.4.4). This mechanism maintains the presence of an agent within the environment, where the agent can also benefit from the vision field detector to self-organize its behaviors. This mechanism calculates the intersection between the sensory fixed tentacle and the target surface. The fixed tentacle system is a circular element, where the center is the agent's tangential location to the target surface and the radius is the parameterized value of the agent's step. In addition, this circle is mapped onto the plane geometry. The plane is developed by the tangential location of an agent on the target surface and two vectors. One vector is a normal on the target surface. The other vector is the agent's displacement vector. The collision of this element with external entities, such as laid fibers, activates the related behaviors. The accuracy of this mechanism is parameterized with system tolerances. The agent step size is related to the predefined tolerance value of the system. Additionally, the length of this sensory system can be adjusted to satisfy the required sensitivity for searching the anomalous area within the environment. This mechanism helps the agent locate its next position on the formwork. The agent explores the topography of the environment without any affiliation to the surface topology.

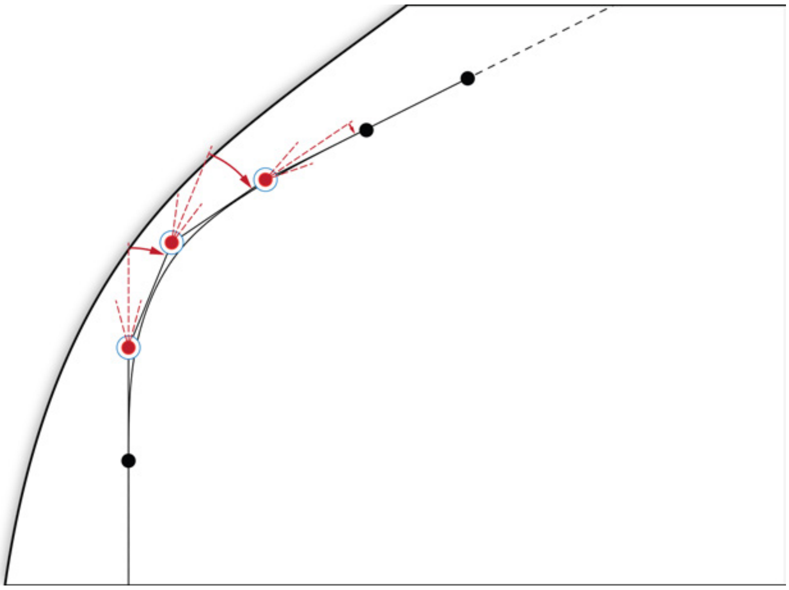


Fig. 4.4.4: A schematic diagram of interaction between a sensory fixed tentacle and the environment anomaly.

Behavioral properties. The behavioral properties of the agent rely on tasks that orient the agent's actions. By accomplishing tasks, the agent is confronted with laying trails on the formwork. The sequence of a behavioral composite is categorized by reinforcing structural and mechanical properties (Figure 4.4.5, left) and following a force field to cover all areas of the formwork (Figure 4.4.5, right). These two tasks describe two layers of actions upon the behaviors of wandering and exploring the surface geometry.

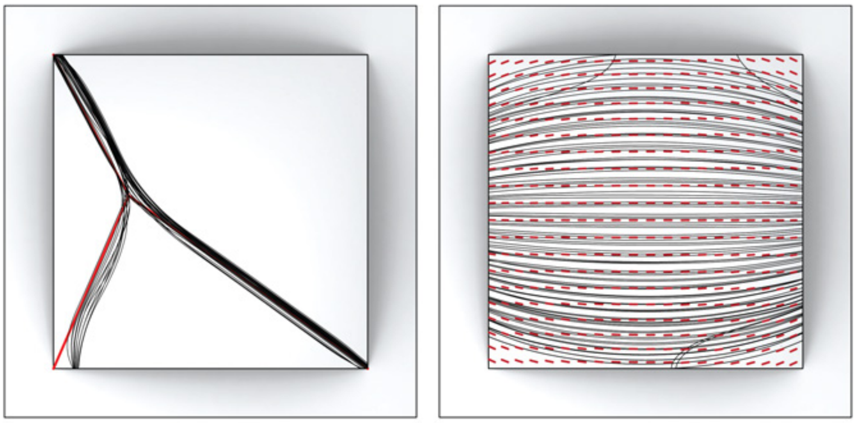


Fig. 4.4.5: Two main behaviors; left image: Reinforcing the structural elements; right image: Covering the surface target.

In the realm of developing inclusive design computation tools, the performative criteria, which are defined through a set of interactions between internal properties and the external environment requires a translation of internal features to external factors. Therefore, analyzing the structural properties of a formwork promotes the externalization of principals of strains and stresses as a set of scalar and vector quantities (Figure 4.4.6). These structural principals explicitly orient the agent's behaviors. The generated trails are aligned with these forces to guide structural loads toward the formwork edges. Gradually, the agent modulates the structural properties of the system by laying trails on the formwork. Therefore, the agent is capable of mediating between performative movements and fabricational morphogenetic principles. Mediating these two inclusive drivers is accompanied by a consideration of environmental effectiveness that includes the topological and geometrical features of the formwork.

Fig. 4.4.6: Embedding structural principals on the formwork. The example of ICD/ITKE research pavilion 2014-15.

Determining a contextual environment or field

In this experiment, the environment is an active field, which considers a manifold as a topological space for an individual agent's exploration. The manifold facilitates communication between an agent and its built environments. This process utilizes stigmergic or sematectonic communication methods to store information about the agent's actions into the topological space. This communication method allows the agent, which has no memory system, to utilize the environment as a memory system (Figure 4.4.7).

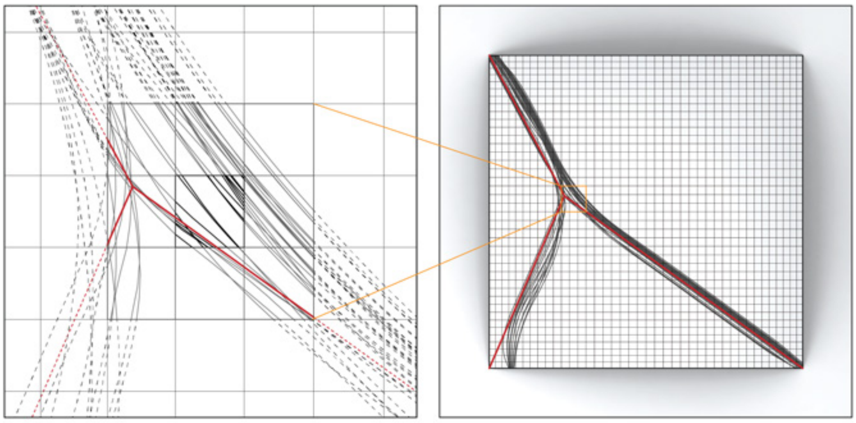


Fig. 4.4.7: The details of a selected patch system with stored data that contains the trails of agents and the external cables.

Patch systems. Developing a memory system within the environment benefits from a topological definition of the field. This process discretizes the field with a topological mesh system, wherein each mesh face is considered a single patch system (Resnick 1994, p. 34). Abstracting patch systems into a static type of agent underlies basic communication mechanisms. At each step, the agent recognizes the value of the patch that it occupies. The patch stores all the locations of an agent as it moves across the patch to produce an agent trail. Each position of an agent on the occupied patch is accompanied with local interactions of the agent with the environment. All the trails in this way are stored in the field. In addition to the patch systems, the generated trails suggest another type of static agent. Each trail demonstrates an agent's behavior. All the previous positions occupied by the agent determine agent's trail (Figure 4.4.8).

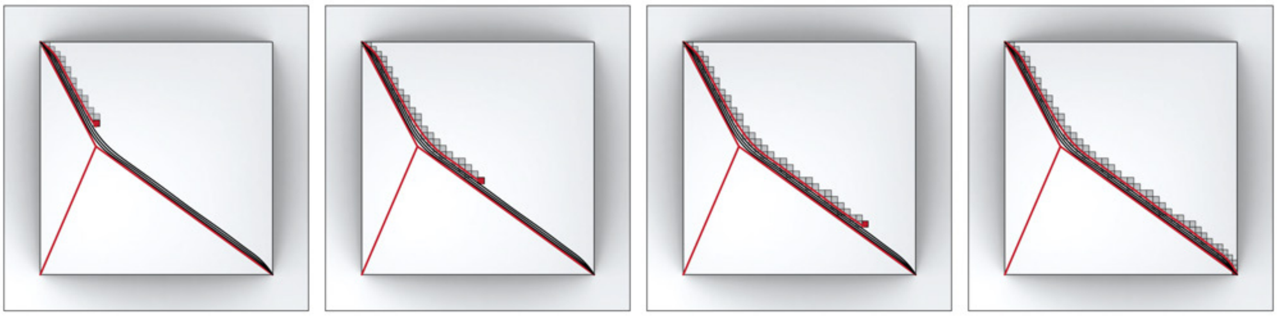


Fig. 4.4.8: Agent-Patch system interactions: The simulation of the agent's behaviors on the environment and the process of storing its behaviors into the environment.

Trail systems. Trails also record all indices of the patches that the agent explores. This process provides further interaction between trail systems and patch systems. This interaction means that the geometrical description of one trail, which manifests as a polyline in this experiment, can dynamically adjust its coordinate points with the patch systems. Even after the generated trails are smoothened, this mechanism updates the relationship between patch systems and trail systems. Both of these systems dynamically update their relationships. This postulates another level of interaction within the environment, through which built environments interact with the patch environments. This process optimizes the embedded signals within the environment, which affects the behavior of the individual agent and regulates its behavior. Figure 4.4.9 represents the process of interactions between trail systems with patch systems. Furthermore, the assemblage of trails demonstrates how local interactions between these two systems affect the global results of individual agent's behaviors.

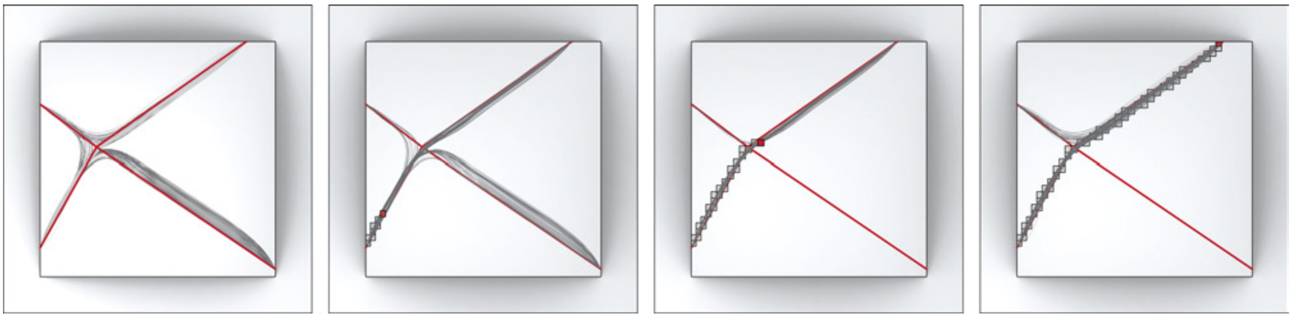


Fig. 4.4.9: Agent's trail system interactions: The simulation of agent's behaviors on the environment and the process of extracting previously stored behaviors from the environment.

Anchor systems. In the context of stacking fibers on the formwork, utilizing anchor points at specific positions, such as the contact positions of external cables and environment edges, enables agents to approach the edge, and then continuously lay trails on the formwork. A gradient patch around each of these anchor points notifies the agents about the anchor points (Figure 4.4.10). The gradient values that are stored within the patch systems provide prioritization values. Agents leave anchor points by applying these values. To put it more concisely, several mechanisms work together to facilitate corrective actions at the anchor points. Perception mechanisms allow the agent to access the anchor point gradients and to prevent edge collision. Tagging mechanisms flag the agent with the anchor point conditions. These mechanisms tell the agent to finish one trail and to commence a new one. After a trail is generated, it is smoothed through post processing.

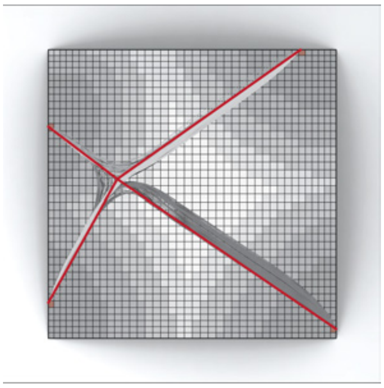


Fig. 4.4.10: A gradient system that informs the agent about the position of the anchor points.

Determining interaction behaviors or rules

In this experiment, the agent is a task-oriented entity with a behavior-based structure that emphasizes the hierarchy of stimuli, processes, and actions. Accomplishing the task relies on the sequential processing of the agent that is determined with a behavioral mechanism to explore the environment. The behavioral method is simplified into a perception mechanism to exploit an environment with sensory mechanisms and act upon pre-determined rules. The pre-determined rules are established to coordinate the behavior of the agent toward gradually realizing the tasks, which are comparable with the process of exhibiting emergent phenomena. From this perspective, achieving the tasks is a complex system that is rooted in simple rules and behaviors. The agent uses perception mechanisms to probe the environment. In addition, "IF/THEN" mechanisms analyze the explored area to trigger appropriate responsive actions, and later flag the agent to provoke a different set of rules (Figure

4.4.11). Provoking the agent to release an action is underlain by predefined rules.

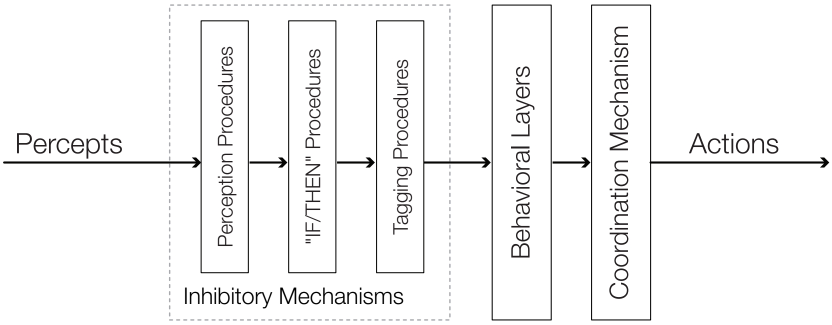


Fig. 4.4.11: The inhibitory mechanisms with three procedures: Perception, "IF/THEN," and Tagging.

Activating predefined rules requires several "IF/THEN" mechanisms to categorize the explored situations. The evaluation mechanisms rely on the tasks that define the agent's behavior. Each task emphasizes a specific set of "IF/THEN" mechanisms. Aggregating all of the agents' behaviors triggers certain responses that manifest the next states of the agents. These states should be adjusted to the predefined rules for determining appropriate actions between all generated response actions. This selective process prioritizes actions through a set of rules that determine the appropriate actions. Accordingly, this process utilizes coordination mechanisms to determine the next state of the agent.

Determining the inhibitory mechanism

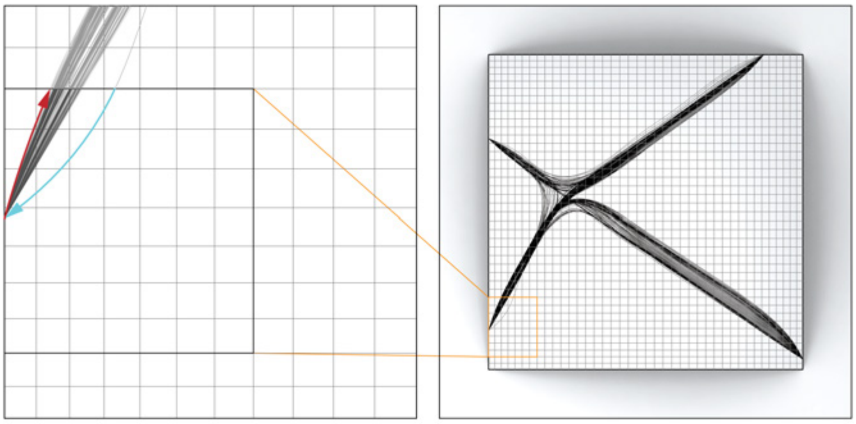


Fig. 4.4.12: The details of a tagging mechanism with two steps: Approaching and leaving anchor points.

Developing an inhibitory mechanism enables the behavioral system to adapt agents to unpredictable situations, for example, avoiding the edges, clustering at the edges, and anchoring at the predefined positions. The tagging mechanism is utilized to inform agents that are approaching the edges or anchors or leaving them altogether (Figure 4.4.12). In addition, the inhibitory mechanism provides smooth transition from one set of behaviors to another. This process follows the development of a parameterizing system that parameterizes agent behavior and the formwork. Figure 4.4.13 schematically illustrates the relation between fabrication tools and the formwork. The development of a machinic morphospace that indirectly determines the design data setup relies on fabrication tools that lay fibers on the formwork to develop a composite shell. The correlation between the robotic fabrication tool and the formwork determines the possible design solution space. In relation to fabrication tools, it is required to study robotic workspace. The workspace of the industrial robot determines the reachability of end-effectors, which provides certain limitations for designing a formwork. For example, if the designed formwork falls outside the workspace, it will not be fabricable. Individual agents prevent the generation robotic tool-paths in these areas. Additionally, the axis limitations of the robotic setup need further consideration, for which computational agents only generate applicable traces.

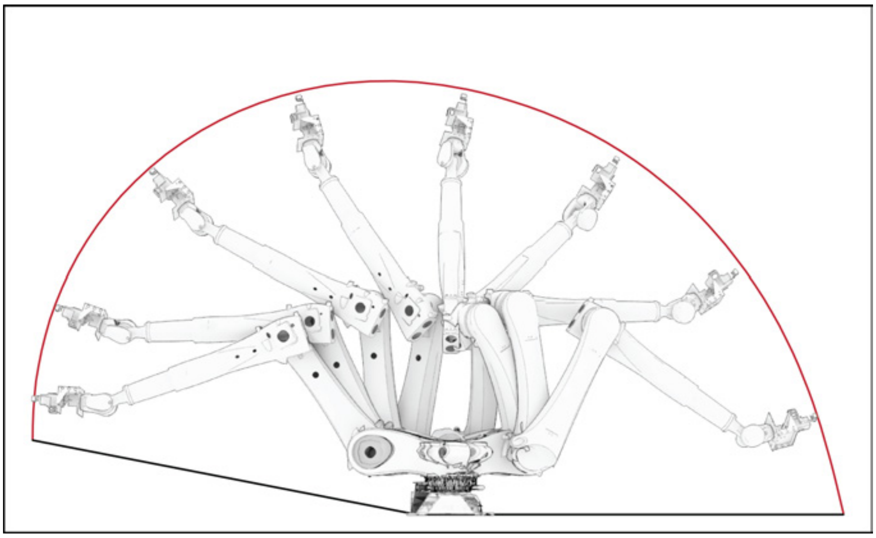


Fig. 4.4.13: The interaction between robot end-effectors and the formwork.

Determining the coordination mechanism

Aggregating the behaviors of the agent requires a mechanism to integrate all responses triggered by environmental factors. This mechanism prioritizes different action layers by ignoring one layer and emphasizing the others. This prioritization advances the agent's behaviors by selecting related responses to perform particular tasks. In *Behavior-Based Robotic* (Arkin 1998), responses to different environmental stimuli are assembled through notational formalisms and other methods of coordinating behaviors (Arkin 1998, pp. 104-119). These methods are accompanied by subsumption-based robotics that include several layers of task-achieving behaviors built upon a set of primitive behaviors (Pfeifer and Scheier 2001, pp. 201-206; Matarić and Michaud 2008). In accordance to notational formalisms developed by Arkin (1998), each layer of behavior at a given time t is noted to consider stimuli S, behavioral response $R = B(S)$, and the relative strength G in which the notational mechanism is defined through the coordination function C of:

$$\rho = C(G \times B(S)) \text{ or } \rho = C(G \times R) \tag{4.1}$$

, where:

$$R = \begin{bmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ r_n \end{bmatrix}, \quad S = \begin{bmatrix} s_1 \\ s_2 \\ \cdot \\ \cdot \\ \cdot \\ s_n \end{bmatrix}, \quad G = \begin{bmatrix} g_1 \\ g_2 \\ \cdot \\ \cdot \\ \cdot \\ g_n \end{bmatrix}, \text{ and } B = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \cdot \\ \beta_n \end{bmatrix}$$

(Arkin 1998, pp. 108-111).

The generated coordinative behaviors are combined through comparative and competitive methods (Arkin 1998, pp. 111-116). In this experiment, the summation of values as a steering mechanism (Reynolds 1987, 1999) is employed to release selective actions.

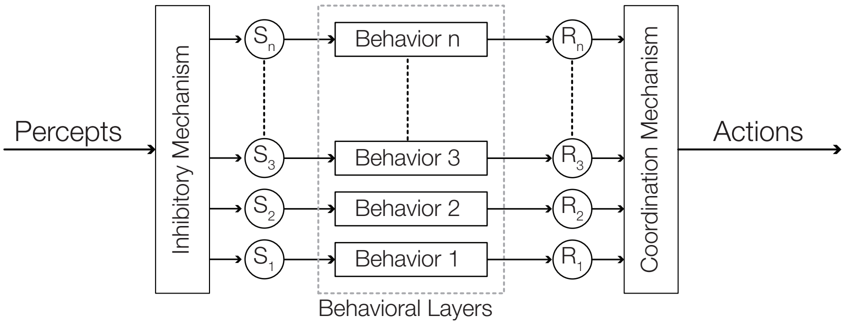


Fig. 4.4.14: The schematic illustration of behavioral layers with their stimuli S and responses R.

A coordination mechanism is developed to follow the two main tasks of this experiment, reinforcing the structural and mechanical properties of the formwork and covering the empty area of the field. The formwork is reinforced by strengthening of the naked edges. Each of these tasks is layered separately on top of a set of primitive behaviors. The primitive behaviors and sophisticated behaviors are described as locomotion, wander, follow external structures, follow generated trails, follow scalar and vector fields, avoid open areas, avoid edge areas, and approaching and leaving the edges. Figure 4.4.14 schematically illustrates the combination of primitive and sophisticated behaviors within behavioral layers. Each behavior requires a stimulus to activate a singular response. All of these behaviors are noted and converted to the vector-based system. This notation facilitates the task-achieving behaviors, amplified with strength values. These values can generate gradient systems informed by the patch systems. The behavior of agent dynamically adjusts and weights with the embedded scalar values. Additionally, parameterizing the strength values establishes a mechanism for users to intensify specific layers or responses, while the users are capable of ignoring several behaviors and focusing on a single behavior. Figure 4.4.15 shows the classifier actions that weigh the responses *S* with the relative strengths of *G* to intensify or weaken a specific behavior. Accordingly, the agent has the potential to behave as an autonomous agent or a semi-autonomous agent. The classifier actions determine the applicable range of strength values for coordinating and combining agent's behavioral layers.

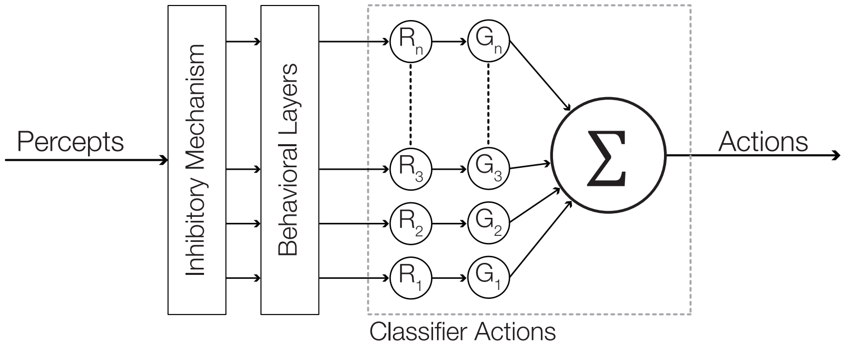


Fig. 4.4.15: The schematic illustration of a coordination mechanism, including the classifier responses *R* with relative strengths *G*.

4.4.6 Results and example of implementation

ICD/ITKE research pavilion 2014-15

Fig. 4.4.16: The research pavilion 2014-15; source: ICD/ITKE University of Stuttgart.

Fig. 4.4.17: The biological role-model, a water spider; source: ICD/ITKE University of Stuttgart.

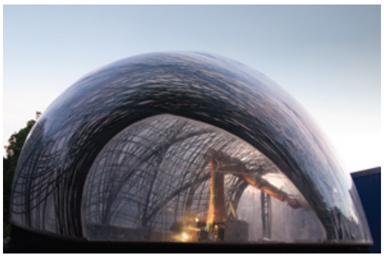


Fig. 4.4.18: The relation between the industrial robot and the composite shell; source: ICD/ITKE University of Stuttgart.

The ICD/ITKE research pavilion 2014-15 (Vasey et al. 2015) was an example of taking inspiration from biological role model, such as a water spider, to develop fabricator agents that modulate the environment. Fabricator agents, in interaction with the environment, erect morphological patterns. Morphological patterns accumulate local patterns, and then emerge as a global morphology. The global morphology, which is a modulation of the environment, determines the built environment of the individual agent. These aspects were translated into the ICD/ITKE research pavilion 2014-15, while respecting the biological principles of the water spider (Figure 4.4.17).

The fabricator agent was required to lay fibers on the inflated membrane, which functioned as the environment. The environment was an inflated Ethylene tetrafluoroethylene (ETFE) membrane, wherein the fabricator agent included a Kuka KR 120 R3900 industrial robot, which was situated at the center of the membrane. These two aspects defined the possible solution space for the agents to execute allocation tasks, such as design intentions, fabrication constraints, material behaviors, and structural and mechanical performances (Figure 4.4.19). Executing these tasks resulted in a layering of fibers on the shell structure. The overlay of agent's traces on the membrane produced a unique emergent fiber layout (Figure 4.4.18).

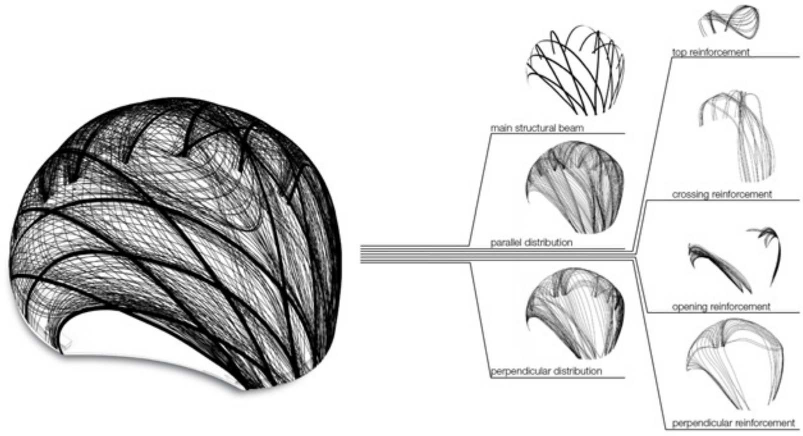


Fig. 4.4.19: Overlapping behavioral layers on the final fiber layers; source: ICD/ITKE University of Stuttgart.

The study on fabrication constraints that was done by ICD/ITKE researchers provided insight about the reachability of the robotic setup (Figure 4.4.20). Overlapping robotic workspaces with the membrane shell developed a hyper-dimensional morphospace, which directly influenced the agent's behavior for generating robotic tool-paths. The behavior of the computation agent reflected in the robot end-effector, which determined the kinematics of industrial agents. The robot fabrication setup imposed two constraints on Axis 1 and Axis 6. The angle rotation around Axis 1 limited the locomotion behavior of the computational agent to 185 degrees. This means that the robot can generate trails between two anchor points, which are positioned relative to the maximum 185 degrees. Axis 6 limits the reachability of the robot to the membrane, in which the robot workspace forces the membrane to fit within a specific volume. Figure 4.4.21 illustrates these limitations for the fabrication setup.

Fig. 4.4.20: The process of laying fibers on the membrane.

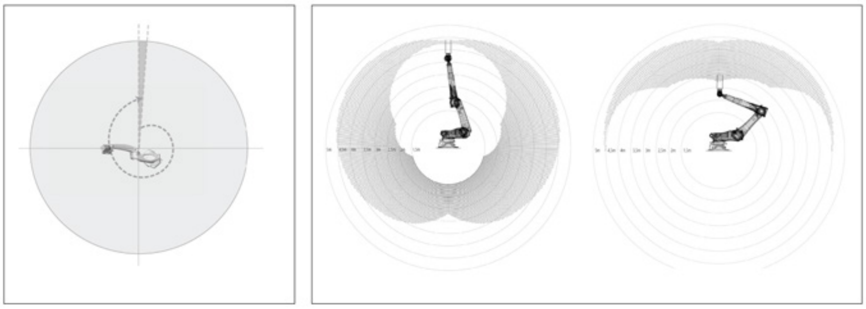


Fig. 4.4.21: The fabrication constraints; left image: The robot Axis 1 limitation to 185 degrees; right images: The robot Axis 6 limitations; source: ICD/ITKE University of Stuttgart.

In the research pavilion 2014-15, the industrial agent, as a physical agent, was associated with different behavioral factors, such as the ethological construction of the biological role model (biomimetic), the structural and mechanical properties of the membrane and stacked fibers (material), and imposed constraints of robotic setup (fabrication). Translating these behaviors into a set of action layers provided through embedded information in the environment, which was indicated as stimuli to trigger action behaviors (Figure 4.4.22).

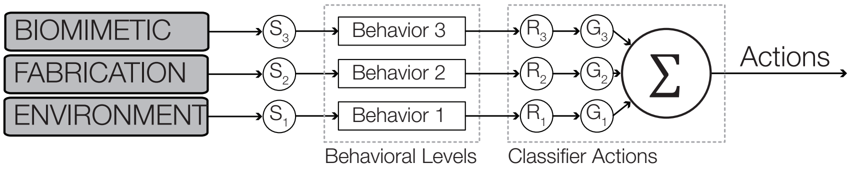


Fig. 4.4.22: Behavioral categorization, including biomimetic, fabrication, and environment.

In addition to overlapping the environment with the robotic workspace, the environment was converted to discrete units as patch systems that utilized procedures of dynamically preserving information. The patch systems were capable of storing different kinds of information, such as geometrical and numerical data. However, in the context of this study, the patch systems were customized for storing mechanical and structural properties, constraints of the robot setup, opening areas, and a number of laid fibers on each patch. Additionally, the environment discretization was associated with the mesh geometry, where the discrete units were related to mesh faces. Due to the mesh geometry, each patch provided supports for utilizing geometrical information, such as the normal vectors at the centroid of patch.

Accessing these properties via a vision field triggered the agent's actions through: following the stress principal directions; perceiving the opening areas and projected robotic constraints; approaching and leaving the anchor points located at the pavilion contacting area with the ground; and adjusting distances to the laid fibers (Figure 4.4.23). Prioritizing these layers of actions indirectly navigated the agent's behaviors, for example, increasing the density of fiber layout at the pavilion opening. Additionally, this technique enhanced the structural properties of the composite shell by increasing the activity of agents on the high stress areas. The notational formalisms of these behaviors provided the high level of negotiations between the predefined limitations and the user intentions. Consequently, the user could activate one behavior and change the magnitude of that behavior, while the other behaviors could act upon predefined values and rules.

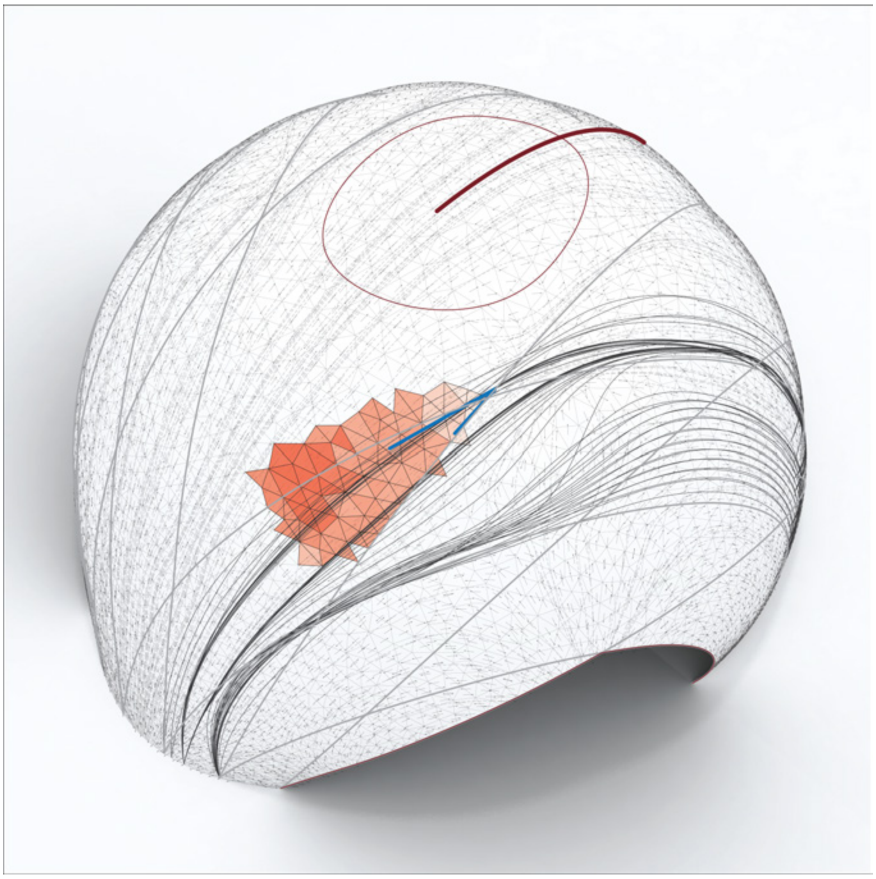


Fig. 4.4.23: Over-layering fibers in response to different behavioral layers; source: ICD/ITKE University of Stuttgart.

4.4.7 Discussion

The stacking sequences defined the fiber placement as an additive fabrication process. The new sequence of fiber path generation will be informed by the previous fiber layouts. The process of informing the system sequentially changes the system behaviors, not only the current fiber path behavior, but also the next path generation. This dynamical system relies on the method of encoding the generated path data into the manifold. The manifold contains stimuli field, which dynamically changes by storing the generated fiber path at each iteration. For example, the path generator tool changes the structural behavior, when it laid one fiber path against the shell structure. The stimuli field triggers the behavior of the agent in which the fiber path-generating agent selects a suitable action to execute an appropriate behavior in response to the induced stimuli. The bilateral relation between the path-generating agent and the field dynamically changes the overall state of the system in which the agent's behaviors simultaneously modulate the field as the field modulates the agent's behaviors. Finally, the adaptation of fiber layout emerges during the process not after the process.

4.4.8 Acknowledgments

The computational design team of the ICD/ITKE research pavilion 2014-15 included Ehsan Baharlou, Lauren Vasey, and Kenryo Takahashi.

The role of author in this project included the development of a generative agent-based tool that was used to generate robotic toolpaths for laying fibers on the formwork.

5 Towards Generative Agent-Based Architectural Design Computation

5.1 Preamble to Generative Agent-Based Design Computation

Model relations

Introduction to utilizing a formal model. Abstracting the physical aspects of materials, such as their geometrical definitions or their fabricational dependencies, eases the transition from natural system to formal system. In the context of architectural design, formalizing the semantic contents of forms to computational syntax relies on mathematical and geometrical descriptions of forms, where mathematical structures of form are imposed on the materials and fabrication systems. In contrast, these mathematical structures can arise from morphogenetic movements. Employing computational applications, such as CAD[1], CAE[2], and CAM[3] assist designers to find flaws before materializing developed forms. The linear nature of this integrated process emphasizes the independent sectors of building industries in which each section, according to their needs, develops specific tools and applications for routinizing design tasks.

Relation between the real world and the computational model. Blurring the lines between design, engineering, and construction processes underlies the formalization of a computational model, which consists of independent and interconnected units. Each individual unit is responsible for adapting the design's emergence with the constraints of material and fabrication systems. The virtual environment reflects the real model of construction at time t, which enables the micro units to model the process of making at time $t+1$. The accompaniment of this transition considers the effectiveness of

---

[1] Computer-Aided Design.

[2] Computer-Aided Engineering.

[3] Computer-Aided Manufacturing

environments and limitations within available fabrication tools and materials. Figure 5.1.1 schematically illustrates the formal model of this transition from inclusive agencies, such as fabricational morphogenesis, performative criteria, and environmental effects. The development of "model relations" enables individual units with a general awareness of construction processes to integrate abstracted principles of materials and fabrication systems. The result of this integration at time $t + 1$ must be consistent with the real construction processes to identify the effective factors of construction failure.

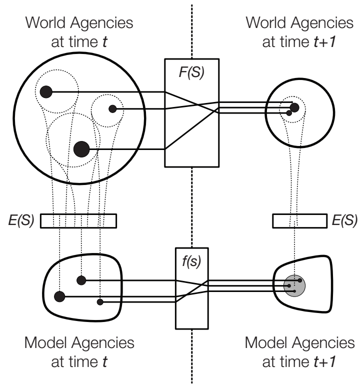


Fig. 5.1.1: Formal model of inclusive agencies, where $F(s)$ defines the transition laws, $f(s)$ is transition algorithms, and $E(S)$ is the observation for abstraction procedures. The development of this diagram is adjusted to the models developed by Holland (2000); Miller and Page (2007).

Transition function. A formal model of integrating material behaviors and robotic fabrication systems includes both real and virtual stages. In real world, transition from the state of fabrication setup at time t to time $t + 1$ is associated with physical laws from which the interactions between materials and fabrication tools at each state can be followed by determined setups. These setups rely on simple geometrical properties in relation to the fabrication tools and techniques, which are utilized within the manufacturing processes. However, in virtual environments, the transition of computational setups at time t to time $t + 1$ is based on translation and the interpretation of physical laws by mathematical algorithms, such as motion or kinematic equations, which provide a virtual simulation of fabrication processes.

In the context of virtual manufacturing, CAM applications enable modeling fabrication setups in which the examination of the fabrication processes helps to recognize the limitations of fabrication procedures. In the realm of robotic fabrication, monitoring the accessibility of the last axis, such as Axis 6 within industrial robots with six axes, particularly determine the fabrication space. This robotic workspace is a reachable space for end-effectors, but even within these space–customized fabrication processes confront some axis limitations. For example, in the ICD/ITKE research pavilion 2014-15, evaluating Axis 6 through random distributed points on the pneumatic formwork determined accessible areas for specific inverse kinematic (IK) solver. Inaccessible areas on the formwork determined the repulsion zones, so the agents consider those areas during the process of virtually determining robotic tool-paths.

The limitations that arise from the interactions between material behaviors and fabrication processes determines the rules and procedures of fabricational morphogenesis. Implementing these constraints within design processes necessitates analytical methods to highlight the producible areas of solution space. Accordingly, the analytical theoretical morphospace facilitates the design space with two areas of producible and improducible form. The dynamic modeling of fabrication processes requires "transition functions" as generative mechanisms to relate producible area in time t and time $t+1$. The transition function includes mechanisms to maintain proper modulation between these two related states (Figure 5.1.2). Figure 5.1.2 represents the two transition functions of $f_1(s)$ and $f_2(s)$, where $f_1(s)$ successfully transits the model towards the producible zone (the dashed area) but $f_2(s)$ transits the model to the non-producible zone.

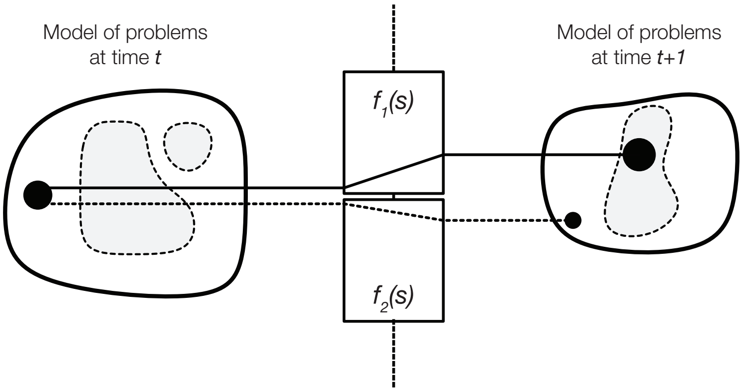


Fig. 5.1.2: The transition model of two generative mechanisms of $f_1(s)$ and $f_2(s)$.

Equivalence classes or observation. The relationship between the real and the virtual world significantly depends on determining the maximum capacities and the effective properties of fabrication setups. Mapping these aspects from real fabrication processes to the virtual setup relies on developing theoretical morphospaces. In theoretical morphospaces, developing geometrical and functional models of morphology is extensible to developing equivalence classes of observation that map the geometrical definitions of forms with their material behaviors and fabrication procedures onto the hyper-dimensional morphospace. At each state, the producible areas in the real world is correspondent with the hyper-dimensional morphospace through which the application of fabrication procedures to virtual materials generates forms that exists within the range of the producible areas at the next state. Figure 5.1.3 schematically represents the mapping process of theoretical morphospaces between real and virtual setups, in which the proper generative algorithm transits the fabrication process at time t to the producible areas at time $t+1$. Neglecting effective factors within the process of mapping the real world to the virtual construction setup generates fundamental flaws in the morphospaces. Accordingly, any faults in developing morphospaces lead a simulated design towards the improducible area of real fabrication setups.

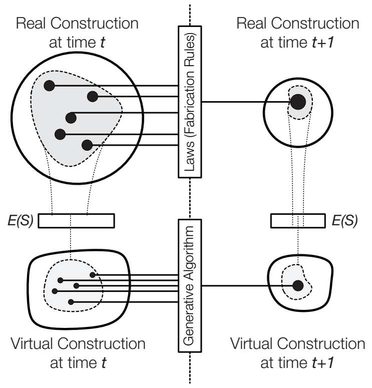


Fig. 5.1.3: The formal model of fabrication. Dashed areas indicate the hyper-dimensional morphospaces. Dotted lines represent the equivalence classes, which map the observation of real fabrication setups.

Abstract models

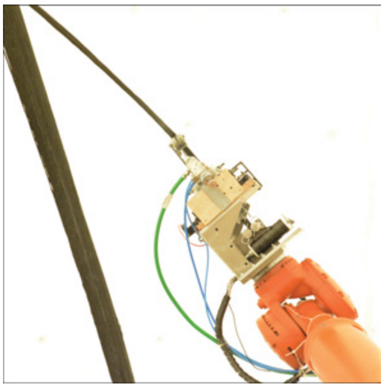


Fig. 5.1.4: The customized end-effector for layering fibers on the membrane; source: ICD/ITKE University of Stuttgart.

A generative agent-based system represents the integration of basic principles and features, which exist within materials and robotic fabrication tools. Mapping these principles onto computational frameworks is a process of developing an "abstract model," a model that only considers a certain part of fabrication processes. Neglecting the physical and mechanical properties of a material system enables designers to focus on the geometrical behaviors of a material system. The geometric definition of raw building components has a direct relation to the manufacturing settings. Utilizing customized fabrication methods, such as different types of end-effectors (Figure 5.1.4), provides a vast range of fabrication settings. This range of settings will enhance insight into the agency of fabrication within the abstract model. Therefore, empirical experiences of working with the specific fabrication setup provides a better understanding about the effectiveness of fabrication parameters. These mathematical parameters specify the constraints and capabilities of fabrication settings, which have decisive roles in agents' behaviors.

The abstraction level of these two agencies determines the accuracy in modeling behaviors of integrative fabrication processes. High-level and low-level of abstractions indicate the proximity of the abstracted model to the real fabrication setup from which a certain level of abstraction is required to avoid oversimplifications and complexifications. Abstracting mathematical and geometrical definitions into a generative algorithm supports the investigation of a formal model that integrates material and fabrication processes. The generative models that were experimented in this study only cover a small part of the construction industry. This study is limited to robotic fabrication and materials, such as plywood and fibers. Developing inclusive models that integrate manufacturing conditions and constraints into design principles provide a generative system with a cognitive knowledge that is experienced within fabrication procedures. On a certain level, it might be possible to validate the output of this abstracted model with the existent empirical data of tools and materials. In some cases, the validation of the model only relies on similarities to real construction conditions.

Agent's agencies

Developing a formal model that is mostly a homomorphism map between an abstracted model and the process of construction requires an agent's agency to interplay between the agencies of material and fabrication. An agent's agencies rely on structures that differentiate the level of interactions with environments to accomplish the task. In one case, agents require external information to achieve their desired goals. While in the other, agents are aware of the way of solving tasks. This awareness reflects the level of the agents' ontology that is embedded within the agents' data-structure. The term ontology refers to the comparison between "Classical AI" and "Modern AI," where developing a robotic agent requires the specification of the level of primitive knowledge in interaction to the real world (Pfeifer and Scheier 2001, p. 117). The low-level of ontology requires that agents yield the necessary knowledge from the environment through a series of negotiation and communication with external systems. The high-level of ontology shifts the focus of agents from exploiting environments to searching within its vast embedded knowledge. Embedding or situating abstracted knowledge about material and fabrication constraints in the agents' ontology fosters agents' behaviors with the limitations of material and fabrication agencies. These processes follow by developing mechanisms and procedures to narrow the generated outputs down to the range of possibilities that are producible with the construction setup. Eventually, this process fosters the definition of agents' morphology and their associated behavioral systems.

5.2 Developing Agent-Based Models in Integrative Design

Purpose of models

The development of an explicit model requires clear problems to find useful solutions. The purpose of modeling an agent's agencies is integrating fabrication and material systems within design processes from the beginning. Consequently, the goal of active agencies is to shift conventional knowledge-based method of design integrations to find behavioral methods. The behavioral methods are consistent with interplaying agents that are embedded with the ontology of construction processes. This process transits integrating processes from a high-level of knowledge about construction processes to a low-level of behaviors from which agents gain an awareness of the abstracted principles of construction and materials. Fusing these abstracted agencies onto the basic elements of production, like building materials and fabrication processes, leads the integration procedures towards constructible components. Accordingly, searching for common features of materializing design computation enables the agencies of material and fabrication systems. This generalization is proportional to the purpose of modeling from which it determines the level of amalgamation among inclusive drivers.

The relationships between micro-levels and macro-levels

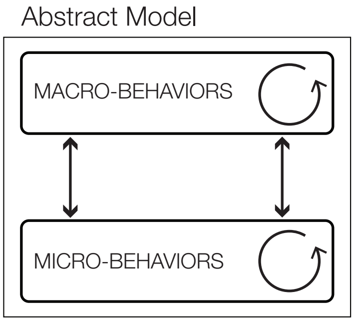


Fig. 5.2.1: The schematic illustration of micro-macro effects with the process of self-organization. Adapted from De Wolf and Holvoet (2005).

Approaching the main purpose of modeling is associated with micro and macro perspectives that determine a general system with subsystems and their interrelations. Figure 5.2.1 shows the relation between micro-behaviors and macro behaviors, where each level considers self-organization processes to demonstrate emergent properties. Interrelating subsystems at the micro-level advance a system with interstitial interactions to produce appropriate outputs. These outputs mostly result in a macro-level, which is in a higher-level than their own generator subsystems. In comparison with design processes, the global design is aligned with a macro-level of regularities that are merged from a local modulation of elements and components. Interplaying at the local level combines micro-behaviors to adjust (regulate) the intended macro-level of global design. Accordingly, there is direct relationship between micro-behaviors and specific macro-regularities from which the global design is divided into local behaviors, instead of generating the global design from decomposed global knowledge.

Coalescing local behaviors might generate behaviors that overarch the global criteria. In behavioral strategies, the macro-regularities form the basic principles for developing rules and procedures. Translating these rules into micro tasks advances the model to distribute the purpose of modeling to the constituents' units. Therefore, the rule-based units are oriented toward the micro tasks, where completing these tasks enables the model to obtain a general state adapted to desired orders. Desired orders, which exist within a cloud of answers, describes the dynamic states that are in balance with the global design.

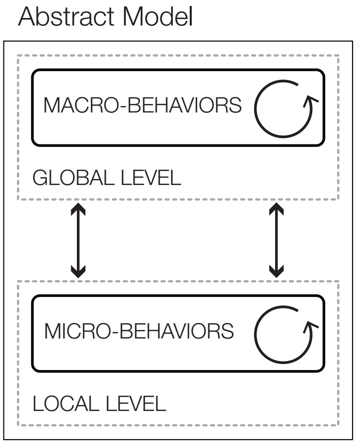


Fig. 5.2.2: The correlations between local-global levels and micro-macro effects.

The relationships between self-organization and emergence

The cloud of solutions considers multiple answers that are close to the regularities that designers intend to achieve these orders within their design. The multi-dimensionality of this space provides the possibility for generative models narrowing the scope of the searching space by considering different criteria. As a result, the generated outcomes must exist within the cloud of solution, including the purpose of the model; otherwise, there is a misinterpretation of the purpose to the elements' behaviors. Interrelating the macro interpretation of the purpose to the micro-behaviors of the elements underlies an understanding about the rules that organize the emergence of the model. Cross relation between micro-behaviors in social science and self-organization in biological systems fosters the design process to describe local interactions among elements under abstracted manufacturing rules, such as laying fibers on the formwork. The abstracted rules at local levels retain the generation of satisfactory states for all elements, while the overall state of the model is far from representing the emergent properties. However, correlating the emergence of global design to the macro-regularities enables a behavioral framework that allows the micro-behaviors not only to complete their tasks but also to self-organize the model to exhibit emergent properties (Figure 5.2.2). This process provides additional intervention to increase regularities within the system, while the elements self-organize their behaviors to generate solutions within the multi-dimensional space. Accordingly, interpreting the global design as the purpose of model requires relating micro-level of self-organization to the macro-level of regularities.

5.3 Developing Agent-Based Systems: Mapping Active Agencies into Agent-Based Systems

5.3.1 Defining agents: Types and attributes

Agent types

Characterizing computational agents with particular behaviors narrows the agents' type down to procedures underlying the morphology of design. In nature, the ethological morphogenesis defines the level of insects' involvement within the development of the natural morphology. In one type, insects apply their bodies to construct temporary structures, such as boats or bridges, in which converting their bodies to smart materials allows individual insects to aggregate into one living structure (Bonabeau 1997, pp. 193–194). The dynamic structure that emerges from the interrelations among insects consists of a distributed network of individual elements. The distributed elements have a global goal while they locally self-organize via their assembly knowledge. For example, in a simulation of assembly agents, such as the process of developing ant bridges, provides procedures for agents to locally evaluate their stability. This process regulates the whole system. The evaluation mechanisms with the agents' morphology feed the outcomes back to the individual elements to attain appropriate states towards exhibiting the global goal.

Insects with knowledge about their degree of competencies obtain their assembly sequences by interacting locally with other individuals. Aggregating the competences of insects together fosters the insects' community with a collective consciousness that will help the insects reach communal goals. In this sense, agents with specific types of morphologies require their structures to adapt to changing conditions. Maintaining the basis of their morphologies affect the implementation of a global goal. Therefore, the morphology of agents is significant in defining the agents' behaviors. In a comparative equivalence to integral design materialization, the correlation between an agent's morphology and an agent's behavior fosters the importance of material systems in fabrication processes. In plate-like agents, the material systems of each plywood plate determines the morphology of the agents, while their interactions under the influence of a geometrical definition interplays with fabrication processes. The cross relation between fabrication tools and utilized materials manifest the form of each agent. Eventually the aggregation of all plate-like agents accomplishes the global goal of panelizing a target surface.

In the other type, insects employ a natural material to build their own nest, such as termite mounds and ants nests, in which the insects use their bodies as patterns to check the constructability of a nest (Theraulaz 2014, p. 59). Modulating local materials are associated with insects' capabilities in utilizing their bodies as smart fabrication tools that are self-aware of limitations and capacities. The awareness of potencies indicates constraints that insects confront to carry, shape, and assemble each piece of their nests. Interacting insects with available materials exhibit behavioral developments of nests' morphology that follow global criteria. For instance, the fibrous-like agent experiments emphasize this process where the motion of an individual agent on the formwork determines the robotic tool-paths. Layering fibers through the industrial robot arm motions conceptually is the imitation of the spider silk spinning. However, the accessibility of end-effectors to the formwork determines the potential areas for agents' behaviors. Avoiding inaccessible areas changes the agents' behaviors. The development of morphology follows these limitations. Similar to the insects that utilize their bodies as main patterns for developing their nests, the workspace of robots indicates one dimension of the theoretical morphospace that defines the possible morphology for robotic tool-paths. The amalgamation of all of these tool-paths fosters the fibrous morphology of the ultimate composite.

The agents' participation in both types of morphological movements establishes ethological or behavioral strategies that arise from involving individual insects with a collective consciousness to build behavioral structures. The ubiquitous insects have a great significance in their built environments. According to their conditions, insects consciously change their roles on establishing the structural morphology. Extending the unique behaviors of insects into the development of computational agents furthers the integration of material and fabrication behaviors into the structure of agents.

Inclusive design computation: Fabricational morphogenesis

Investigating the built environments of insects fosters the significance of agents in adapting the process of construction via mediating between material and fabrication agencies (Figure 5.3.1). Further study on the development of morphology entrenches the importance of constructional procedures by interrelating the fabricational morphogenetic principles with functional and environmental effectiveness. Conceptually, considering these factors in the design process determines the concept of inclusive design. The fabricational morphogenesis includes both materials and fabrication agencies. Establishing these two agencies fosters the basis of behavioral constructions. In the context of manufacturing processes, agents' agencies are developed into two categories of materials and fabrication tools. Agents that correspond to these agencies change their focus from the geometrical definitions of material systems to manufacturing procedures.

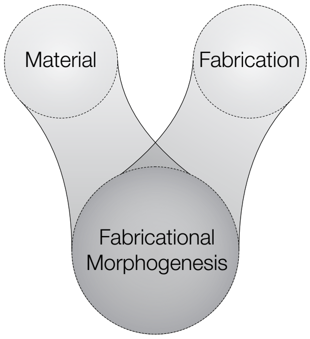


Fig. 5.3.1: The schematic illustration of the fabricational morphogenesis that includes material agencies and fabrication agencies.

Negotiating between these two separated systems, as it is predicted within the fabricational morphogenesis, begins with developing theoretical morphospaces to coordinate the agents' behaviors indirectly. Hyper-dimensional morphospaces indicate analytical spaces that examine the constructability of the simulated morphogenesis from which the theoretical morphospace might determine courses of actions for agents. Intrinsic and extrinsic constraints within a theoretical morphospace are effective in structuring two types of agents. One type emphasizes the geometrical principles of material systems and the other determines the effectiveness of manufacturing systems.

Fabricational morphogenesis: Material agencies

Agents that are imputed to consider geometrical principles consist of a specific morphology with behavioral attributes. Programming agents to consider geometrical principles initiates the agents' morphology from which the agents' behaviors are mostly inherited. In addition to the primitive behaviors, such as exploring the environment with motion behaviors, computational agents with a morphological body have behavioral layers that are relevant to the geometrical definitions. The class of geometrical behaviors determines the interaction among agents. Differentiating the morphology of agents dynamically produces new bodies. Moreover, the geometrical behaviors are associated with internal mechanisms that maintain the agents' body within the morphological tolerances. Accordingly, any deviation from the defined tolerances informs the internal mechanisms to tend towards the morphological stable states, or homeostasis state of agents.

Fabricational morphogenesis: Fabrication agencies

On the other type of agent, the manufacturing process indicates the behaviors of agents, which are inseparable form defining agents' body. Embedding fabrication behaviors in agents' structures directly relies on the different types of available fabrication tools. In the context of robotic fabrications, the mounted industrial robots, i.e., KUKA KR125/2 robot, provides necessary information about the work space, the axes limitations, and the pay loads of the robots. Numerically controlling the robot's end-effector establishes specific performances within the fabrication constraints. Generating codes that enables robots' movements indicate the rules and regularities of modeling a computational agent. Transferring the agency of this process to the computational agents virtually merges the tool-path generating of robotic industries with the simulating agents. In the context of fabrication agencies, agents might reduce the geometrical definition of the morphology to the behavioral actions of agents. Neglecting the morphology of the agents corresponds to the fact that the physical shape of end-effectors is a customized setup suitable for fabrication processes. However, colliding or impacting the effector tool with its built environment necessitates a consideration of the physical shape of tool as the agent's body from which the agent is informed with its limitations. This consideration is similar to the process that occurs in nature in which animals or insects involve morphologies of their body as comparative factors within the process of constructing their nests.

The intelligent behaviors of agents rely on dynamic interactions with external entities that include other agents, the built environment, and the initial environment. Agents' awareness of the external entities are associated with the perception mechanisms that observe the surrounding environment. The process of observation and action is compatible with the agent's tasks, which correlates the agents to their milieu. In the context of computational agents, awareness of the external entities requires a consideration of both the agents' morphological definitions and perception mechanisms; the former is suitable for static environments, such as target surfaces, where the surface provides necessary geometrical knowledge to limit the agents' actions by defining "IF/THEN" behavioral mechanisms. The latter is capable of coupling with dynamic environments, which detect any change to the environments and the built environments through perception mechanisms. Computational agents with perception mechanisms technically advance agents with procedures to gain knowledge of the external environments through sensory mechanisms. Therefore, determining the morphology of agents relies on the specification of tasks that are required to be fulfilled within the environment.

5.3.2 Defining environments: Types and significances

The significance of the environment

Situating agents within the environment enables them to explore the environment to obtain necessary information for solving the problem. Exploring embedded information in the environment improves the agents' behaviors, where agents with a limited ontology have a certain level of knowledge about their milieu. Interactions with the environment through a series of rules and regularities enables the agents to accomplish the defined tasks via limited competencies and specifications. Coupling agents with the environment through different mechanisms allows agents to justify their actions in accordance with environmental stimuli. In these processes, hidden layers of actions, which evoke the required responses to the environments maintain higher-level actions towards the desired intentions.

In the realm of design morphologies, the environment underlies the surface geometry as a target of design or a formwork of stacking elements. Extending the environments to overall forms of design establishes two levels of effectiveness on the morphology. At one level, the environment participates actively on the design processes by modifying the agents' behaviors, while in the other, the environment is passively modulated by the agents' behaviors. Within the context of morphodynamics, the effectiveness of environment amends the regulation of inclusive organisms by considering the environmental factors within the morphogenetic movements. Including environmental factors in the development of organisms adjusts functional criteria and fabricational morphogenetic principles to the natural circumstances that organisms are accustomed.

Environment as a multi-layer space

Relating the environment to the geometrical realm provides access to the structural and mathematical definitions that are the simulation of mechanical and structural properties of force flows or the derivatives of geometric equations. In the context of a mathematical definition, the environment as a target surface provides further information to self-organize agents' behaviors. Extracting this information changes the agents' behaviors. For example, the principal curvatures of the surface provide a gradient transition among different curvatures from which the environment imposes drastic changes to the agents' morphology. In plate-like agents, the transition from positive Gaussian curvature to negative curvature changes the types of agents' morphology from convex to concave polygons.

Moreover, simulating force flows on the surface determines the strain and stress principals that explain the structural behaviors of the environment. Applying structural behaviors as effective parameters enriches the agents' activities with flow diagrams that might intervene in the steering behaviors of agents on the surface. Such effective parameters allow agents to locally reinforce the weak area of the environment, so that a stabilized morphology can globally emerge. For example, the implementation of this process informs the fibrous-like agents to gradually reinforce the predefined beam elements of the compression shell within the ICD/ITKE research pavilion 2014-15. Imposing environmental effectiveness on agents enhances the performances of agents by adapting agents' behaviors to external criteria.

The mathematical definition of an environment provides a level of topography and a level of topology. Mediating between these two levels enables agents to benefit from two sets of information. The topological map informs the agents about the inner and outer loops that indirectly coordinate relationships between agents and the environment. The topographical map provides access to the information that exist within the Euclidean space that enable interactions among the agents and the external entities. Overlapping these two system fosters the behavioral development of agents through an interplay between the two environments. Mapping this information onto the agents' data structure increases an awareness of their milieus.

Developing an environment with parallel layers, such as Euclidean and topological layers, enables agents to simultaneously access a variety of information. Within the context of agent-based modeling, the contextual environment is not limited to these two layers. By adding more layers of information, the contextual environment can be considered a multi-layer of information. This multi-layer is a new type of environment, an environment that is informed by a variety of information. Retrieving this information affects the agents' behaviors through which it helps agents to converge towards a focal point of the environment. Developing this multi-layer of information offers different ways of reading the existing information about geometrical and mathematical definition of the environment. Extending these layers of information to the fabricational morphogenesis facilitates the development of an analytical layer, which considers information about material and fabrication constraints. This analytical layer enhances the multi-layer of information to the extent that mapping agents on these layers raises the understanding of neighborhood connectivity, surface topography, and fabricational constructability.

Coalescing these layers onto the environment establishes a knowledge space from which agents coordinate their actions towards the cloud of solutions. The effectiveness environment enables agents with a low-level of ontology (specification) to rely on their competences and extract information from their interactions with the environments. From this perspective, the environment provides necessary knowledge about the problem that allows agents to extend their competences for finding a proper solution. It means that the agents require rules and mechanisms to extract knowledge from environments. These mechanisms advance agents with the freedom of interacting locally with subproblems. Solving local problems might lead agents to solve the global problems. Accordingly, local behaviors correspond to local problems from which self-organizing these behaviors exhibit emergent behaviors.

5.3.3 Defining behavioral rules

Transferring behavioral agencies to agent's behaviors

Behavioral agencies determine that the whole system knows about the communal behaviors. The transition between a global level of behaviors to a local level of actions provides agents with layers of action to foster their decision-making procedures. The assemblage of these behavioral layers furthers agent-based systems to exhibit self-organization and emergent behaviors, which might go beyond the initial purpose of determining global behaviors. Therefore, global behaviors arise from accumulating sub-layers of actions or behaviors that are generated through local interactions. Connecting global behaviors to macro-regularities evokes a state to consider the purpose of model as a series of tasks. Individual and collective agents have to perform these tasks within the process of modeling.

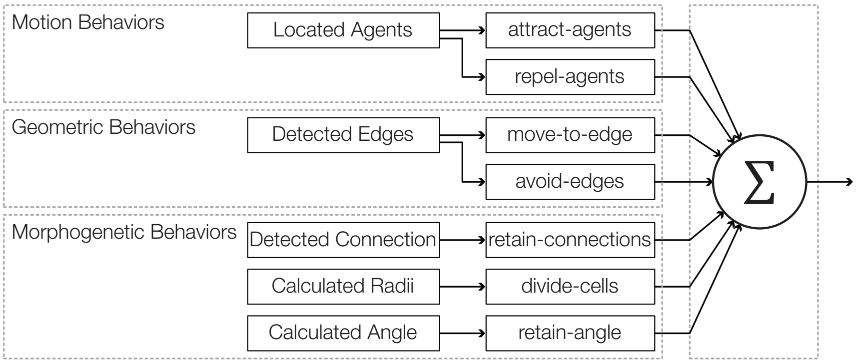


Fig. 5.3.2: The schematic diagram for generating plate-like-agents with one setup of behaviors. Inspired by Arkin (1998, p. 119).

In the context of behavior-based systems, local behaviors orient agents' structures toward task achieving mechanisms consisting of different behavioral layers. Therefore, completing a task requires an assembly of singular layers, each layer is responsive to specific stimuli. For example, generating one plate-like agent requires a consideration of behavioral layers, such as motion behaviors, geometric behaviors, and morphogenetic behaviors (Figure 5.3.2). Each of these behaviors needs a stimulus to activate the behavioral layer. The appropriate responses to internal and external stimuli relies on algorithms or rules developed within each layer. In addition, behavioral layers further the generated responses with coordination systems to properly assemble generated responses. Therefore, assembling layers of behaviors merges various responses into a specific action that leads the agents towards performing the task. Figure 5.3.3 shows the intrinsic and extrinsic activation of behavioral levels. The inhibitory mechanism restrains the internal and external stimuli and the coordination mechanism coordinate the generated responses to complete the agents' tasks.

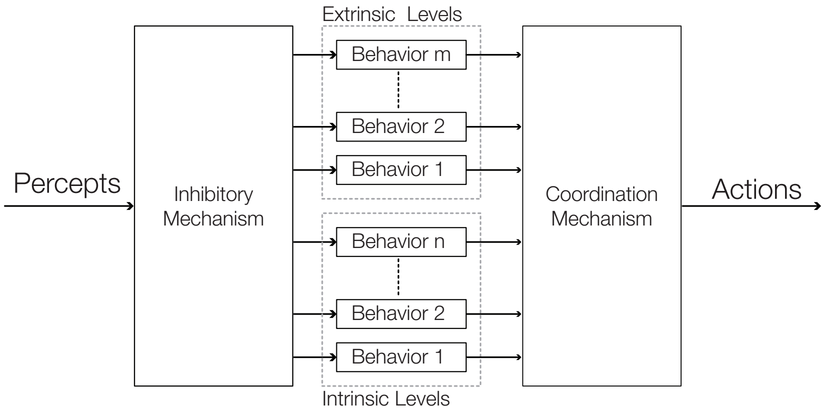


Fig. 5.3.3: The extrinsic and intrinsic levels of behaviors.

Behavioral classification

Behaviors in the generative agent-based system arise from a combination of intrinsic and extrinsic causes that rely on initial rules and tasks embedded within the agent's data structure. Embedding some beliefs within the agents enables these entities to express their genuine desires from which agents' willingness might change the course of actions towards their beliefs. Internal desires that motivate agents to perform particular actions reflect the individual autonomy. Agents are able to independently maintain their internal states, such as their morphologies. Intrinsic behaviors as homeostatic behaviors regulate the internal properties to preserve individual agents' boundaries in interaction with external entities. Confronting with external entities requires a classification of the importance of extrinsic behaviors in two levels of interactions among agents and the environment.

Behaviors of agent-agent interactions. In the case of agent-agent interaction, interplay among agents consist of a set of behaviors that specify the depth and complexity of interactions. The interstitial behaviors are effective enough to modify the connectivity among agents through attracting and repelling mechanisms, which also follow steering behaviors to aggregate agents. In addition, the complexity of extrinsic behaviors carries more engagements in the agents' morphology in which establishing and maintaining their morphological shapes require geometrical mechanisms. For instance, interaction among plate-like agents are accompanied with different algorithms, such as clipping algorithms and tangent plane intersection (TPI) algorithms. At each time step, these algorithms adaptively compute and generate the border cell of each agent (Figure 5.3.4). However, the maintenance of border cell relies on intrinsic behaviors, while the development of border cells depends on extrinsic behaviors.

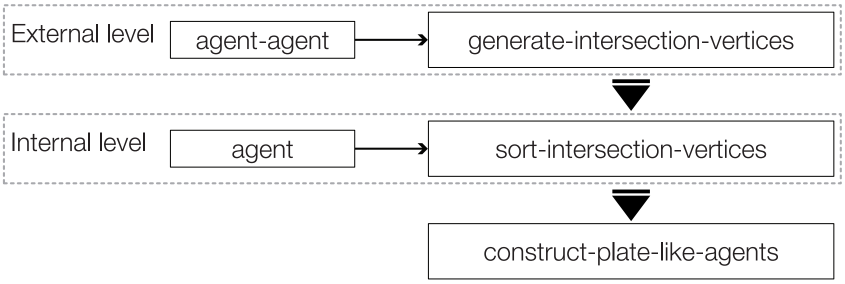


Fig. 5.3.4: The external and internal levels of agent-agent interactions for generating a plate-like-agent.

Behaviors of agent-environment interactions. In the case of agent-environment interactions, the environment provides a space of knowledge for agents, accessing that knowledge is essential for agents with a low-level of ontology (specification). In the context of behavior-based systems, agents try to accomplish their tasks by exploring and extracting embedded knowledge from the environment. The exposure of an agent's agency to the environment has direct impacts on triggering agents to appropriately act upon environment stimuli. Behaviors that emerge from the process of perceiving, processing, and acting provide a behavioral method to accomplish agents' tasks. The inclusion of agents' built environments on the environment establishes external storing mechanisms to keep the computational cost of agent behavior low. This establishment enables agents to exploit the external entities with simple mechanisms of wandering including repulsion and adhesion behaviors. Through these behaviors, agents seek to regulate their actions in confronting identical elements, such as surface boundary edges, or any type of environmental anomaly. For example, the individual fibrous-like agent converges its wandering on the target formwork to (semi-) autonomously lay fibers on its own built environments, while extracting the embedded information determines the course of its behavior on the formwork (Figure 5.3.5).

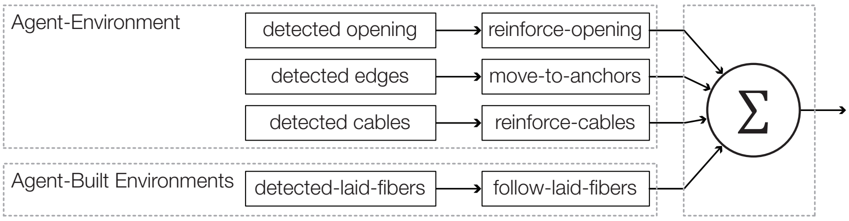


Fig. 5.3.5: The based behavior definitions for interacting the fibrous-like agent with the environment and the laid fibers.

Primitive and sophisticated behaviors. In the context of behavior-based systems, particularly subsumption architecture, the generative agent-based system employs primitive layers of behaviors[1] to exploit the external entities. These primitive layers provide basic behaviors for agents to interact with other agents and the environment. One of these primitive behaviors is the locomotion/motion behavior that relies on applying displacement factors to the current locations of agents. Determining displacement factors entails interactions between agents together and the environment in which considering the significance of each one represents the influence of that interaction on the agents' behaviors. The first layer of behavior is highly effective on approaching agents to the desired goal. For example, an individual fibrous-like agent wanders across the formwork based on its motion behaviors. Orienting these actions towards the agents' tasks requires stimuli to activate corresponding layers. Therefore, when the individual agent perceives attractor or repulsion points, such as external cables or an anomaly on the formwork, the agent activates attraction or repulsion layers upon the motion behaviors. After that, a summation of these layers of actions steers the behaviors of agent.

In the context of multi-agent systems, randomly distributing agents in the environment increases the probability of finding appropriate states. The basic locomotion behavior fosters the distribution of agents to their particular weights and values, which effectively changes the states of agents. In addition, the state of agents includes different layers, which are accompanied with negotiations among agents and the environment. In the sense of complex adaptive systems, these negotiations tend to adapt generated complexities with the main purpose of modeling. It means that applying the primitive behaviors accompanied with constraining mechanisms might gradually reduce the system volatility with convergence towards a satisfactory state. Figure 5.3.6 schematically represents the over layering of sophisticated behaviors on primitive behaviors. It allows agents to calculate the summation of the sophisticated behaviors onto the primitive behaviors. Moreover, within individual agent systems, reducing the number of agents to one individual agent necessitates the utilization of a suitable strategy to explore embedded information. In this case, primitive behaviors focus on exploring the main aspect of the model, which

---

[1] According to Pfeifer and Scheier (2001), the term behavior is defined as the result of interaction between agents and the environment. In subsumption architecture, they suggest the use of "layer" or "module" for internal mechanism instead of "behaviors" or "task-achieving behaviors" that Brooks (1986) applied in "behavior-based approaches" (Pfeifer and Scheier 2001, pp. 199-200).

is embedded within the environment, for example, following the structural elements to enhance the stability of the global geometry.

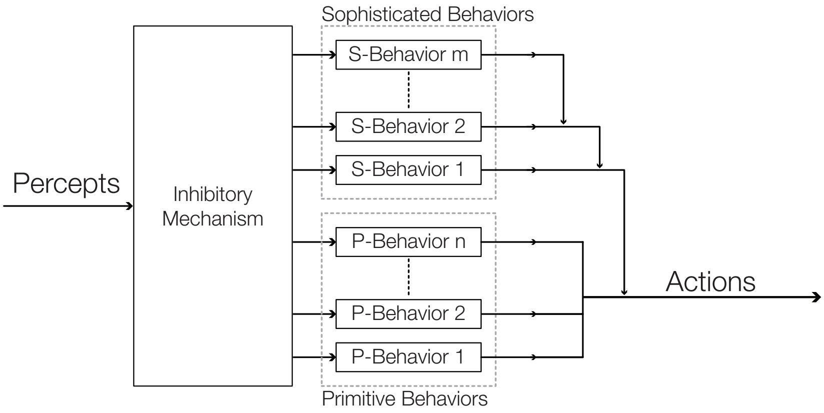


Fig. 5.3.6: The mechanisms of assembly sophisticated behaviors onto the primitive behaviors.

Comparing the behavioral layers within these two types of agent-based systems enables an understanding of the significance of primitive behaviors in driven behaviors. Primitive behaviors at the low-level layers enable advancing agents' behaviors by adding more sophisticated behaviors at higher layers. Extending behavioral layers with more sophisticated actions necessitate determining a set of predictive responses to various stimuli. Structuring and applying these behaviors relies on the purpose of the model.

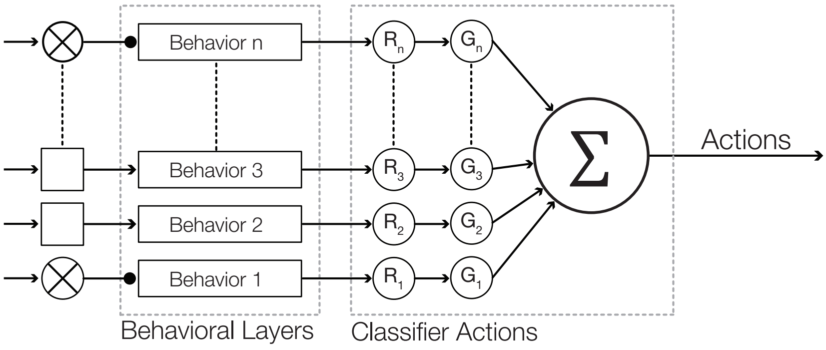


Fig. 5.3.7: The intervening mechanism of behaviors by activating and deactivating the behavioral layers.

In the behavior-based system, disentangling various behaviors into basic and advanced layers enables an examination of the effectiveness of sophisticated layers on arising macro behaviors. After that, the behavioral weighing mechanisms apply these assessments to value sophisticated layers. Assembling the rated layers with primitive behaviors triggers the ultimate response to the environment. In addition, the intervening mechanism provides an interface for the designers to activate and deactivate behavioral layers. Isolating behavioral layers furthers insight into the behavioral layers (Figure 5.3.7)

Eventually, utilizing control mechanisms to constrain the agent's behaviors enables the most exact determination of the next displacements of agents. These mechanisms allow the agents to smoothly reduce their disturbance within their actions. The significance of constraining mechanisms is evident in Constrained Generating Procedures (CGPs). In this sense, limiting agents' behaviors with control mechanisms indicates generative agent-based systems that constantly constrain generating alternative behaviors upon primitive behaviors. Therefore, Constrained Generating Behaviors (CGBs) within agent-based systems integrate behavior-based systems with the constrained generating approaches. Coalescing these two methods into agent-based systems assists steering behaviors to regulate the purpose of modeling.

5.3.4 Defining communication network

The establishment of a generative agent-based system relies on the integration of two levels of agent-agent and agent-environment communications. Furthermore, considering elements of the environment, such as patch systems, as static types of agents fosters another level of communication between environment-environment, which suggests a structural similarity of transition information to agent-agent communications. Concisely, the interstitial communications among agents within agent-based systems underlay two methods of direct or indirect contacts. For instance, the experiments of the assembly agents and the plate-like agents benefit from direct communication while the fibrous-like agent, as an individual-based system, was facilitated with indirect communication.

Direct communications are associated with negotiating one agent with either an agent or a group of agents. This negotiation comes with exchanging information by accessing the data structures of agents. Existing data within the agent's structure specifies different classes of data, such as geometric properties, availability states, spatial coordination systems, and neighborhood connectivity. Direct access to all existing information within an agent requires proper procedures to control the level of access for avoiding unnecessary computations. The network of direct communications among agents facilitates two methods to find the adjacent agents. One method considers topological connectivity, such as von

Neumann or Moore neighborhoods and the other includes Euclidean space to find the Cartesian distances between agents. Due to the need to control the number of interactions, this network connectivity is accompanied with "IF/THEN" mechanisms to reduce or increase the number of agents. The simple mechanism that provides this level of control sorts agents' relationships by considering agents within the specific range of radii (Topological connectivity) or distances (Euclidean space) (Figure 5.3.8). Considering the different numbers of agents within the connectivity network alters the number of interactions. This alteration is significantly effective on generating macro behaviors, through which the inclusion of one or a group of agents, in the process of interaction, exhibits different self-organization and emergent patterns.

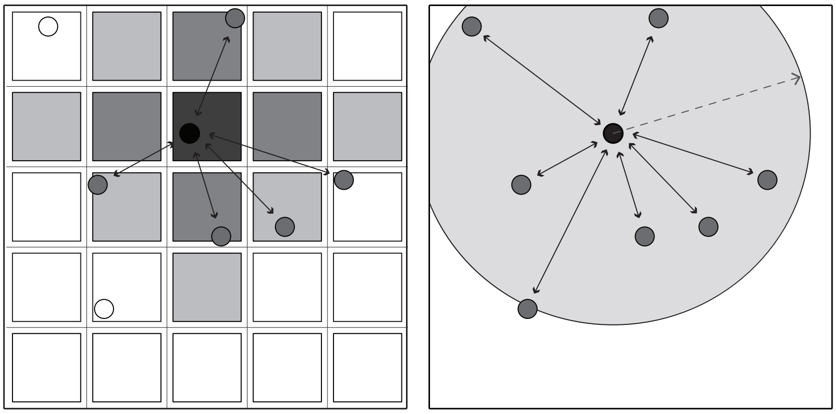


Fig. 5.3.8: The communication network; left image: The topological connectivity of von Neumann neighborhoods with the radius three; right image: The Euclidean space with the specific radius.

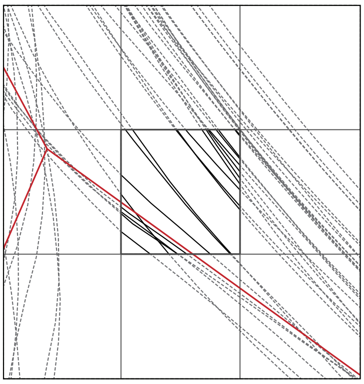


Fig. 5.3.9: A patch system with embedded trail paths and the external cable.

Indirect communication requires at least one mediator to transfer data between agents. This method is necessary when the modeling system consists of only one agent. The individual agent with low-level of ontology relies on stored data in the environment or the mediator. However, this method is also applicable for multi-agent systems. Both the individual agent systems and the multi-agent systems exploit the environment as a mediator, where agents, through storing data, modulate the environment for further exploration. Storing data establishes an indirect link among agents by adding signals to the environment, or between agents and their built environments (Figure 5.3.9). In biology, indirect communication methods, such as sematectonic or stigmergic communication, enable agent-based systems to access hidden signals in the environment.

The process of embedding information within the environment relies on discretization. In relation to the environment dimension, these small parts are described as pixel or voxel systems from which the pixel system is consistent with patch systems. The geometry of a patch system relies on the method of discretizing the surface; for example, triangulating the surface generates a triangular mesh topology. Each mesh face as a patch follows the topological definition of the mesh system. Storing information in patch systems enables agents to get embedded information via a mediator. The mediator allows the transfer of data among agents, the built environment, and the environments through a transfer protocol. The protocol includes the topological connectivities among patch systems and the process of storing and accessing the information. These processes foster a simple indexing mechanism. The indexing mechanism assigns a reference number to each patch. The reference number enables agents to access the stored information. For example, the ICD/ITKE research pavilion 2014-15, the inflated formwork represents the environment that is discretized with a triangular mesh algorithm.

In the indirect communication among agents, the significance of the environment refers to mediating between stored information and agents, while these processes reveal a direct contact between agents and the environment. This method of storing data utilizes the topological definition of the mesh system to gain the topological connectivity of a discretized environment. However, the effectiveness of the environment is independent from the stored data. Agents that are behaving on the environment detect external factors by predefined mechanisms. Exploring the environment with topographical and topological methods advances the perception mechanisms to interact with external factors.

At the topographical method, direct communications coordinate the involvement level of agents with external factors, for example, avoiding the obstacle, attracting to the anchors, and tangential behaviors to the external structural elements. Recognizing the relationships between agents and external elements within Euclidean space enables the agents to coordinate their displacements factors. In parallel, topological methods determine the topological connectivity of agents by informing agents about the manifold. The inner and outer loops restructure the topological connectivity of the agents while they dynamically redefine the topological types of agents, such as agents at the edges, corners, or at the middle. Accordingly, the direct communications of agents with the environment underlies two parallel systems that have the necessary information about the environment’s effectiveness.

6 Behavioral Strategies for Inclusive Design Computation

6.1 Preamble to Behavioral Inclusive Design Computation

Behavior-based systems versus knowledge-based systems

The determination of knowledge-based systems relies on the global establishment of hierarchically distributed knowledge among parts and components of systems. The top-down distribution of knowledge determines the ultimate specifications for system components. The accumulation of specified knowledge of components is limited to results within established boundaries. One circumstance of this hierarchical organization of knowledge is system rigidity to unknown situations, for instance, "perpetual novelty"[1] of the environment. Therefore, constraining systems with limited knowledge and with controlled inputs excludes systems from adaptations beyond the determined knowledge. In this sense, knowledge-based systems are extremely volatile to undetermined areas of problem domains. Exploring these problem domains with limited knowledge is narrowed to exploit only predefined solution spaces. Therefore, knowledge-based systems provide overall knowledge about the problem domain through which it explores parts of problem domains that systems have complete domination on those.

Accordingly, knowledge-based systems resist any heuristic methods of generating new solutions, which are consequently located outside of dominated areas. In confrontation with perpetual novelty, systems are required to generate alternatives to further their adaptation with problem domains. On the contrary, behavior-based systems rely on developing components with competencies. These competencies further components to perform as individual units

---

[1] see Holland (1995, p. 35); Holland (2010, p. 23).

within the lower-level of systems. Each unit, based on its structured capability, interacts with other units and its relevant framework to trigger micro-behaviors. Coalescing these micro-behaviors advances the system to assemble macro-regularities. Generating high-level regularities is some developments of heuristic methods that underlies the microelement competencies. When developments of heuristic methods are not comparable with macro-regularities, then revising the units' competencies is required to narrow the generated macro behaviors down to the range of acceptable macro-regularities. Indirect linkages between lower-level competencies and higher-level behaviors entails intangible analysis that correct emerging behaviors out of microelements.

Heuristic models versus analytical models

In the context of soft systems, applying a heuristic method to develop a generative system enables an exploration of the solution space via behavioral mechanisms to find adequate answers. Therefore, the exploration of solution space relies on defining the units' competencies and applying them. Implementing these competencies as generative mechanisms in units provides proportional outcomes to the explored inputs. The generated approaches via heuristic algorithms reflect the experimental methods that produce non-optimal alternatives. The alternatives that provide satisfactory explanations require some mechanisms to narrow generated possibilities down to an acceptable range of aspects. The establishment of bottom-up processes within the heuristic methods controls generative systems with internal mechanisms. These controlling mechanisms underlay rule-based systems that adopt micro-behaviors to the general regularities of the model. Hence, the model is accompanied with analytical mechanisms to examine the state of generated possibilities with specific criteria that is developed by modelers.

For inclusive organisms within the context of morphodynamics, the rules are abstracted from functional factors, fabricational morphogenetic constraints, and environmental effectiveness. The generative agent-based system that abstracts the inclusive drivers to develop agents' systems requires a computational framework to model interconnections among these agencies via behavioral strategies. These interconnections are convoluted networks of different properties and capacities, derived from the geometrical aspects of material systems and fabrication processes. Moreover, the whole system is an attempt to adapt fabricational movements to performative criteria and environmental factors.

Correlations among inclusive agencies enables the adaptation of morphogenetic movements with the intrinsic and extrinsic properties of environmental effectiveness and other performative criteria. Translating these factors requires a structuring of agent systems with different attributes. These attributes could include geometric specificity (agents' morphology), perception mechanisms, the environment, behavioral rules, and communication procedures. In this sense, each aspect of the inclusive design is covered with one of these structures. For example, the geometrical definition of material system particularly contributes on the agents' morphology, while the geometric relationships determine behavioral fabrication rules and procedures among agents.

6.2 The Development of Generating Agent-Based Systems

Generative explanandum

The generative agent-based system is an explanatory model that attempts to explain macro-regularities. Macro-regularities emerge from the interactions among microelements that include different features and properties. Each feature characterizes agents with specific behaviors, from which utilizing different features develops different characters for the same type of agents. In "generative explanandum,"[1] combining different microelements as agents with various characters can demonstrate emergent properties that might explain the macro-regularities.

With plate-like agents, macro-regularities consider panelizing a target surface, such as synclastic surfaces in which micro-behaviors follow morphogenetic constraints, polygonal radii for example. Distributing an initial set of plate-like agents is expected to perform a particular task, which are allocated within the agents' rules at the lower-level of system. In the knowledge-based system, panelizing a surface with a specific size of elements determines the optimum number of plates for panelizing the target surface. On the contrary, agents with a schematic knowledge of the macro-regularities execute the micro process of cell division. After several iterations, the cell division process regulates the population of agents while it maintains the size of polygonal radii.

Accomplishing macro-regularities only relies on the precise determination of micro-behaviors. The comparison between

---

[1] This term borrowed from "macroscopic explanandum" (Epstein 2006, p. 6).

simulated results and empirical data provides insight into the main generative parameters and constraining procedures. Allocating these parameters within a generative agent-based system demonstrates the importance of a behavioral framework in producing different possibilities under the same macro-regularities. The emergence of these possibilities demonstrates the existence of multiple solutions in which optimum solutions might exist within the generated solutions. In detail, panelizing a target surface with a specific area of a and the radii of r for each component, might approximately consider n components. It can be obtained through a simple calculation of $n = c \times (a/r)$, while c denotes the constant values of complexity of the surface. This indicates the approximate number of components while considering the value of c, which determines the exact number of panels. In the behavioral framework, however, the results converge towards the optimum calculated values.

Generative ABM: Definition

A behavioral framework, which is a generative exploration method, includes two types of agent systems. One in which the agent systems are performing individually (individual-based systems), and another where the agents are collectively (multi-agent systems). Explaining the performance of these systems requires a consideration of their methods of interaction – both the individual and the collective. In individual-based systems, complex interplays between an individual agent and the relevant environment highlight the importance of the environment and the way that an agent can modulate the environment and the built environments. In this case, the generative tool explores the development of an additive process to build a structure, while it considers the environmental effectiveness onto the agent's behaviors.

In multi-agent systems, interactions between agents produce a form of collective consciousness. For example, the assembly behaviors of agents emerge in generating a complex surface, from which clustering active agents around the assembled group represents adaptive behaviors with a specific level of complexity. This level of adaptation arises from a collective collaboration among generative agents. Utilizing the generative process in the context of integrative design computation provides a structure for a developmental morphospace. This developmental approach requires heuristic methods that are accompanied with material and fabrication constraints. Developing an analytical morphospace fosters the environment with producible areas as a possible solution space. The agents' behaviors are constrained to emerge in the developed solution spaces where generating macropatterns might be comparable to actual fabrication setups. Accordingly, exploring this fabricable space generates complex constructible geometries.

Constrained generating behaviors (CGBs)

Extending Constrained Generating Procedures (CGPs) to the agent-based system advances the agents data structures with mechanisms and procedures. In this case, agents require mechanisms to generate possibilities and procedures to constrain them. The generation of possibilities relies on triggering the perception mechanisms of agents through which rule-based agents analyze the driving factors with internal criteria, and then release the appropriate signals to the actuators. This process generates a wide variety of behaviors that selectively enable layers of actions. Generating behaviors blindly explores the problem domains, wandering without specific purpose. Therefore, similar to the CGPs, the generative mechanisms require constraining procedures, such as an analytical morphospace to improve the agents' behavior. Enhancing this process requires a balance between exploration mechanisms and analytical techniques to narrow the problem domain down to the possible solution spaces.

In particular, an agent-based system with Constrained Generating Behaviors (CGBs) utilizes the exploration mechanisms, which rely on a different level of interactions between agents together and the contextual environment. This process includes locomotion behaviors that enables the exploration of the environments considering solution spaces. Applying the constraining mechanisms adjusts the agents' exploitation to some rules. The outcome of this generative process is a set of adaptive behaviors. For example, fibrous-like agents generate robotic fabrication tool-paths by mapping their trail tracks, as constrained behavioral outputs, on the formwork. Constraining the agents' behaviors on the formwork relies on sequential stacking fibers (material agency), as well as, the velocity and reachability of the robotic arm to the formwork (fabrication agency). In addition, Figure 6.2.1 illustrates another level of constraining mechanisms that include the design data, such as external beams and opening areas.

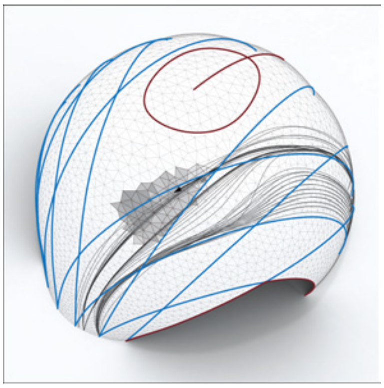


Fig. 6.2.1: The design data for constraining the agent's behaviors.

Significance of micro-levels in macro-levels

Similar to Constrained Generating Procedures (CGPs), the generative agent-based system is expected to benefit from contriving micro CGPs within agents' structure. A network of micro-generators improves the computational framework by merging the framework with a macro generative system (CGP). The macro CGPs employ the interrelated systems as a system-of-system to generate a complex inclusive system. Aligning micro-generators with agent systems emphasizes the significance of constraining the generating behaviors. In each iteration, outcomes of the micro interactions are assessed with the micro limitations. In the next iteration, the differences between the micro constraints are looped back to the generative mechanisms to reduce deviations to the acceptable range of defined criteria. Repeating this process leads agents towards the adequate range of limitations that are linked with the macro-regularities. Figure 6.2.2 illustrates the simulation of controlling connection angles, where the generative mechanism produces new locations for plate-like agents and the micro-constraints try to adopt the connection angles within the acceptable ranges.

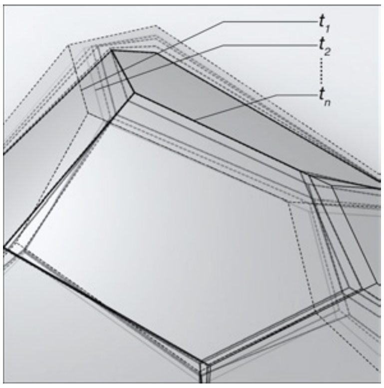


Fig. 6.2.2: The process of constraining the connection angles, where $t_1, t_2, \cdots, \textit{and } t_n$ indicate the time steps for constraining the generated connection angles.

Consequently, the macro-regularities determine the general boundary of global criteria to distribute the micro tasks to the agent systems. The distributed macropatterns indirectly establish a link between the agent's behavior and the global behavior of the system. The comparison between emerged global behaviors and the actual macropattern clearly indicates the properly engineering of micro-behaviors. The generating micro-behaviors are the rule-based methods that are task-oriented approaches. The micro-generators compute the outcomes via defined rules. If the generated micro-behaviors were unsatisfying, then it is required to consider changing the rules or the input values. Approaching the behavioral norms ensures the accuracy of the rules and the input values. In this case, the generative model is applicable for further investigations. The interplay between generative and controlling parameters exhibits emergent phenomena comparable to macro-regularities.

6.3 Controlling Systems' Behaviors

Global and local levels of interaction

The behaviors of a system emphasize the importance of interaction mechanisms that couple the system with the external environment and the internal allocated tasks. Accordingly, the interaction procedures indicate that the behavior is the consequence of accumulating various actions and reactions in response to internal and external factors. In the context of agent-based systems, agents are in constant interaction with adjacent agents and the environment. In the sense of interaction with the environment, indication of both local and global interactions provide insight into the agents' awareness of the surrounding system. Accordingly, agents with a certain level of environmental awareness will benefit from both local and global interactions to accomplish allocated tasks or design intentions. Utilizing agents to access the global level subjects to preclude agents from the imposition of top-down intentions, and to demonstrate the agents' competences for task accomplishing. Figure 6.3.1 schematically illustrates the global-local interactions, in which the global behavior adjusts the tangential normal of plate-like agents, when the local behavior constrains the polygonal radii.

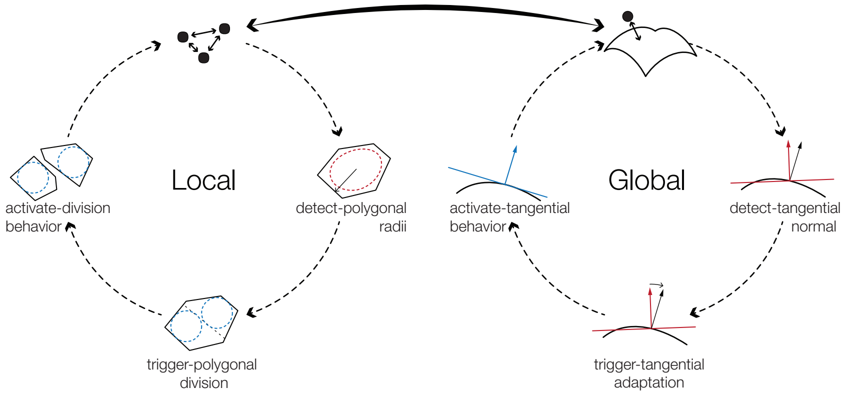


Fig. 6.3.1: The global-local effects on the plate-like agents behaviors.

At the local level, communicating and interacting with the environment require processes to integrate the environmental effectiveness into the system. The environmental factors include all necessary information that is effective in generating and triggering micro-behaviors. In addition, micro-behaviors depend on low-level communications among agents that might change the structure and behavior of the agents. In the bottom-up approach, achieving macro-regularities entirely relies on a proper understanding of generating micro-behaviors and aggregating them. However, combining the top-down and bottom-up approaches might benefit from informing agents about the global environment through which insight into the main features of the global systems will further agents to steer their behaviors towards the macro-regularities.

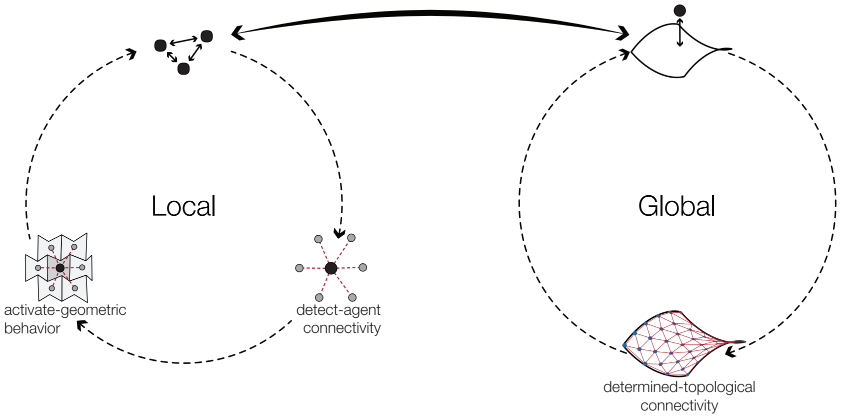


Fig. 6.3.2: The global-local effects on the plate-like agents behaviors.

For example, utilizing tangent plane intersection to dynamically compute the morphology of plate-like agents relies on triangulation algorithms, which require global knowledge about the distribution of agents' locations in the environment. Determining the topological connectivity via Delaunay triangulation algorithms requires a consideration of the relation of each agent's location with the whole set of agents. Therefore, this algorithm globally determines the local connectivity of each agent with adjacent agents (Figure 6.3.2). Regardless of informing agents with global systems, agents' system behaviors consider this information, global and local information, with different layers of actions. Each layer determines their effectiveness in the final actions, for example, generating the agents' border cells in the case of plate-like agents. Establishing system behaviors with the concept of subsumption architecture advances agents' actions through parameterizing the agents' behaviors. In this sense, each layer responds to the specific stimulus through behavioral rules that correspond to the type of interactions with the external entities.

Development of internal and external mechanisms

The bottom-up strategies for organizing the internal structure of agents allows the consideration of a specific approach to solving problems, which exist in the problem domain. This method reflects on a set of behavioral procedures to complete particular tasks, such as maintaining the polygonal connection angles in the case of plate-like agents. Considering these behaviors in behavioral layers extends the system behaviors to behave on the problem domain. Accordingly, solving problems with behavioral procedures relies on the rules that generate responses to the task of facing certain problems. Responsive rules in interaction with the problem domain converges agents toward the possible solution space. This behavioral convergence demonstrates aggregating agents around optimal solutions as a cloud of answers with diverse concentrations. Each dimension of this solution space emphasizes a particular combination of behavioral parameters in which the rule-based agents attain combinations in interaction with the problem domain. In contrast, knowledge-based approaches determine systems that are already aware of the problem, and solving the problem is accompanied by searching the predefined knowledge structures.

In fabrication processes, problems are derived from correlating material systems with fabrication tools, when designers intend to materialize their generated forms. Therefore, the process of materialization is the main problem within design processes. From the realm of CGPs, constraining the mathematically generated possibilities of forms fosters a synergy between material systems and fabrication tools. In morphologic studies, the coexistence of extrinsic and intrinsic constraints develops a theoretical morphospace that analyzes the simulated morphogenesis. In this sense, the generative agent-based design computation requires a maintenance of agents' behaviors within this hyper-dimensional morphospace. Figure 6.3.3 illustrates the developed morphospace for plate-like agents to retain agents' connection angles and their polygonal radii. The development of adaptive agents necessitates a consideration of the morphospace's constraints as driving behaviors. Therefore, conducting a theoretical morphospace that participates in navigating agents' behaviors, requires a consideration of the intrinsic constraints and extrinsic constraints of the theoretical morphospace. The extrinsic constraints that consider geometrical rules (material systems) and physical laws (fabrication tools) enables designers to consider the materialization process within the concept of a machinic morphospace. That insists on the morphological developments of forms, such as the generative process of formation that is derived from mathematical and evolutionary definition of organisms[1].

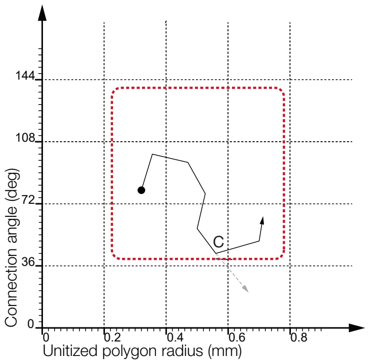


Fig. 6.3.3: A theoretical morphospace with two variable polygonal radii and connection angles. The agent behaves within the acceptable area, and it requires mechanism to adjust its behaviors at the critical point *c* to remain in the boundary. Inspired by Meyer and Guillot (1991, p. 2) and Pfeifer and Scheier (2001, p. 93).

The evolutionary pathways guide the agent's behavior toward the existent area of the organisms in which the fabricational noises, as effective factors, justify the morphogenetic movements. Developing a computational tool for employing behavioral strategies relies on the methods of elaborating rules for considering the fabricational noises within the generative design processes. Each of these rules specifies inconsistencies between selected material systems and fabrication tools. Isolating each divergence enables the

---

[1] Historical or phylogenetic factors within the Morphodynamics (Seilacher 1991a).

development of specific rules for certain problems. Validating each rule manifests an action that leads agents toward the acceptable ranges of the sub problem domain. However, the total validation of system behaviors requires an aggregation of all behavioral layers to remedy the overall problems. The consideration of fabricational noises with multi-layers of actions results in a fabricable element.

6.4 Internal and External Behavioral Mechanisms

In the context of fabricational noise, the multi-layers of actions that consider the extrinsic and intrinsic constraints of the theoretical morphospace rely on generative mechanisms to mathematically simulate forms and structures. At the same time, the generated possibilities are limited to the extrinsic and intrinsic constraints. For example, in generative plate-like agents, the extrinsic constraints express the geometry of plate-like agents by inheriting geometrical properties from plywood materials (material agencies) and the method of their interactions is related to the fabrication and assembly processes (fabrication agencies). However, the process of developing agents' morphology commences with basis geometries, such as the plane geometry within the example of plate-like agents that identifies the intrinsic constraints of a theoretical morphospace. This morphological constraint contributes as a developmental layer within the generative system.

Since the behavior of the agents relies on interaction with the external entities, the mechanisms that deal with this level of interaction determine the external level of behavioral mechanisms. On the other hand, the generating procedures that deal with the intrinsic features of agents establish the internal level of behavioral mechanisms. This means that, material agencies are effective in determining the geometric definition of agents' morphology while the interrelations among agents relies on fabrication agencies. Both of these agencies, as external mechanisms, are under the influence of internal mechanisms. The generative process of developing form correlates significantly with these agencies through their geometric and functional mediation. In addition, ignoring the significance of the environment in generating internal and external behaviors interferes with the process of stabilizing agents' attributes, such as derived morphologies by agents. Therefore, the environment not only participates in coursing external behaviors, but also is involved in the formation of internal properties, such as agents' morphology.

For example, generating cells with the clipping algorithm is only applicable for synclastic type of surfaces. Neglecting the surface curvature as intrinsic criteria fails computational tools to properly generate cells on doubly curved surfaces. This means that the clipping algorithm is only reliable for generating convex polygons, and that altering the surface curvature begins to shift the convex polygonal types to concave polygons. Moreover, the environment facilitates agents' communications with stored data, other agents, and built environments. This inclusion in environment modifies agents' performances for constructing built environments. It is obvious that stacking fibers on the formwork is indirectly coordinated by external entities. The agents' locomotion behaviors represent aggregating reactions to various external stimuli. Without attending to the external factor, agents only wander across the environment, only employing internal parameters. Figure 6.4.1 illustrates the translation of external behaviors into sophisticated behaviors that are overlaid on the locomotion behaviors.

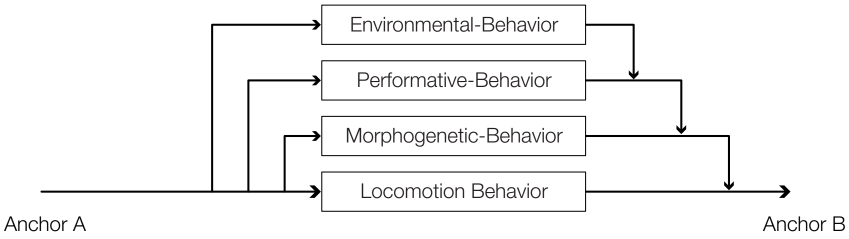


Fig. 6.4.1: The sophisticated behavioral layers, which navigate the agent's behavior from one anchor point to the other anchor point.

In the context of morphodynamics, synthesizing both material and fabrication agencies into fabricational morphogenesis considers the environmental effectiveness, such as principals surface curvatures. Agents' interactions with the relevant environment also require adapting their behaviors to functional criteria, such as structural performance. Although the environment is an external element, access to the embedded information might internally inform agents' morphology with different possibilities. For example, with the plate-like agents, different types of polygonal geometries are influenced by the principal surface curvatures. Hence, the external agency, like the environment, contributes significantly to the internal properties of an agent. The bilateral relation between internal and external levels also effect the micro-behaviors of agent systems. Changing external behaviors modulates the internal behaviors of the agents. For instance, interaction between two plate-like agents slightly changes the overall arrangement of the collective agents that might emerge in different panelization. Eventually, these two levels comprise of two different behavioral libraries that facilitate the adjustment and adaptation of agents with the intrinsic and extrinsic properties of inclusive drivers.

6.4.1 Internal level of agents' behaviors

Intrinsic properties underlay the internal features of agents that include the "morphological definition" and the "perception mechanism." The morphological definition is the extension of material agencies, while the agencies rely on the interpretation and abstraction of material systems into the agents' data structures. This abstraction requires insight into the relation between the geometrical means of materials and the morphological definition of agent systems. In addition, the geometrical procedures of material systems determine the intrinsic properties of agent's morphology. Therefore, at the level of material agency, specifying the agents' morphology denotes a geometric basis, such as plane geometry in relation to building material systems, to initiate a structuring of the agents' forms.

The process of generating the forms of agents effectively contributes to the behavioral structure that agents follow during their interaction with external entities. Manifesting the agents' form as morphology relies on a direct negotiation between the initial geometries and the influence of internal and external forces. These pressures might drastically change the morphology of the agents. For example, the morphology of plate-like agents, under the influence of surface curvatures, interplay between convex and concave polygons. In that case, agents rely on internal mechanisms to adjust the morphology of the agents with the principles of the original morphology, such as retrieving planarization aspects of plate-like agents or adjusting the polygonal structure within assembly agents. These internal mechanisms inhibit agents from losing their internal cohesions. Maintaining the agents' structures are consistent with their morphogenetic movements.

The morphogenetic movements that manifest agents' morphologies are either a derivative process of geometrical elements, such as the derivation of plane geometries to plate-like agents, or an assembly of sub-elements, such as n-gons geometrical elements to generate a complex surface. In the derivative processes that generate agents' morphology, differentiating between geometrical elements determines the agents' morphology with a specific type of geometry. The ultimate forms of agents emerge out of modulating these basis geometries. In addition, the geometric relation influences the agents' behaviors. The interaction between two adjacent agents relies on their methods of geometric intersections. Moreover, the geometric intersections change the morphologic base of agents to the representative forms of agents. The derived form, as an abstracted form, indicates the interrelation between the base geometry and its modulation.

In the integration processes of erecting agents' morphology, the aggregation properties are effective at organizing the agents, meta-agents, and meta-meta-agents. The geometrical structures of agents rely on the behavioral process of assembling elements. Similar to the derivative method, the assembling process of agents (sub-agents) follows basis geometric rules, such as planarity of agents, with determining type of polygons that imply main features for assembling sub-elements. The implication of these features underlies the sequence of interrelating sub-elements, which establish the agents' morphology. The behavioral rules for assembling sub-agents are abstracted from the geometric definitions of polygonal patterns that typically determine the ultimate morphology of agents. The connectivity of each sub-agent with the others is consistent with behavioral rules, derived from geometrical procedures. The geometric pattern determines the number of sub-agents and their behavioral correlations that structure the agents' morphology.

Accordingly, the agents' morphology evolves from two levels of geometric constituents that consist of the basis form and the abstract form. These two interrelated geometrical structures require that their consistencies be maintained in interactions with external entities. Applying the main principle of the agents' morphology, such as controlling length of linkages between sub-agents, requires mechanisms to examine deviation from basic principles of the initial geometry. This process leads polygonal structures to have equilateral edges (Figure 6.4.2). On the other hand, avoiding self-intersection among edges also requires mechanisms to order the vertices' arrangement, through which the irregular polygon is adjusted to simple regular polygons (Figure 6.4.3). The homeostasis approaches in biological organisms that adjust internal deviations with external pressures, conceptually supports the necessity of this mechanism within agent systems. However, finding the deviation from the basic geometric patterns requires mechanisms that approximate the generated morphology to those particular geometrical features. Transferring defective morphologies to the expected forms requires gradual changes to prevent completely destructing the agents' morphology. This process is accompanied with another mechanism that is responsible for keeping the integrity of the agents' morphology.

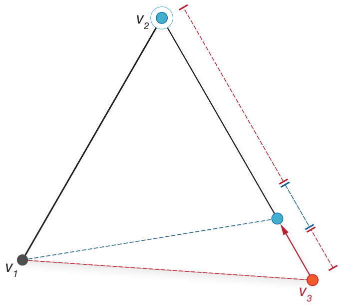


Fig. 6.4.2: The inter-node length adjustment in assembly agents experiment. Inspired by Wengzinek (2016).

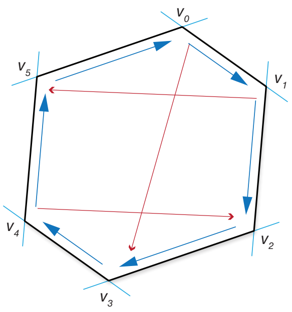


Fig. 6.4.3: The intersected vertices arrangements in plate-like-agents.

Consequently, examining the internal cohesion of the agents' morphology is a necessary procedure within the proposed generative agent-based system through which the agents self-organize the geometric structures. The intrinsic properties depend on the type of agents' morphology from which the mechanisms control different geometrical aspects of agents from planar qualities in the plate-like agents to periodic features in the n-gon agents. Each of these two types of agents, derivative and integrating, inherit the controlling mechanisms from geometric definitions and geometric relations. Extending the geometric rules to the agents' constraining mechanisms advances the layers of actions with various "IF/THEN" mechanisms through which the self-organizing mechanisms adjust internal behaviors via feedback loops. Analyzing the internal states of agents through "IF/THEN" mechanisms confirms the significance of feedback loops to adjusting agents' deviation to the intrinsic properties. This adjustment relies on internal procedures, which are regulatory mechanisms within the "IF/THEN" mechanisms, to perform comparative processes. The comparative processes consist of positive and negative feedbacks that amplify some actions and quiet others. These processes utilize self-regulating and self-organizing mechanisms to prevent internal fluctuations from intensifying the deviations. It enables agents to maintain internal consistencies against extrinsic and intrinsic factors through inhibitory mechanisms.

6.4.2 External level of agents' behaviors

In the context of morphodynamics, extrinsic properties describe external influences that change the development of organisms. Specifically, the environmental and functional factors are particularly effective on the morphological development of organisms. Adapting these extrinsic properties into generative agent-based systems furthers the development of the environment. The environment contributes to activate the functional and behavioral characteristics of agents. Considering the environments and other agents–external entities–require perception mechanisms to collect data from external entities.

For example, plate-like agents and fibrous-like agents extract extrinsic properties of target surfaces in both topology and topography. At the topological level, extrinsic properties typically consider a two-dimensional manifold of bounded areas that indicates the internal and external edges of target geometries. In addition, at the topographical level, extrinsic properties inherently include a mathematical definition of surface curvatures and its structural properties, such as stress-strain principals. In addition to extracting information from the environment, utilizing a patch system, such as pixel and voxel systems, furthers the environments to store selected agents' actions into the environment. This process allows an agent-based system to update extrinsic properties, and consequently enhances the agents' behavior. For example, the process of reinforcing external beams (cables) in fibrous-like agents dynamically alters the extrinsic properties that are related to the functional features of the environment.

Agents' access to extrinsic properties triggers different layers of actions. Adjusting agents' behaviors to perform allocated tasks requires coordinating mechanisms. The coordination mechanisms adapt agents to any complex situations via activating and deactivating behavioral layers. For instance, coordinating complex situations, such as anomaly within the environment is accompanied by changing the course of agents. As an example, fibrous-like agents wandering on the formwork consider external edges, approaching the formwork's edges triggers layers of action that adapt agents to the environment. The process of adaptation is a summation of diverse behavioral layers from which integrating these layers results in a selection action. Applying the selected action coordinates agents' behaviors toward special adaptive behaviors. In general, extrinsic properties effectively coordinate the agents' actions within the environment. Developing proper layers of actions can effectively ease approaching agents to the adaptive level.

In the context of the agent's agency at the fabrication level, the morphological type of agents that actively participate in the fabrication processes as fabricator agents relies on the external environment to communicate with other agents or their built environments. Indirect interaction with other groups of agents promotes an extension of individual-based systems to the collective systems. Storing agents' trails and signals onto the environments enhances their level of adaptation and learning. In addition, considering the environmental effectiveness and self-organizing the performative criteria enable agents to regulate their actions on the environment as the formwork. The individual rule-based agents, instead of collecting their behaviors into the memory systems, store traces as the built environments on the formwork. Utilizing this method of stigmergic or sematectonic communication enables agents to retrieve their modulations from patch systems within the environment. Fabricator agents, such as fibrous-like agents, leave traces on the formwork, which are interpreted as built environments. Considering these traces, as tool-paths for robotic fabrication tools, advances materializing the design process along with design formation.

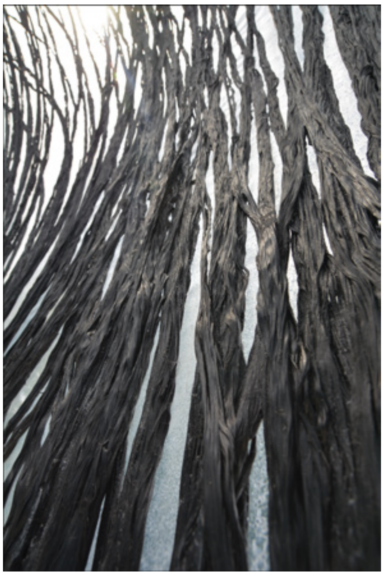


Fig. 6.4.4: The laid fibers on the formwork; source: ICD/ITKE University of Stuttgart.

In the fibrous-like agents, the additive process of fabrication lays down fibers on two different systems: The environment as the formwork and previously laid fibers as the built environments. Erecting morphology by stacking fibers on the formwork relies on the processes of coordinating agents' behaviors on the environment (Figure 6.4.4). This process follows aggregating interrelated behaviors abstracted from structural, biological, and fabrication principles in which they all have different influences on the agents' behaviors. The derived morphology gradually combines the agent's behaviors as tool-paths for robotic fabrications. Linking robot end-effectors to the agent's traces simulates fabrication movements based on the agents' behaviors on the formwork. This process reflects the simulation of behavior-based robotics. The movements of an industrial robotic arm are coupled with the behavior of the computational agent. In this computational simulation, the agent behaviors are manifested in the simple NURBS curves geometry. These curves interpret the simulated geometry into robot tool-paths that determine the physical robots' behaviors. Coalescing the simulated morphogenesis with physical manufacturing blurs the gap between formation and materialization. Figure 6.4.5 shows the process of laying fiber on the formwork based on the simulated tool-path via the computational agent.

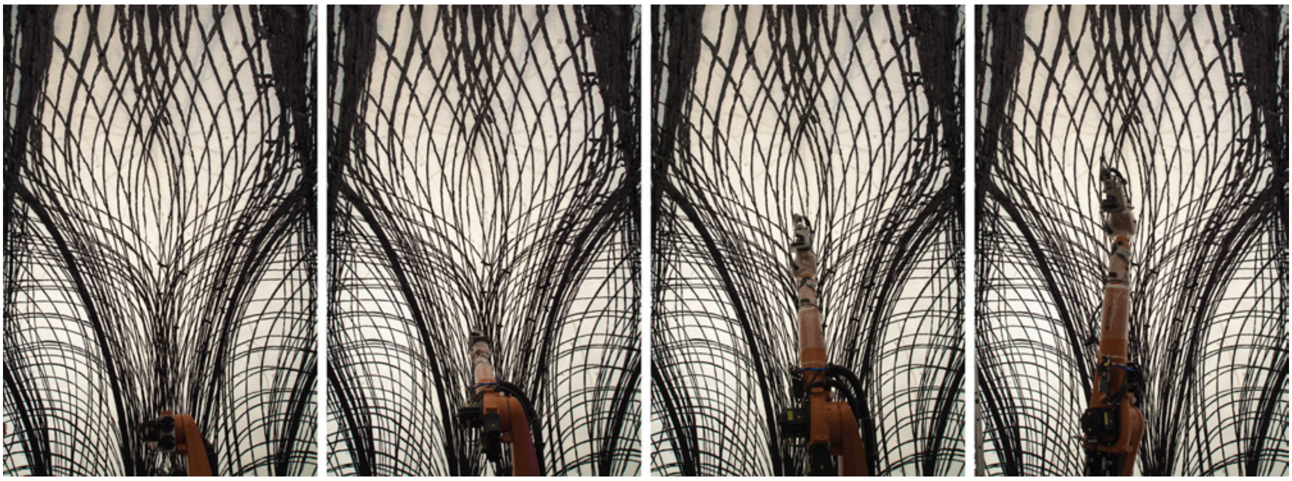


Fig. 6.4.5: The sequence of laying fiber on the formwork.

In the case study ICD/ITKE research pavilion 2014-15, the fibrous morphology arose from generated computational patterns that are the result of micro modulations between a computational agent and the fibers systems. In this process, the agent, without any knowledge of the ultimate morphology, mounds fibers on top of laid layers until it achieves the design principles. Gradually developing a fiber composite promotes the growth of an adaptive system by coordinating agents' activities in both the local and global levels. The local interactions between the fabricator agent and the environment traces geometrical patterns that effect on the next actions of agent. The evaluation of growing patterns at both the micro and macro levels is accompanied with the local adaptation to environment and global implementation of performative criteria. Adapting agents' activities to local conditions, which are consistent with the fabrication setups, includes the accuracy of agents' behaviors on the formwork, maintaining agents within the acceptable zone and associating agents' traces with the built environments. The local controlling mechanisms coordinates agents' behavior in each pace through which the agent tries to adjust its behaviors with local rules. The effect of local modulation might emerge in the global adaptation, however, failures in global adaptations requires an implementation of performative criteria to regulate the local behaviors for the global adaptations. Global rules for this level of adaptation are abstracted constraints that indirectly affect the agents' behaviors. These procedures include performative adaptations, such as mechanical and structural properties, which globally analyze the generated morphology and locally inform the necessary modifications of local rules. In addition to coordinating behaviors, this process is effective in maintaining the morphological structures by considering the significance of global performances.

In general, extrinsic properties can be summarized in two adaptation levels of environmental factors and allocating tasks to agents. Abstracting design intentions as demanded tasks into the behavioral classes advances the environmental effectiveness, for example, interpreting this process through the fields of vectors that indirectly inform agents with the global design patterns. Integrating tasks with the environmental factors reflect interactive negotiations between these two levels to obtain macro-regularities. Therefore, the developed vector fields and environmental factors indicate stimuli layers to trigger the agents' responses that are intended to adapt their behaviors to external pressures. Systematically weighting these layers contributes to investigate on the leverage points to find the adaptability of each layers with the environments. For example, prioritizing structural layers might significantly increase the strength of the generated morphology. Consequently, iterating this process by modifying effective parameters might enhance the desirable states.

6.4.3 Synchronization agents' behaviors

Similar to behavior-based systems, the differentiation of behavioral procedures into inhibitory mechanisms and coordination systems enables agent-based systems to contain internal and external pressures. Filtering these influences facilitates inhibitory mechanisms to adjust internal behaviors with sudden imposed conditions. This means that synchronizing the internal mechanisms with an external one follows two sequences of the adjustment of behavioral inputs and the activation of behavioral outputs. Therefore, the behavioral procedures require a delay to adjust internal mechanisms and then send the adjusted signal to coordination systems to activate behavioral layers. This delayed activation provides agents with time lapses between maintaining internal consistencies and employing next coordination. In addition, the temporary lapses filter the external influences through inhibitory mechanisms to trigger specific behavioral layers. Considering inner states of stabilities alternates the triggered responses in which the generated signals at these states are optimum levels for consideration in coordination systems. Accordingly, scheduling procedures enables agent systems to systematize correlations between behavioral inputs restrained within inhibitory mechanisms and behavioral outputs generated via coordination systems. Therefore, behavioral sequencing in agent-based systems requires adjusting input parameters with output behaviors to increase the adaptation of system to external complexities.

For example, the adjustment of input parameters fosters the assembly agents to stabilize n-gon geometries before each individual polygon tries to assemble with the main cluster of polygons. In the plate-like agents, the inhibitory mechanism furthers geometrical behaviors to avoid self-intersection during the process of generating plate-like agents. Therefore, inhibitory mechanisms consider several procedures to restrain the input parameters, to adjust inner states of agents and activate behavioral layers. In particular, inhibitory mechanisms include three main levels: perception to contain the inputs, "IF/THEN" rules to adjust and activate behavioral layers, and tagging sequences to flag agents' behaviors. Within a behavioral system, the implementation of these levels provides a prohibiting layer of certain features of perceiving environments, categorizing inputs, adjusting internal states of agents, and tagging current conditions of agents. The perception mechanism within the fibrous-like agent determines conditions of the agent. The "IF/THEN" mechanisms evaluate the agents' conditions and attributes. The tagging sequences flag agents to signify their states. Then, the inhibited or abstracted states activate essential actions to control the flow of agents' behaviors.

Coordination mechanisms consist of integrating different layers of actions to select the next possible action for the agents. This process is accompanied with prioritizing mechanisms that weigh each action in accordance to their significance in fabrication movements. This prioritization directly modifies the agents' behaviors in response to external and internal levels. However, the internal properties of agents, such as their morphologies, dynamically adapt their geometries to external demands. It means that the intrinsic properties of agents are coupled with agents' efforts to find the adequate solutions. Accordingly, coordinating agents' locomotion to exploit the problem domains is accompanied with the lower-level of agents' properties. The agents synchronize internal and external levels through different procedures, such as adjusting its morphology or adapting its behaviors on the environment. Moreover, coordination systems through exploiting the environment establishes self-organizing mechanisms to complete assigned tasks.

6.5 Intervening Mechanisms: Monitoring Systems

Developing an inclusive design tool that provides a study of integrating fabrication and material constraints into design processes, requires a computational framework based on agent-based systems. The structure of generative agent-based systems requires specific procedures, such as developing agent systems, determining the environment, and indicating agents' behaviors and communications. Interrelating these procedures fosters agent-based systems to perform desired tasks. Formulating tasks for agents reflects a general overview of macro-regularities in order to indicate types of agents, rules, and regularities for determining the interaction mechanisms of agents. The completion of tasks follows accumulating different values of behavioral variables. Employing different set of behaviors steers agents towards the acceptable range of solutions. Hence, parameterizing the behavior systems might enhance the computational framework to explore the problem domain through user interventions. The intervening mechanism that is a parametric method of changing the value of behavioral parameters eases direct access to behavioral layers of actions in which users can dynamically change behavioral rules and communication mechanisms. Moreover, this mechanism, by maintaining the type of agents and the environment, can alter their properties during the simulation.

The agent-based system that exhibits self-organization and emergent properties, requires an analysis of the importance of each parameter within the process of modeling and simulation. In this context, the experiment of determining the appropriate rules for controlling the polygonal radii exhibits the emergence of system, when the cell division algorithms are applied with isolated setups. Isolating all parameters enables designers to understand the importance of the initial number of agents. The comparison between two initial states, one with three agents and the other with seven agents, emphasizes the role of hidden criteria, which arise from differences between the number of interactions and communications. The number of elements within the simulation indicates that the same constraining mechanisms will generate different behaviors. The emergence of the system represents the final equilibrium states of the system, which exist within the solution spaces, but with a different numbers of generated cells.

Recognizing the role of each parameter relies on monitoring all possible states generated by different sets of values. In generative agent-based systems, each setup might reflect a collection of behaviors. The generated behavior requires consistency with "IF/THEN" procedures, such as the analytical morphospace. It means that monitoring mechanisms for each parameter require a consideration of an interval correlation with each dimension of morphospace. In addition, the analytical morphospace, as a possible solution space, can indicate a parametric space. This parametric space changes the dimensions and the values of the morphospace. In addition, determining each dimension of the hyper-dimensional morphospace expands the parametric space by defining a specific range of values. Moreover, parameterizing these values alters the analytical morphospace and, consequently, the overall agents' behaviors. In that case, changes in theoretical morphospaces modulates micro-behaviors through which agent-based system self-organizes its properties to lead the system toward a new level of emergent behaviors. It follows by constraining the solution space through the development of a parametric system to confine agents' behaviors.

In addition to the analytical morphospace, parameterizing a generative agent-based system includes modifying agents, remodeling the environment, and revising the behavioral systems. Modifying agents commence with initializing agents as a sequence of defining a number of agents, stochastic distribution mechanisms, and determining the agents' location in accordance to topographical or topological map systems. Moreover, parameterizing the agents' attributes related to morphologic properties or auxiliary perception mechanisms, such as the range of field vision, might dynamically alter exploiting mechanisms in which it might directly change their course of agents' behaviors. After parameterizing the agent system, remodeling the environment is too delicate for a generative agent-based system where the agents' behaviors directly or indirectly rely on the effects of the environment. Although, a slight modification of the environments fosters a new configuration, drastically remodeling the environment requires informing the system at the early stage of initializing agents. This means that determining the topological and topographical definition of the environments informs agent distribution. Any topological redefinition must be informed at the early phase. In the context of revising behavior systems, the layer structure of behavior-based systems enables the weight of each layer of action. Users are able to revalue the predefined set of weights for each layer. Revaluing the behaviors, from neutralizing to intensifying them, advances the generative system to study the significance of each action through a transfer of the relative autonomy of agents' decision-making to the user demands.

7 Conclusion and Discussion

7.1 General Review

The main objective of this thesis was to investigate the generative potential of behavior-based systems when integrating materials and fabrication constraints into a computational framework. A generative computational framework based on a low-level of distributed units or agents was developed to facilitate this integration. This development was investigated through a series of experiments and case studies that highlight the importance of behavioral strategies within formation and materialization.

This research proposes the utilization of agent-based systems as a computational approach to adapt materials and fabrication processes for architectural design. Utilizing this approach in four main experiments reflects the theoretical framework of this computational method. In parallel, the theoretical development of these approaches was implemented in practical studies to showcase the importance and applicability of this research. The transition from theoretical approaches to built architectural elements further enhanced this research. The results of these practical developments were effective in fostering behavioral strategies within integral design computation.

This thesis required a background of knowledge on the development of a computational model that mediates effective drivers for coalescing materialization and formation. Therefore, the background and ramification of integrated design were divided into four parts to challenge the integration processes within architectural design. The first part raised the question of integration methods, which applied to architectural design, particularly in materializing developed forms. This question applied more specifically to non-standard design, which often challenges designers, and requires innovative methods to rationalize design intentions. The second part, which drifted towards biology, briefly addressed morphologic studies to highlight the main factors in the evolvements of inclusive organisms. This investigation demonstrated that inclusion drivers, such as fabricational morphogenesis, environmental effectiveness, and performative criteria worked as active potencies in design computation. An amalgamation of these drivers required a framework to mediate their fusions. Part three studied the development of generative systems to model the synthesis of the inclusive drivers. The complexity that arose from this inclusion led the inquiry to part four, which introduced agent-based modeling and simulation that exhibits adaptive complexity. This introduction was followed by the next part, which emphasized the importance of agent-based systems in different field of sciences, more specifically, in architectural design. The experiments and case studies were structured to adjust and examine the potential of agent-based systems for integrating inclusive drivers. Chapter 5 explained the main approaches to setting-up generative agent-based applications. In particular, generative agent-based computation was argued as a mediator between material and fabrication agencies, while these two agencies relied on environmental factors and performative aspects. Chapter 6 highlighted the significance of behavioral strategies within the generative agent-based architectural design computation. In detail, this chapter discussed a negotiation between effective drivers of inclusive design, and how these drivers can change the process of integration from the bottom-up.

7.2 Generative Agent-Based Design Computation

Generative agent-based design computation introduces a promising method for coalescing materials and fabrication characteristics into design processes. The main structure of the experiments was followed by an investigation on behavioral integration to demonstrate a new strategy for fabricational morphogenesis. Implementing behavioral strategies within a computational framework required gaining insight into agent-based modeling through soft systems. Expanding soft systems into a design and fabrication paradigm required the development of computational frameworks in modeling to encode semantic models into syntactic models. The formalization of semantic models was investigated to emphasize the significance of the agent's agencies in both abstraction and transition phases. In the abstraction phase, material and fabrication agencies are mapped to a computational model at time t through equivalence classes. And then in the transition phase, the generative model produces new states at time $t + 1$ through behavioral algorithms.

The purpose of models was discussed as the main factor for developing models. The purpose of behavioral models was addressed by macro-regularities, which arose from the interplay between micro-behaviors. The indirect relation between outcomes at the macro-level and interactions within a micro-level is considered through the development of agents and their behaviors. The specification of agents and the process of generating regulatory behaviors were considered the key to achieving the purpose of the integrative models. The experiments represented the relation between self-organization and emergence within both the micro and macro levels. Self-organization regulates the interplay among agents to obtain the purpose of the model. This revealed that the equivalences between the emergent properties of the model and the desired regularities are an outgrowth of the agents' self-organization properties at both local and global levels.

In particular, when the purpose of the model materialized complex forms during the process of formation, the model was expected to generate producible building components or fabrication processes. The producible outcomes were investigated through agents' agencies; the interactions among agents demonstrated the importance of controlling agents' behaviors. The ethological or behavioral morphology was discussed within insects' formation and their artifacts. Ethological morphogenesis was introduced at two levels, by aggregating agents or by modulating agents. This classification furthered the material and fabrication agencies to develop an inclusive design computation. The inclusion of the capacities and the constraints of these agencies entrenched the main principle of generative agent-based design computation. This principle was designated in four steps: agent specifications, environments' structures, behavioral procedures, and interaction or communication principles.

The agents' attributes and types were organized through ethological morphologies, in which the realization of complex forms relied on the agent's morphology or was constrained to the agents' body. Accordingly, experiments were developed to consider both of these ethological concepts, which emphasized the significance of agents' morphologies and how these morphologies modulated during their interactions with the other agents and the environments. Developing agents' types answered the need to relate the material and fabrication agencies to the geometrical definition of agents. The material agencies emphasized the morphological aspects of the agents, while the fabrication agencies underlined the importance of the agents' body to the construction processes. This comprehensive classification fostered by environmental consideration, which included extrinsic and intrinsic criteria. Therefore, the environment imposed another level of modulation on the material and fabrication agencies, from which it was effective on the development of morphologies and the behaviors of agents. Concisely, under the influence of the environment, the fabricational morphogenesis self-organized the relationship between material and fabrication agencies.

Self-organizing mechanisms in active agencies were provided by the topological and Euclidean spaces that were overlaid on theoretical morphospaces. The amalgamation of various spaces under the context of the environment utilized the development of behavioral procedures and communications. The behavioral procedures extended the inclusion of material, fabrication, and environmental agencies over behavior-based systems. The use of behavior-based systems allocated different layers of actions for each agency, wherein prioritizing processes highlighted the importance of each layer upon primitive behaviors, such as locomotion behaviors. The summation of these layers determined the transition behaviors from one state to the next state, which could change the morphology of agents or could build some variations on the environment.

The last step for developing the generative agent-based system argued that the significance of the communication networks in agent-agent, environment-agent, and environment-environment interactions. More specifically, the argument categorized the communication among agents into direct and indirect methods. Direct methods focused on the immediate interactions among agents, while indirect method emphasized the significance of interaction with the environment as a mediator.

These two methods examined the application of material and fabrication agencies to represent the importance of communication and interaction within morphogenetic movements. These four steps determined the agent-based modeling and simulation in the inclusive design computation. The development of scaled models, such as the rob|arch 2012 prototype and its advanced model, which was shown at the Landesgartenschau Exhibition Hall 2014, and also ICD/ITKE research pavilion 2014-15, proved the potential of agents' agencies to integrate materials, fabrications, and the environment. Due to this potential, generative agent-based design computation opens a new chapter to study behavioral strategies within computational design and construction.

7.3 Behavioral Design Computation and Construction

In the context of inclusive design computation, behavioral strategies were investigated to propose a novel method of integrating fabrication and material constraints into design processes. In particular, a comparison between knowledge-based systems and behavior-based systems unfolded the confrontation methods to the problem domain, wherein the former required the global domination and the latter relied on local competencies. The micro unit competencies consisted of material and fabrication constraints and capacities from which their interactions could erect producible design elements. The behavioral strategy of integrating these competencies raised the possibilities and objections of an extremely versatile system, which dealt with the problems of Constrained Generating Behaviors (CGBs) towards the adequate solution space. It was determined that behavioral strategies can only exhibit emergent properties or organized complexities when all generated behaviors were examined and compared with the possible solution space. Therefore, in the context of generative agent-based systems, a main contribution to behavioral strategies, applied within design computation and construction, leads to the following three main conclusions:

Generative agent-based design computation

The investigation on utilizing generative agent-based systems in computational design and construction determined the importance of micro-behaviors in explaining macro-regularities. Macro-regularity was employed as the main purpose of the model, because it regulates the process of design. In contrast, a hierarchical organization imposes detailed behaviors on lower-level components. The micro-behaviors were indicated as the competence of units to accomplish the determined tasks, where each task required customized rules and regularities. It was indicated that connecting micro and macro levels require an analysis of the macro-level to find the general regularity for determining the units' tasks at the micro-level. Therefore, the behavioral framework was considered as individual agent systems and multi-agent systems, through which the micro-behaviors arose from the competence of agents. It was discernible that merging micro-behaviors regulate orders at the level of macro behaviors that were implicit to the macro-regularities, due to the behavioral characteristics of the system. This suggested that the interactions among agents could exhibit emergent properties, while the generated results might be in correlation with macro-regularities. The experiments confirmed that the completion of micro-units' tasks perpetually produces novelties to attain macro-regularities.

Constrained generating behaviors (CGBs)

The concept of constrained generating behaviors was developed to extend behavioral-based systems to constrained generating procedures. This method was implemented within generative agent-based systems to steer agents' behaviors towards the purpose of the model. The experiments demonstrated that exploring the problem domain requires constraining mechanisms to converge agents' behaviors to the appropriate solutions. The constraining mechanisms within the concept of fabricational morphogenesis was advanced with the development of the theoretical morphospace. The theoretical morphospace was used to fuse material and fabrication agencies together. The influence of each agency was changed from adjusting material components together (e.g., plate-like agents) to facilitating the reachability of fabrication tools within its workspace (e.g., fibrous-like agents). It was realized that the inclusion of constraining procedures, in particular the theoretical morphospace, is an important mechanism that regulates the micro-behaviors of agents to synthesize the producible design.

Controlling behavior systems

The interaction mechanism for agents was elaborated, with limited specifications, at local and global levels. The study emphasized that the interaction at the local level triggered micro-behaviors, to approach macro-regularities. It concluded that the utilization of constraining behavioral mechanisms also required global levels of knowledge to course further agents' behaviors. The adjustment of local behaviors via interactions with global knowledge proved the importance of this knowledge for finding agents adjacent to the generated agents' morphology (e.g., triangulation mechanisms applied in plate-like agents), or informing agents about the strain and stress principals of target surfaces (e.g., reinforcing the external cables on the compression shell in fibrous-like agents). In accordance to behavior-based systems, the controlling behavior system was entrenched to different layers of actions from which this system was advanced to prioritize the behaviors.

The study of the theoretical morphospace unveiled intrinsic constraints, which dealt with morphological developments and extrinsic constraints, which are considered fabricational noises (machinic morphospace). The association of constraining mechanisms with controlling behavior systems emphasized the development of multi-layered actions, which established the Constrained Generating Behaviors (CGBs). Corresponding these multi layers of actions with internal and external levels of constraints confirmed that the development of morphodynamics, which is associated with fabricational morphogenesis, environment effectiveness, and performative aspects were determined by the bilateral effects of these constraints on the micro-behaviors. Therefore, behavioral libraries were developed to address the adaptation and adjustment of agents with the intrinsic and extrinsic properties of morphodynamic developments.

Analogous to ethological morphogenesis, the intrinsic and extrinsic properties of morphodynamic developments provided insight into the agents' agency with their tendencies towards material systems and fabrication tools. The intrinsic properties indicated that the internal features of the agents' agency, such as the geometrical definition of the agents' morphology, required behavioral libraries to maintain the main principles of geometrical structures. It was found that the agents' morphology could be determined by derivative or aggregative procedures, through which the interaction among agents was inferred from a geometrical definition of material agencies and the physical laws that govern the fabrication agencies. It was elaborated that the consistencies of the agents' morphology relied on inhibitory mechanisms to adjust driven forms with their basis geometrical definitions. Furthermore, the inhibiting mechanisms furthered the agents' agency to self-organize the geometrical structure of agents against any deviations imposed from internal or external factors.

In contrast to the intrinsic properties, the extrinsic properties corresponded to the external influences of the morphological studies on agents' behaviors. The emphasis on fabrication processes was extended to the agents' agency from internal features to external factors. The external factors advanced agents to perceive information from external entities, such as their relevant environment and to act upon the processed information. The study of agents without specific morphology provided support for developing fabricator agents, which were operators on the environment. Consistent with this level of agency, agents could adapt their behaviors to environmental effectiveness and performative criteria, which was indirectly determined by users. It was argued that a parallel implementation of both of these levels required synchronization mechanisms to sequence the execution of internal mechanisms and external systems.

7.4 Outlook and Trends

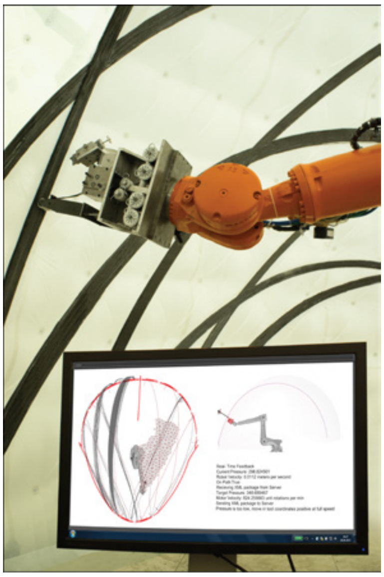


Fig. 7.4.1: The cyber-physical system, research pavilion 2014-15; source: ICD/ITKE University of Stuttgart.

The ongoing studies on behavioral strategies for design computation and construction postulate the development of agent-based systems. This research attempts to open a dialogue for behavioral integration. In the context of "Industry 4.0," the development of cyber-physical systems require novel methods to encode the real world with virtual models (Figure 7.4.1). In particular, there is growing interest in the investigation of material behaviors and manufacturing processes that coalesce design computation through distributed and decentralized units. Individual and collective units reflect the complex behavioral adaptations between computational agents and physical industrial agents. In the realm of computational agents, the developed experiments and case studies present agent-based systems to further behavioral strategies in the building industries. This study indicates that agent-based systems have the capacity to adopt building industries with the fourth industrial revolution through behavioral strategies.

In the context of behavioral strategies, developing applications via agent-based systems involves both constituent units of fabrication processes (material agencies) and constructor units, which participate directly in manufacturing processes (fabrication agencies). Material agencies and fabrication agencies highlight two important trends in construction industry. The development of applications for material tendencies were exhibited in agent assemblies and plate-like agents, wherein modulating raw materials requires adaptation with fabrication tools. In contrast, the fabrication tendencies, which were demonstrated in fibrous-like agents, signify the importance of manufacturing tools, through which material systems follow fabrication processes to erect the structures and forms. Deliberate transition between these tendencies are accompanied by analytical mechanisms that determine the producibility of virtual designs. The significance of these trends becomes notable when design computation and construction are equipped with cyber-physical systems. The development of cyber-physical systems as advanced soft systems blurs the distinction between virtual and physical systems.

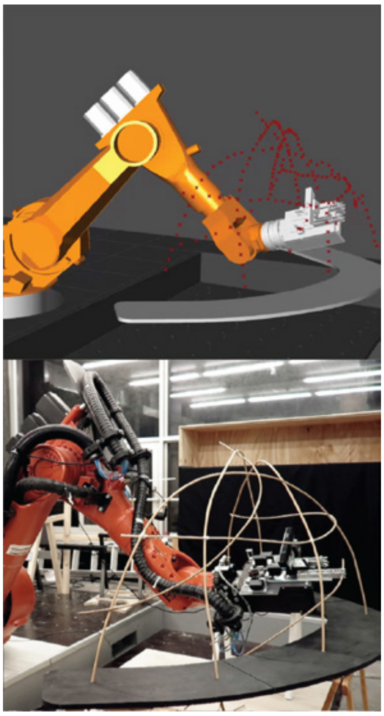


Fig. 7.4.2: The woven structure developed through the behavioral fabrication process; source: Brugnaro (2015).

This transition is consistent with distributed networks of intelligent units that interrelate the micro-behaviors of each agency with the macro-regularities of design intentions. In a cybernetic manner, computational agents could merge with materials and robotic tools to enhance manufacturing processes. In intelligent manufacturing, negotiating between virtual and real construction through agent-based systems fosters real time construction processes by advancing efficiency in cost, sustainability, and a reduction of the manufacturing time. In addition, the process of human-machine interaction within cyber-physical manufacturing expects to benefit from semantic technologies by applying augmented realities (AR) and virtual realities (VR). In that case, the established network of real and virtual agents actively informs digital design with the physical constraints of materials and robotic fabrications, in which the mediators dynamically generate a catalog of novel solutions to achieve the design goals.

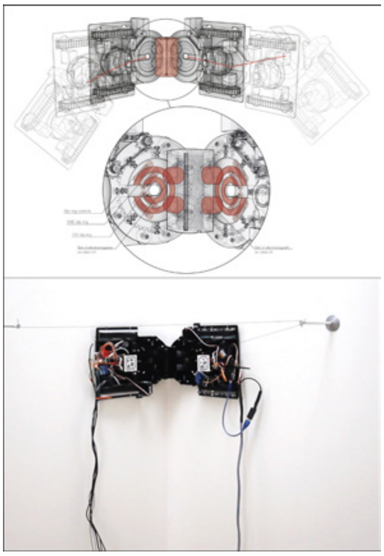


Fig. 7.4.3: The filament structures developed via overlapping virtual and physical agents; source: Yablonina (2015).

Coalescing semantic and syntactic systems could demonstrate the potential of agent-based systems for developing future applications that link the simulated world to the real world. Utilizing these applications, such as the ones partially developed in ICD/ITKE research pavilion 2014-15, will advance the fabrication process with smart methods. The study of online controlling fabrication processes transits the computational process of simulating and modeling to the real-time modeling. Although this process is in its early stages, the research developed under the author's supervision represents the high potential of this method in manufacturing processes. The studies were developed to simulate and fabricate design intentions online through industrial robotic agents and (semi-) autonomous mobile agents. Under the author's supervision, one of these studies, developed by Giulio Brugnaro (2015), investigated woven structures through online controlling mechanisms, through the behavior of computational agents in real-time, determined the movements of end-effectors (Figure 7.4.2).

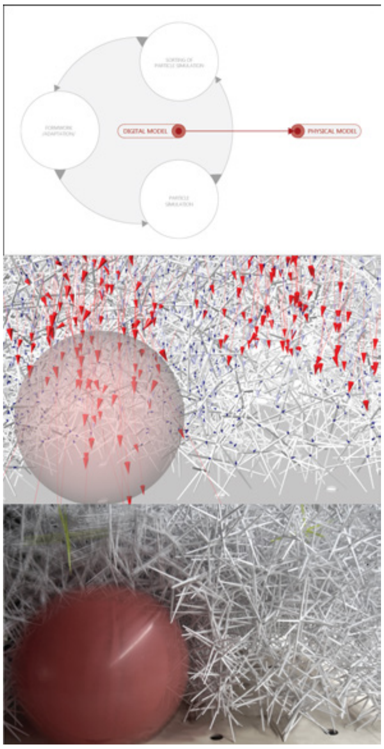


Fig. 7.4.4: The transfer of a computational simulation to a physical model; source: Rusenova (2015).

In another study, the development of mobile end-effectors as a fabricator was of interest. The research developed by Maria Yablonina (2015) demonstrated overlaying virtual reality with physical reality to reflect controlling computational agents with mobile agents (Figure 7.4.3). Another research developed by Gergana Rusenova (2015) investigated controlling mechanisms to adapt the behavior of physical models through computational simulation (Figure 7.4.4). These three studies, along with other experiences that were developed under the author's supervision at the ITECH master thesis (2014-16), proved the greatest potential of extending computational agents with behavioral controlling mechanisms to the industrial agents. Eventually, further investigation on behavioral aspects in design computation and construction will be embraced within industrial constructions.

Glossary

Agent: as an individual computational unit, has abstracted characteristics and behavioral rules, which exhibit autonomous and adaptable features (Holland 1995; Wooldridge and Jennings 1995; Casti 1997b; Gilbert 2008).

Agent-based modeling: is an approach to modeling that is based on a set of computational agents that interact with each other and with the environment (Epstein and Axtell 1996; Axelrod 1997; Epstein 1999; Bonabeau 2002; Gilbert 2008).

Behavior-based system: emphasizes modeling building blocks with some competences that "behave" in problem domains (Maes 1993).

Complex adaptive system: is a "complex system" with a network of rule-based units or agents, in which their interaction steer the complexity of the system towards adaptation (Holland 1992, 2006).

Complex system: is a system consisting of several elements that interact with each other in such a way that the sum of the elements is beyond the individual element (Simon 1962).

Constrained generating procedure: is a dynamic model that sets mechanisms to generate possibilities and procedures to limit them (Holland 2000).

Constructional morphology: is concerned with the different criteria that are involved in the process of generating constructible forms, such as "morphogenetic" (or "fabricational"), "functional," and "phylogenetic" (or "traditional") aspects (Seilacher 1970, 1973, 1991a).

Disorganized complexity: is a system with a large number of interrelated elements, that their overall behavior is predictable through statistical techniques (Weaver 1948; Miller and Page 2007).

Emergence: is the irreducible phenomena that the global properties of a system are not reducible to individual properties of its subsystems (Heylighen 1989).

Generative system: is a system that produces holistic behaviors, in which the generation of these behaviors relies on the elements within a system, their interactions, and the way they interact (Alexander 1968).

Knowledge-based system: emphasizes modeling a system with an overall knowledge of the problem domain (Maes 1993).

Modeling relation: is a process of encoding a natural system to a formal logical system through symbolic logics (Casti 1994).

Morphodynamics: as a dynamic morphology, exhibits an interplay among "functional," "fabricational," and "environmental effectiveness," which is connected to the "historical development" of an organism (Seilacher 1991a,b; Seilacher and Gishlick 2014).

Morphogenesis: is the process of growth and "the development of pattern and form in living [organisms]" (Murray 1990).

Morphospace: is a "morphological space" that exhibits the description and relation of a phenotypical definition of an organism (Mitteroecker and Huttegger 2009).

Organized complexity: is a system with a finite number of interrelated elements, that their overall behavior is not predictable through statistical techniques (Weaver 1948; Miller and Page 2007).

Self-organization or autogenesis: is a process of self-regulating within an open system (Anderson 1999).

Soft system: is a flexible system that adapts to its changing environment through feedback loops and internal regulating mechanisms (Kwinter 1993).

Theoretical morphology: studies the development of forms in two steps: first, by mathematically modeling and simulating a morphogenesis, and then, by developing a "hypothetical morphospace" to analyze and evaluate the generated form (McGhee 1999).

References

Addington, D. M. and Schodek, D. L. (2005). *Smart Materials and New Technologies: For the Architecture and Design Professions*. Architecture & design. Elsevier, Amsterdam and Heidelberg.

Alexander, C. (1964). *Notes on the Synthesis of Form*. Harvard University Press, Cambridge, MA, 7 edition.

Alexander, C. (1968). Systems generating systems. *Architectural Design*, 7(6):605–610.

Alvarez, M. and Martinez, E. (2015). *Explorations in ant path optimization*. Computational Design Seminar, Institute for Computational Design, University of Stuttgart.

Anderson, P. (1999). Perspective: complexity theory and organization science. *Organization Science*, 10(3):216–232. doi:10.1287/orsc.10.3.216.

Anzalone, P. and Clarke, C. (2003). Architectural applications of complex adaptive systems. In Klinger, K. R., editor, *Proceedings of the 2003 Annual Conference of the Association for Computer Aided Design In Architecture*, pages 325–335, Muncie, Ind. Ball State University.

Anzalone, P. and Clarke, C. (2004). The siren's call. In Beesley, P., Cheng, N. Y.-W., and Williamson, R. S., editors, *Fabrication: Examining the Digital Practice of Architecture*, pages 150–161, Toronto, Ontario. University of Waterloo School of Architecture Press.

Arkin, R. C. (1998). *Behavior-Based Robotics*. Intelligent robots and autonomous agents. MIT Press, Cambridge, Mass.

Atmosukarto, I., Luping Zhou, Wee Kheng Leow, and Zhiyong Huang (2001). Polygonizing non-uniformly distributed 3D points by advancing mesh frontiers. In *Computer Graphics International 2001*, pages 175–182. doi:10.1109/CGI.2001.934672.

Axelrod, R. (1997). Advancing the art of simulation in the social sciences. In Fandel, G., Trockel, W., Conte, R., Hegselmann, R., and Terna, P., editors, *Simulating Social Phenomena*, volume 456 of *Lecture Notes in Economics and Mathematical Systems*, pages 21–40. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-662-03366-1_2.

Baharlou, E. and Menges, A. (2013a). Behavioural Prototyping: an approach to agent-based computational design driven by fabrication characteristics and material constraints. In Gengnagel, C., Kilian, A., Nembrini, J., and Scheurer, F., editors, *Rethinking Prototyping*, pages 291–303.

Baharlou, E. and Menges, A. (2013b). Generative agent-based design computation: Integrating material formation and construction constraints. In Stouffs, R. and Sariyildiz, S., editors, *eCAADe 2013: Proceedings of the 31st International Conference on Education and Research in Computer Aided Architectural Design in Europe*, volume 2, pages 165–174.

Baharlou, E. and Menges, A. (2015). Toward a Behavioral Design System: An Agent-Based Approach for Polygonal Surfaces Structures. In Combs, L. and Perry, C., editors, *Computational Ecologies: Proceedings of the 35th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, pages 161–172. University of Cincinnati, Cincinnati OH.

Ball, P. (2007). Social science goes virtual. *Nature*, 448(7154):647–648. doi:10.1038/448647a.

Beardon, A. F., Dubejko, T., and Stephenson, K. (1994). Spiral hexagonal circle packings in the plane. *Geometriae Dedicata*, 49(1):39–70.

Björk, B.-C. (1989). Basic structure of a proposed building product model. *Computer-Aided Design*, 21(2):71–78. doi:10.1016/0010-4485(89)90141-3.

Bletzinger, K.-U. and Reitinger, R. (1991). Shape optimization of shells. In Belger, F., editor, *Natural Structures*, Proceedings of the ... symposium of the Sonderforschungsbereich 230, pages 126–131, Stuttgart. Geschäftsstelle, Univ.

Bonabeau, E. (1997). From classical models of morphogenesis to agent-based models of pattern formation. *Artificial Life*, 3(3):191–211.

Bonabeau, E. (2002). Agent-based modeling: methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3):7280–7287.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York.

Boulding, K. E. (1968). General systems theory: the skeleton of science. In Buckley, W. F., editor, *Modern Systems Research for the Behavioral Scientist*, pages 3–10. Aldine, Chicago.

Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*. MIT Press, Cambridge, Mass., 1st mit press pbk. ed. edition.

Brampton, C. J. and Kim, H. A. (2013). Optimization of tow steered fiber orientation using the level set method. In *10th World Congress on Structural and Multidisciplinary Optimization*.

Brodey, W. M. (Autumn 1967). The design of intelligent environments: soft architecture. *Landscape*, 17(1):8–12.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1):14–23. doi:10.1109/JRA.1986.1087032.

Brooks, R. A. (1990). Elephants don't play chess. *Robotics and Autonomous Systems*, 6(1-2):3–15. doi:10.1016/S0921-8890(05)80025-9.

Brooks, R. A. (1991). Intelligence without Reason. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence -*

Volume 1, IJCAI'91, pages 569–595, Sydney, New South Wales, Australia. Morgan Kaufmann Publishers Inc.

Brugnaro, G. (2015). *Robotic Softness: Behavioural Fabrication Process of a Woven Space*. M. Sc. thesis, Institute for Computational Design and Construction (ICD), University of Stuttgart.

Bryson, J. J. (2003). The behavior-oriented design of modular agent intelligence. In Goos, G., Hartmanis, J., van Leeuwen, J., Carbonell, J. G., Siekmann, J., Kowalczyk, R., Müller, J. P., Tianfield, H., and Unland, R., editors, *Agent Technologies, Infrastructures, Tools, and Applications for E-Services*, Lecture Notes in Computer Science, pages 61–76. Springer Berlin Heidelberg, Berlin, Heidelberg.

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2003). *Self-organization in Biological Systems*. Princeton studies in complexity. Princeton University Press, Princeton, N.J. and Oxford.

Carpo, M. (2011). *The Alphabet and the Algorithm*. Writing architecture. The MIT Press, Cambridge, Mass.

Carranza, P. M. and Coates, P. (2000). Swarm Modelling: The use of swarm intelligence to generate architectural form. *Proceedings of the 3rd International Conference on Generative Art, Milan (GA2000)*.

Casti, J. L. (1994). *Complexification: Explaining a Paradoxical World through the Science of Surprise*. HarperCollins, New York, NY.

Casti, J. L. (1997a). *Would-be Worlds: How Simulation is Changing the Frontiers of Science*. Wiley, New York and Chichester and Weinheim.

Casti, J. L. (1997b). Would-be worlds: Toward a theory of complex systems. *Artificial Life and Robotics*, 1(1):11–13. doi:10.1007/BF02471105.

CECA (2004). *form_script.workshop*. Retrieved April 2016, from https://www.uel.ac.uk/schools/ace/research/xyz{_}constructed-environments-and-critical-architecture/resources.

Chen, C.-H. (2009a). A prototype using multi-agent based simulation in spatial analysis and planning. In *Between Man and Machine*, pages 513–521.

Chen, C.-H. (2009b). An agent-based street simulator. In Tidafi, T. and Dorta, T., editors, *Joining Languages, Cultures and Visions*, pages 833–846.

Chen, C.-H. and Chiu, M.-l. (2006). SCALE: a street case library for environmental design with agent interfaces. In van Leeuwen, J. and Timmermans, H. J. P, editors, *Innovations in Design & Decision Support Systems in Architecture and Urban Planning*, pages 137–150. Springer, Dordrecht.

Chen, H.-S. and Lin, F.-T. (2003). A simulation study on public building's staircase fault tolerance. In Chiu, M.-l., editor, *CAAD Futures - Proceedings of the 10th International Conference on Computer Aided Architectural Design Futures*, pages 393–402, Tainan (Taiwan).

Chilton, J. (2000). *Heinz Isler*. The engineer's contribution to contemporary architecture. Thomas Telford, London.

Clarke, C. and Anzalone, P. (2004). Trusset: parallel development of software and construction systems for space-truss structures. *International Journal of Architectural Computing*, 2(2):229–244. doi:10.1260/1478077041518700.

Coates, P. (2010). *Programming.architecture*. Routledge, London, New York.

Coates, P. and Schmid, C. (1999). Agent based modelling. In Brown, A., Knight, M., and Berridge, P., editors, *eCAADe Conference Proceedings*, eCAADe, pages 652–661. doi:10.1016/j.trstmh.2009.01.006.

Coates, P. S. (2004). Review paper: some experiments using agent modelling at CECA. In Soddu, C., editor, *Generative Art 2004*, Italy.

Cohen, J. and Stewart, I. (1994). *The Collapse of Chaos: Discovering Simplicity in a Complex World*. Viking, New York, NY, 1. publ edition.

Cummings, F. W. (1985). A pattern-surface interactive model of morphogenesis. *Journal of Theoretical Biology*, 116(2):243–273. doi:10.1016/S0022-5193(85)80267-8.

Cummings, F. W. (1989). On surface geometry coupled to morphogen. *Journal of Theoretical Biology*, 137(2):215–219. doi:10.1016/S0022-5193(89)80207-3.

Cummings, F. W. (1990). A model of morphogenetic pattern formation. *Journal of Theoretical Biology*, 144(4):547–566. doi:10.1016/S0022-5193(05)80089-X.

Cutler, B. and Whiting, E. (2007). Constrained planar remeshing for architecture. In Healey, C. G. and Lank, E., editors, *Graphics Interface*, pages 11–18. http://portal.acm.org/citation.cfm?doid=1268517.1268522.

De Wolf, T. and Holvoet, T. (2005). Emergence versus self-organisation: different concepts but promising when combined. In Brueckner, S. A., editor, *Engineering Self-Organising Systems*, volume 3464 of *Lecture Notes in Computer Science*, pages 1–15. Springer, Berlin [u.a.].

Debout, P., Chanal, H., and Duc, E. (2011). Tool path smoothing of a redundant machine: application to automated fiber placement. *Computer-Aided Design*, 43(2):122–132. doi:10.1016/j.cad.2010.09.011.

DeLanda, M. (1997). The Machinic Phylum. In Brouwer, J. and Hoekendijk, C., editors, *Technomorphica*. V2 Organisatie, Rotterdam, the Netherlands. http://v2.nl/archive/articles/the-machinic-phylum.

DeLanda, M. (2001). Philosophies of design: the case of modeling software. In Salazar, J., editor, *Verb Processing*, pages 132–142. Actar, Barcelona.

DeLanda, M. (2010). Multi-agent systems. In Leach, N. and Snooks, R., editors, *Swarm Intelligence*.

Deleuze, G. and Guattari, F. (1987). *A Thousand Plateaus: Capitalism and Schizophrenia*. University of Minnesota Press, Minneapolis.

Dino Viehland (2014). *IronPython*. Jim Hugunin. www.ironpython.net.

Doumpioti, C. (2008). Adaptive growth of fibre composite structures. In Kudless, A., Oxman, N., and Swackhamer, M., editors, *Silicon + Skin: Biological Processes and Computation*, pages 300–307. ACADIA.

Durmazoglu, M. Ç., Çagdas, G., and Tong, H. (2008). An agent based approach for evaluation of free-form surfaces. In *Architecture in Computro*, pages 405–412, Antwerpen, Belgium.

Eastman, C. (2009). Essay: what is BIM? In Miyauchi, T., Tsukui, N., Nishimaki, A., and Yokoyama, K., editors, *Architectural Transformations via BIM*, Architecture and urbanism Special issue, pages 16–17. a+u Publishing Co, Tokyo.

Eastman, C., Teicholz, P., Sacks, R., and Liston, K. (2011). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. Wiley, Hoboken, NJ, 2. ed edition.

Eastman, C. M. (1999). *Building Product Models: Computer Environments Supporting Design and Construction*. CRC Press, Boca Raton, Fla.

Eble, G. J. (2000). Theoretical morphology: state of the art. *Paleobiology*, 26(3):520–528. doi:10.1666/0094-8373(2000)026<0520:TMSOTA>2.0.CO;2.

Eble, G. J. (2003). Developmental morphospaces and evolution. In Crutchfield, J. P. and Schuster, P., editors, *Evolutionary Dynamics*, pages 33–63. Oxford University Press, USA.

Eleni, P., Turner, A., and Thum, R. (2002). Interacting unities: an agent-based system. In Soddu, C., editor, *Generative Art 2002*, pages 23.1–23.13, Politecnico di Milano University, Italy.

Ellers, O. (1993). A mechanical model of growth in regular sea urchins: predictions of shape and a developmental morphospace. *Proceedings of the Royal Society B: Biological Sciences*, 254(1340):123–129. doi:10.1098/rspb.1993.0136.

Epstein, J. M. (1999). Agent-based computational models and generative social science. *Complexity*, 4(5):41–60. doi:10.1002/(SICI)1099-0526(199905/06)4:5<41::AID-CPLX9>3.0.CO;2-F.

Epstein, J. M. (2006). *Generative Social Science: Studies in Agent-Based Computational Modeling*. Princeton studies in complexity. Princeton University Press, Princeton.

Epstein, J. M. (2008). Why model? *Journal of Artificial Societies and Social Simulation*, 11(4):12. http://jasss.soc.surrey.ac.uk/11/4/12.html.

Epstein, J. M. and Axtell, R. (1996). *Growing Artificial Societies: Social Science from the Bottom Up*. Complex Adaptive Systems. Brookings Institution Press, Washington D.C.

Fatah gen. Schieck, A., Penn, A., Mottram, C., Strothmann, C., Ohlenburg, J., Broll, W., and Aish, F. (2004). Interactive space generation through play: exploring form creation and the role of simulation on the design table. In Rüdiger, B., Tournay, B., and Ørbæk, H., editors, *Architecture in the Network Society*, pages 148–158, Copenhagen, Denmark. Kunstakademiets Arkitektskole.

Franklin, S. and Graesser, A. (1997). Is it an agent, or just a program? a taxonomy for autonomous agents. In Müller, J. P., Wooldridge, M. J., and Jennings, N. R., editors, *Intelligent Agents III: ATAL 1996.*, volume 1193 of *Lecture Notes in Computer Science*, pages 21–35. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/BFb0013570.

Frazer, J. (1995). *An Evolutionary Architecture*, volume 7 of *Themes*. Architectural Association, London.

Gardner, M. (1970). Mathematical games: the fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, 223:120–123.

Genesereth, M. R. and Ketchpel, S. P. (1994). Software agents. *Communications of the ACM*, 37(7):48 – ff. doi:10.1145/176789.176794.

Gerber, D. J., Shiordia, R., Veetil, S., and Mahesh, A. (2014). Design agency: prototyping multi-agent system simulation. In *Proceedings of the Symposium on Simulation for Architecture & Urban Design*, SimAUD '14, pages 24:1–24:8, San Diego, CA, USA. Society for Computer Simulation International. http://dl.acm.org/citation.cfm?id=2664323.2664347.

Gero, J. S. and Nath, G. (1997). Formalising situated learning in computer-aided design. In Junge, R., editor, *CAAD futures 1997*, pages 801–808, Dordrecht and London. Kluwer Academic.

Gero, J. S. and Peng, W. (2004). A situated agent-based design assistant. In Lee, H. S. and Choi, J. W., editors, *CAADRIA*, pages 145–158, Seoul, Korea. Yonsei University Press.

Gershenson, C. (2007). *Design and Control of Self-organizing Systems*. CopIt ArXives.

Gilbert, G. N. (2008). *Agent-Based Models*, volume 07-153 of *Quantitative applications in the social sciences*. SAGE Publications, Los Angeles.

Gilbert, G. N. and Troitzsch, K. G. (2005). *Simulation for the Social Scientist*. Open University Press, Maidenhead, 2 edition.

Goethe, J. W. v. (1988). Scientific studies. In Miller, D., editor, *The Collected Works, Vol. 12.* Suhrkamp Publishers New York, Inc.

Goethe, J. W. v. (1989). *Goethe's Botanical Writings: Translated by Bertha Mueller*. OX BOW Press, Woodbridge.

Grassé, P.-P. (1959). La reconstruction du nid et les coordinations interindividuelles chezBellicositermes natalensis etCubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, 6(1):41–80. doi:10.1007/BF02223791.

Grimm, V. (1999). Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future? *Ecological Modelling*, 115(2-3):129–148. doi:10.1016/S0304-3800(98)00188-4.

Grimm, V. and Railsback, S. F. (2005). *Individual-Based Modeling and Ecology*. Princeton Series in Theoretical and Computational Biology. Princeton University Press.

Grimm, V., Wyszomirski, T., Aikman, D., and Uchmański, J. (1999). Individual-based modelling and ecological theory: synthesis of a workshop. *Ecological Modelling*, 115(2-3):275–282. doi:10.1016/S0304-3800(98)00186-0.

Hao, H. and Jia, T.-L. (2010). Floating bubbles: an agent-based system for layout planning. In *New Frontiers*, pages 175–183, Hong Kong.

Helbing, D. (2012). Agent-based modeling. In Helbing, D., editor, *Social Self-Organization*, Understanding Complex Systems, pages 25–70. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-642-24004-1_2.

Hensel, M. (2010). Performance-oriented architecture: towards a biological paradigm for architectural design and the built environment. *FORMakademisk*, 3(1):36–56. http://hdl.handle.net/11250/92973.

Hensel, M. U. (2009). Heterogeneous materials and variable behaviour: potentials for the design disciplines. *Nordes 2009 - Engaging Artifacts Conference, Oslo School of Architecture and Design*, (3).

Hensel, M. U. (2011). Performance-oriented architecture and the spatial and material organisation complex: rethinking the definition, role and performative capacity of the spatial and material boundaries of the built environment. *FORMakademisk*, 4(1). doi:10.7577/formakademisk.125.

Hensel, M. U. and Menges, A. (2008). Inclusive performance: efficiency versus effectiveness: towards a morpho-ecological approach for design. *Architectural Design*, 78(2):54–63. doi:http://dx.doi.org/10.1002/ad.642.

Hensel, M. U., Menges, A., and Weinstock, M. (2010). *Emergent Technologies and Design*. Routledge, Oxon U.K. and New York NY.

Heylighen, F. (1989). Self-organization, emergence and the architecture of complexity. In *Proceedings of the 1st European Conference on System Science*, volume 18, pages 23–32.

Hickman, C. S. (1993). Theoretical design space: a new paradigm for the analysis of structural diversity. *Neues Jahrbuch für Geologie und Paläontologie*, 190:169–182.

Höbinger, M. (2009). *Packing Circles and Spheres on Surfaces*. Diploma thesis, Discrete Mathematics and Geometry.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, Cambridge and Mass, 1 edition.

Holland, J. H. (1995). *Hidden Order: How Adaptation Builds Complexity*. Addison-Wesley, Reading and Mass.

Holland, J. H. (2000). *Emergence: From Chaos to Order*. Perseus Books, Cambridge and Mass, 3 edition.

Holland, J. H. (2006). Studying complex adaptive systems. *Journal of Systems Science and Complexity*, 19(1):1–8. doi:10.1007/s11424-006-0001-z.

Holland, J. H. (2010). Excavating emergence. In Leach, N. and Snooks, R., editors, *Swarm Intelligence*.

Holland, J. H. and Miller, J. H. (1991). Artificial adaptive agents in economic theory. *The American Economic Review*, 81(2):365–370. http://www.jstor.org/stable/2006886.

Holzer, D. (2007). Are you talking to me? why BIM alone is not the answer. In Orr, K. and Kaji-O'Grady, S., editors, *Fourth International Conference of the Association of Architecture Schools of Australasia*, Sydney, Australia. University of Technology, Sydney.

Ireland, T. (2008). Space diagrams. In *Architecture in Computro*, pages 91–98, Antwerpen, Belgium.

Ireland, T. (2009). Emergent space diagrams: the application of swarm intelligence to the problem of automatic plan generation. In Tidafi, T. and Dorta, T., editors, *Joining Languages, Cultures and Visions*, pages 245–258.

Ireland, T. (2010). Stigmergic planning. In Sprecher, A., Yeshayahu, S., and Lorenzo-Eiroa, P., editors, *LIFE in:formation, On Responsive Information and Variations in Architecture*, pages 183–189, New York. ACADIA.

Jennings, N. R. (2000). On agent-based software engineering. *Artificial Intelligence*, 117(2):277–296. doi:10.1016/S0004-3702(99)00107-1.

Jeronimidis, G. (2000). Structure-property relationships in biological materials. In Elices, M., editor, *Structural Biological Materials*, Pergamon materials series, pages 3–16. Pergamon, Amsterdam.

Jeronimidis, G. (2004). Biodynamics. In Hensel, M. U., Menges, A., and Weinstock, M., editors, *Emergence*, Profile, pages 90–95. Wiley-Academy, Chichester.

Jeronimidis, G. (2008). Bioinspiration for engineering and architecture: materials—structures—function. In Kudless, A., Oxman, N., and Swackhamer, M., editors, *Silicon + Skin: Biological Processes and Computation*, pages 26–32. ACADIA.

Karunakaran, A. (2005). Organisation of pedestrian movements: an agent-based approach. In *Digital Opportunities*, pages 305–313, New Delhi, India.

Kauffman, S. A. (1993). *The origins of order: Self-organization and selection in evolution*. Oxford Univ. Pr, New York.

Kieran, S. and Timberlake, J. (2004). *Refabricating Architecture: How Manufacturing Methodologies are Poised to Transform Building Construction*. McGraw-Hill, New York.

Kitazawa, K. and Batty, M. (2004). Pedestrian behaviour modelling: an application to retail movements using a genetic algorithm. In van Leeuwen, Jos P and Timmermans, Harry J. P, editors, *Developments in Design & Decision Support Systems in Architecture and Urban Planning*. Dept. of Architecture, Building and Planning, Eindhoven University of Technology, The Netherlands.

Klinger, K. R. (2008). Relations: information exchange in designing and making architecture. In Kolarevic, B. and Klinger, K. R., editors, *Manufacturing Material Effects*, pages 25–36. Routledge, New York.

Kolarevic, B. (2003). Digital morphogenesis. In Kolarevic, B., editor, *Architecture in the Digital Age*, pages 12–28. Spon Press, New York.

Kolarevic, B. (2008). Architecture in the post-digital age: towards integrative design. In *Architecture in Computro*, pages 653–658, Antwerpen, Belgium.

Kolarevic, B. (2009). Towards integrative design. *International Journal of Architectural Computing*, 7(3):335–344. doi:10.1260/147807709789621248.

Krause, J. C. S. (1996). *Interactive agent generated architecture*. M. Sc. Arch. thesis, The Massachusetts Institute of Technology.

Krause, J. C. S. (1997). Agent generated architecture. In *Design and Representation*.

Kuo, C.-M., Takahashi, K., and Chou, T.-W. (1988). Effect of fiber waviness on the nonlinear elastic behavior of flexible composites. *Journal of Composite Materials*, 22(11):1004–1025. doi:10.1177/002199838802201101.

Kwinter, S. (1993). Soft systems. In Boigon, B., editor, *Culture Lab 1*, pages 207–228. Princeton Architectural Press, New York.

Kwinter, S. (2003). The computational fallacy. *Thresholds - Denatured*, 26:90–92.

Kwinter, S. (2008). Who is afraid of formalism. In Davidson, C. C., editor, *Far From Equilibrium*, pages 144–149. Actar, New York.

Leach, N. (2009). Digital morphogenesis. *Architectural Design*, 79(1):32–37. doi:http://dx.doi.org/10.1002/ad.806.

Li, J. and Wilensky, U. (2009). *NetLogo Sugarscape 3 Wealth Distribution model*. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. http://ccl.northwestern.edu/netlogo/models/Sugarscape3WealthDistribution.

Liew, P. S. and Gero, J. S. (2002a). A memory system for a situated design agent based on constructive memory. In Mohamed, A. R., editor, *CAADRIA 2002*, pages 199–206, Selangor, Malaysia. Prentice Hall Pearson Malaysia.

Liew, P.-S. and Gero, J. S. (2002b). An implementation model of constructive memory for a design agent. In Gero, J. S. and Brazier, F. M., editors, *Agents in Design*, pages 257–276.

Lindenmayer, A. (1968). Mathematical models for cellular interactions in development II. Simple and branching filaments with two-sided inputs. *Journal of Theoretical Biology*, 18(3):300–315. doi:10.1016/0022-5193(68)90080-5.

Lindsey, B. (2001). *Digital Gehry: Material Resistance, Digital Construction*. The IT revolution in architecture. Birkhäuser, Basel and Boston.

Liu, Y., Pottmann, H., Wallner, J., Yang, Y.-L., and Wang, W. (2006). Geometric modeling with conical meshes and developable surfaces. In *ACM SIGGRAPH 2006 Papers on - SIGGRAPH '06*, pages 681–689. doi:10.1145/1179352.1141941.

Luping, Z. (2001). *3D Surface Reconstruction by Tangent Plane Intersection*. M. Sc. thesis, Computer Science.

Macal, C. M. and North, M. J. (2009). Agent-based modeling and simulation. *Proceedings of the 2009 Winter Simulation Conference (WSC)*, pages 86–98. doi:10.1109/WSC.2009.5429318.

Macal, C. M. and North, M. J. (2010). Tutorial on agent-based modelling and simulation. *Journal of Simulation*, 4(3):151–162. doi:10.1057/jos.2010.3.

Maes, P. (1993). Behavior-based artificial intelligence. In Meyer, J.-A., Wilson, S. W., and Roiblat, H. L., editors, *From Animals to Animats 2*, pages 2–10. MIT Press, Cambridge and Mass. [u.a.].

Manahl, M., Stavric, M., and Wiltsche, A. (2012). Ornamental discretisation of free-form surfaces. *International Journal of Architectural Computing*, 10(4):595–612. doi:10.1260/1478-0771.10.4.595.

Matarić, M. J. and Michaud, F. (2008). Behavior-based systems. In Siciliano, B. and Khatib, O., editors, *Springer Handbook of Robotics*, pages 891–909. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-540-30301-5_39.

Mattheck, C. (1998). *Design in Nature: Learning from Trees*. Springer, Berlin and Heidelberg.

Mattheck, C., Burkhardt, S., and Erb, D. (1991). Shape optimization of engineering components by adaptive biological growth. In Eschenauer, H. A., Mattheck, C., and Olhoff, N., editors, *Engineering Optimization in Design Processes*, volume 63 of *Lecture Notes in Engineering*, pages 15–24. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-642-84397-6_2.

McGhee, G. R. (1991). Theoretical morphology: the concept and its applications. In Gilinsky, N. L. and Signor, P. W., editors, *Analytical Paleobiology*, Short Courses in Paleontology, pages 87–102. the Paleontological Society and the University of Tennessee, Knoxville.

McGhee, G. R. (1999). *Theoretical Morphology: The Concept and its Applications*. Perspectives in paleobiology and earth history. Columbia University Press, New York.

McGhee, G. R. (2006). *The Geometry of Evolution: Adaptive Landscapes and Theoretical Morphospaces*. Cambridge University Press, Cambridge.

Menges, A. (2008). Integral formation and materialization: computational form and material gestalt. In Kolarevic, B. and Klinger, K. R., editors, *Manufacturing Material Effects*, pages 195–210. Routledge, New York.

Menges, A. (2010). Form generation and materialization at the transition from computer-aided to computational design. *Detail*, 2010(04):330–335.

Menges, A. (2013). Morphospaces of robotic fabrication: from theoretical morphology to design computation and digital fabrication in architecture. In Brell-Cokcan, S. and Braumann, J., editors, *Rob/Arch 2012*, pages 28–47, Wien. Springer.

Meyer, J.-A. and Guillot, A. (1991). Simulation of Adaptive Behavior in Animats: Review and Prospect. In Meyer, J.-A. and Wilson, S. W., editors, *From animals to animats*, pages 2–14. MIT Press, Cambridge, Mass.

Microsoft (2015). *Visual Basic .NET*. https://msdn.microsoft.com/en-us/vstudio/hh388573.aspx.

Miller, J. H. and Page, S. E. (2007). *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*. Princeton University Press, Princeton, New Jersey.

Minsky, M. L. (1988). *The Society of Mind*. A Touchstone book. Simon and Schuster, New York, 2. print edition.

Mitchell, W. J. (1990). *The Logic of Architecture: Design, Computation, and Cognition*. MIT Press, Cambridge Mass.

Mitchell, W. J. (2009). Essay: thinking in BIM. In Miyauchi, T., Tsukui, N., Nishimaki, A., and Yokoyama, K., editors, *Architectural Transformations via BIM*, Architecture and urbanism Special issue, pages 10–15. a+u Publishing Co, Tokyo.

Mitteroecker, P. and Huttegger, S. M. (2009). The concept of morphospaces in evolutionary and developmental biology: mathematics and metaphors. *Biological Theory*, 4(1):54–67. doi:10.1162/biot.2009.4.1.54.

Moss, J., Cagan, J., and Kotovsky, K. (2004). Learning from design experience in an agent-based design system. *Research in Engineering Design*. doi:10.1007/s00163-003-0042-4.

Moussaïd, M., Helbing, D., and Theraulaz, G. (2011). How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences of the United States of America*, 108(17):6884–6888. doi:10.1073/pnas.1016507108.

Murray, J. D. (1990). Discussion: Turing's theory of morphogenesis: its influence on modelling biological pattern and form. *Bulletin of Mathematical Biology*, 52(1-2):117–152. doi:10.1007/BF02459571.

Narahara, T. (2008). New methodologies in architectural design inspired by self-organization. In Kudless, A., Oxman, N., and Swackhamer, M., editors, *Silicon + Skin: Biological Processes and Computation*, pages 324–331. ACADIA.

Narahara, T. (2013a). A generative approach to robotic fabrication. In Stouffs, R. and Sariyildiz, S., editors, *eCAADe 2013: Proceedings of the 31st International Conference on Education and Research in Computer Aided Architectural Design in Europe*, pages 673–681.

Narahara, T. (2013b). Adaptive growth using robotic fabrication. In Stouffs, R., Janssen, P., Roudavski, S., and Tunçer, B., editors, *CAADRIA 2013: Conference on Computer-Aided Architectural Design Research in Asia*, pages 65–74.

Negroponte, N. (1975). *Soft Architecture Machines*. M.I.T. Press, Cambridge, Mass. and London.

Niazi, M. and Hussain, A. (2011). Agent-based computing from multi-agent systems to agent-based models: a visual survey. *Scientometrics*, 89(2):479–499. doi:10.1007/s11192-011-0468-9.

NIBS (2007). *United States National Building Information Modeling Standard, Version 1 - Part 1: Overview, Principles, and Methodologies*.

Otto, F., Rasch, B., Pfafferodt, G., Schönborn, A. G., and Schanz, S. (1995). *Frei Otto, Bodo Rasch: Finding Form: Towards an Architecture of the Minimal*. Deutscher Werkbund Bayern, Germany, 5. edition 2006 edition.

Parascho, S., Baur, M., Baharlou, E., Knippers, J., and Menges, A. (2013). Agent-based model for the development of integrative design tools. In Beesley, P., Stacey, M., and Khan, O., editors, *ACADIA 2013 Adaptive Architecture*, pages 429–430.

Penttilä, H. (2009). Services in digital design: new visions for AEC-field collaboration. *International Journal of Architectural Computing*, 7(3):459–478. doi:10.1260/147807709789621257.

Pfeifer, R. and Scheier, C. (2001). *Understanding Intelligence*. MIT Press, Cambridge and Mass, 1 edition.

Philippi, U. and Nachtigall, W. (1991). Constructional morphology of sea urchin tests. In Belger, F., editor, *Natural Structures*, Proceedings of the ... symposium of the Sonderforschungsbereich 230, pages 183–191, Stuttgart. Geschäftsstelle, Univ.

Pottmann, H. (2013). Architectural geometry and fabrication-aware design. *Nexus Network Journal*. doi:10.1007/s00004-013-0149-5.

Pottmann, H., Asperl, A., Hofer, M., and Kilian, A. (2007). *Architectural Geometry*. Bentley Institute Press, Exton, Pa., 1. ed edition.

Railsback, S. F. and Grimm, V. (2012). *Agent-Based and Individual-Based Modeling: A Practical Introduction*. Princeton University Press, Princeton.

Rao, A. S. and Georgeff, M. P. (1991). Modeling rational agents within a BDI-architecture. *KR*, 91:473–484.

Raup, D. M. and Michelson, A. (1965). Theoretical morphology of the coiled shell. *Science (New York, N.Y.)*, 147(3663):1294–1295. doi:10.1126/science.147.3663.1294.

Reeves, D. (2013). *Pack Attack*. Retrieved April 2016, from http://spatialslur.blogspot.co.uk/2013/10/pack-attack.html.

Resnick, M. (1994). *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*. Complex Adaptive Systems. MIT Press, Cambridge, Mass, 1 edition.

Reynolds, C. W. (1987). Flocks, herds and schools: a distributed behavioral model. In Stone, M. C., editor, *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '87*, pages 25–34. ACM Press. doi:10.1145/37401.37406.

Reynolds, C. W. (1999). Steering behaviors for autonomous characters. *Game Developers Conference*, 13:763–782.

Robert McNeel & Associates (2015). *Rhinoceros 3D*. Seattle, USA. www.rhino3d.com.

Rosenblueth, A., Wiener, N., and Bigelow, J. (1943). Behavior, purpose and teleology. *Philosophy of Science*, 10(1):18–24. http://www.jstor.org/stable/184878.

Rusenova, G. (2015). *Emergent Space: Simulations and Analyses of Spatial Aggregate Formations*. M. Sc. thesis, Institute for Computational Design and Construction (ICD), University of Stuttgart.

Russell, E. S. ([1916] 1982). *Form and Function: A Contribution to the History of Animal Morphology*. University of Chicago Press, Chicago.

Russell, S. J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice Hall Series in Artificial Intelligence. Prentice Hall, Upper Saddle River, NJ, 2 edition.

Rutten, D. (2005). *Construction of Individual Voronoi-Cells in 2-Dimensional Space*. Delft, The Netherlands.

Sakoda, J. M. (1971). The checkerboard model of social interaction. *The Journal of Mathematical Sociology*, 1(1):119–132. doi:10.1080/0022250X.1971.9989791.

Schelling, T. C. (1971). Dynamic models of segregation. *The Journal of Mathematical Sociology*, 1(2):143–186. doi:10.1080/0022250X.1971.9989794.

Schelling, T. C. (1978). *Micromotives and Macrobehavior*. Fels Lectures on Public Policy Analysis. Norton, New York, 1 edition.

Scheurer, F. (2003). The groningen twister: an experiment in applied generative design. In Soddu, C., editor, *Generative Art 2003*, pages 90–99.

Scheurer, F. (2005). Turning the design process downside-up: self-organization in real-world architecture. In Martens, B. and Brown, A., editors, *Computer Aided Architectural Design Futures 2005*, pages 269–278, Dordrecht. Springer.

Scheurer, F. (2007). Getting complexity organised. *Automation in Construction*, 16(1):78–85. doi:10.1016/j.autcon.2005.10.006.

Schiftner, A., Höbinger, M., Wallner, J., and Pottmann, H. (2009). Packing circles and spheres on surfaces. In Inakage, M., editor, *ACM SIGGRAPH Asia 2009 papers*. doi:10.1145/1661412.1618485.

Schwinn, T., Krieg, O. D., and Menges, A. (2014). Behavioral strategies: synthesizing design computation and robotic fabrication of lightweight timber plate structures. In Gerber, D., Huang, A., and Sanchez, J., editors, *Design Agency*, pages 177–188.

Schwinn, T., Krieg, O. D., Menges, A., Mihaylov, B., and Reichert, S. (2012). Machinic morphospaces: biomimetic design strategies for the computational exploration of robot constraint spaces for wood fabrication. In Cabrinha, M., Johnson, J. K., and Steinfeld, K., editors, *Synthetic Digital Ecologies*, pages 157–168, San Francisco: California College of the Arts. ACADIA.

Seilacher, A. (1970). Arbeitskonzept zur Konstruktions-Morphologie. *Lethaia*, 3(4):393–396. doi:10.1111/j.1502-3931.1970.tb00830.x.

Seilacher, A. (1973). Fabricational noise in adaptive morphology. *Systematic Zoology*, 22(4):451. doi:10.2307/2412952.

Seilacher, A. (1991a). Self-organizing mechanisms in morphogenesis and evolution. In Schmidt-Kittler, N. and Vogel, K., editors, *Constructional Morphology and Evolution*, pages 251–271. Springer-Verlag, Berlin and New York.

Seilacher, A. (1991b). Self-organizing morphogenetic mechanisms as processors of evolution. *Revista Espanola De Paleontologia*, n.° Extraordinario:5–11.

Seilacher, A. and Gishlick, A. D. (2014). *Morphodynamics*. CRC Press, Boca Raton.

Sharma, S. and Turner, J. A. (2004). An object oriented approach to simulating agent-based behavior. In *Proceedings of the 8th Iberoamerican Congress of Digital Graphics (SIGraDi)*.

Shea, K. (2004). Directed randomness. In Leach, N., Turnbull, D., and Williams, C., editors, *Digital Tectonics*, pages 89–101. Wiley-Academy, Chichester.

Shelden, D. R. (2002). *Digital Surface Representation and the Constructibility of Gehry's Architecture*. Doctor of Philosophy In The Field Of Architecture: Design And Computation, The Massachusetts Institute of Technology.

Shih, S.-G., Chen, Y.-H., and Hu, Shan-Ching and Lin, Ching-Yuan (2009). On the simulation of pedestrian behavior: a stochastic model based on Markov chain and information space. In Tidafi, T. and Dorta, T., editors, *Joining Languages, Cultures and Visions*.

Simon, H. A. (1962). The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482. http://www.jstor.org/stable/985254.

Simon, H. A. (1996). *The Sciences of the Artificial*. MIT Press, Cambridge and Mass, 3 edition.

Smith III, A. R. (1976). Introduction to and survey of polyautomata theory. In Lindenmayer, A. and Rozenberg, G., editors, *Automata, Languages, Development*, pages 405–422. North-Holland Pub. Co. and Sole distributors for U.S.A. and Canada, American Elsevier Pub. Co, Amsterdam and New York and New York.

Snooks, R. (2012a). *Fibrous Assemblages and Behavioral Composites*. Retrieved December 2014, from http://thefunambulist.net/2012/04/20/guest-writers-essays-25-fibrous-assemblages-and-behavioral-composites-by-roland-snooks/.

Snooks, R. (2012b). Volatile formation. *Log 25 (New York)*, pages 55–62.

Spirov, A. (1993). The change of initial symmetry in the pattern-form interaction model of sea urchin gastrulation. *Journal of Theoretical Biology*, 161(4):491–504. doi:10.1006/jtbi.1993.1069.

Squazzoni, F. (2012). *Agent-Based Computational Sociology*. Wiley & Sons, Hoboken and N.J.

Starfield, A. M., Smith, K. A., and Bleloch, A. L. (1990). *How to Model It: Problem Solving for the Computer Age*. McGraw-Hill, New York.

Stavric, M., Hirschberg, U., and Wiltsche, A. (2010). Spatializing planar ornaments: towards esthetic control in segmenting and building curved surfaces. In Schmitt, G., editor, *Future cities*, pages 437–444. vdf Hochschulverl. [u.a.], Zürich.

Stavric, M. and Wiltsche, A. (2011). Ornamental plate shell structures. In Leclercq, P., Heylighen, A., and Martin, G., editors, *Designing together*, Les Éditions de l'Université de Liège 2011. https://iam2.tugraz.at/fwf/papers/CAADFutures-OrnamentalPlateShellStructures.pdf.

Stiny, G. and Gips, J. (1971). Shape grammars and the generative specification of painting and sculpture. In *IFIP Congress (2)*, volume 2.

Sullivan, L. H. (1896). The tall office building artistically considered. *Lippincott's Magazine*, 57(3):403–409.

Sun, C., Vries, B. D., and Dijkstra, J. (2007). Measuring human behaviour using a head-cave. In Dong, A. A.-S., Vande Moere, Andrew Elzar, and Gero, J. S., editors, *Computer-aided architectural design futures (CAADFutures) 2007*, pages 501–511, Dordrecht, the Netherlands. Springer.

Sutherland, I. E. (1964). Sketch pad a man-machine graphical communication system. In *Proceedings of the SHARE Design Automation Workshop on - DAC '64*, pages 6.329–6.346. doi:10.1145/800265.810742.

Tamke, M., Riiber, J., and Jungjohann, H. (2010a). Generated lamella. In Sprecher, A., Yeshayahu, S., and Lorenzo-Eiroa, P., editors, *LIFE in:formation, On Responsive Information and Variations in Architecture*, pages 340–347, New York. ACADIA.

Tamke, M., Riiber, J., Jungjohann, H., and Thomsen, M. R. (2010b). Lamella flock. In Ceccato, C., Hesselgren, L., Pauly, M., Pottmann, H., and Wallner, J., editors, *Advances in Architectural Geometry 2010*, pages 37–48. Springer Vienna, Vienna. doi:10.1007/978-3-7091-0309-8_3.

Telford, M. (1991). Echinoids: Domes, pneus, or something else? In Belger, F., editor, *Natural Structures*, Proceedings of the ... symposium of the Sonderforschungsbereich 230, pages 103–109, Stuttgart. Geschäftsstelle, Univ.

Terzidis, K. (2003). *Expressive Form: A Conceptual Approach to Computational Design*. Spon Press, London.

Theraulaz, G. (2014). Embracing the Creativity of Stigmergy in Social Insects. *Architectural Design*, 84(5):54–59. doi:10.1002/ad.1808.

Theraulaz, G. and Bonabeau, E. (1999). A brief history of stigmergy. *Artificial Life*, 5(2):97–116. doi:10.1162/106454699568700.

Thompson, D. W. (1942). *On Growth and Form*. Dover Publications, INC., New York.

Torrens, P. M. (2010). Agent-based models and the spatial sciences. *Geography Compass*, 4(5):428–448. doi:10.1111/j.1749-8198.2009.00311.x.

Troche, C. (2008). Planar hexagonal meshes by tangent plane intersection. In Pottmann, H., Kilian, A., and Hofer, M., editors, *AAG 2008: First Symposium on Architectural Geometry Vienna, Austria September 13-16, 2008*, pages 57–60. Vienna.

Trotter, J. and Koob, T. (1989). Collagen and proteoglycan in a sea urchin ligament with mutable mechanical properties. *Cell and Tissue Research*, 258(3):527–539. doi:10.1007/BF00218864.

Tsiliakos, M. (2010). *Adaptive Materiality: Fiber Structures: Using Swarm Intelligence for Simulating Stress Driven Material Growth and Organisation in Predefined Design Scenarios*. Master of Science in Adaptive Architecture and Computation, University College London, London, UK.

Tsiliakos, M. (2012). Swarm materiality: a multi-agent approach to stress driven material organization. In Achten, H., Pavlicek, J., Hulin, J., and Matejdan, D., editors, *Digital Physicality*, pages 301–309, Czech Republic and Prague. eCAADe, Education and Research in Computer Aided Architectural Design in Europe and Czech Technical University in Prague, Faculty of Architecture (Czech Republic).

Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72. doi:10.1098/rstb.1952.0012.

Turk, G. (1992). Re-tiling polygonal surfaces. In *Proceeding SIGGRAPH '92 Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, pages 55–64. doi:10.1145/133994.134008.

Van Campen, J., Kassapoglou, C., and Gürdal, Z. (2011). Design of fiber-steered variable-stiffness laminates based on a given lamination parameters distribution. *AIAA paper*, 1894.

Van Gigch, J. P. (1991). *System Design Modeling and Metamodeling*. The Language of Science. Plenum Press, New York.

Vasey, L., Baharlou, E., Dörstelmann, M., Koslowski, V., Prado, M., Schieber, G., Menges, A., and Knippers, J. (2015). Behavioral design and adaptive robotic fabrication of a fiber composite compression shell with pneumatic Formwork. In Combs, L. and Perry, C., editors, *Computational Ecologies: Proceedings of the 35th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, pages 297–309. University of Cincinnati, Cincinnati OH.

Vincent, J. F. (2006). The materials revolution. *Journal of Bionic Engineering*, 3(4):217–234. doi:10.1016/S1672-6529(07)60005-5.

Von Bertalanffy, L. (1968). *General System Theory: Foundations, Development, Applications*. George Braziller New York.

Von Neumann, J. (1958). *The Computer and the Brain*. Mrs. Hepsa Ely Silliman Memorial Lectures. Yale Univ. Press, New Haven.

Von Neumann, J. and Burks, A. W. (1966). *Theory of self-reproducing automata*. University of Illinois Press, Urbana.

Von Uexküll, J. (1926). *Theoretical Biology*. London, K. Paul, Trench, Trubner & co. ltd. and New York, Harcourt, Brace & company, inc.

Von Uexküll, J. (2009). An introduction to Umwelt. In Hensel, M. U., Hight, C., and Menges, A., editors, *Space Reader*, AD reader, pages 145–148. Wiley, Chichester, U.K.

Walter, M. (1998). *Integration of Complex Shapes and Natural Patterns*. Doctor of Philosophy.

Walter, M., Fournier, A., and Reimers, M. (1998). Clonal mosaic model for the synthesis of mammalian coat patterns. In *Graphics Interface*, pages 82–91.

Wang, W. and Liu, Y. (2010). A note on planar hexagonal meshes. In Emiris, I. Z., Sottile, F., and Theobald, T., editors, *Nonlinear Computational Geometry*, volume 151 of *The IMA Volumes in Mathematics and its Applications*, pages 221–233. Springer New York, New York, NY. doi:10.1007/978-1-4419-0999-2_9.

Wang, W., Liu, Y., Yan, D., and Chan, B. (2008). *Hexagonal meshes with planar faces*. Technical Report TR-2008-13, Department of Computer Science, University of Hong Kong, Hong Kong. http://www.cs.hku.hk/research/techreps/document/TR-2008-13.pdf.

Weaver, W. (1948). Science and complexity. *American Scientist*, 36(536-544).

Weinstock, M. (2004). Morphogenesis and the mathematics of emergence. In Hensel, M. U., Menges, A., and Weinstock, M., editors, *Emergence*, volume 096 of *Profile*, pages 10–17. Wiley-Academy, Chichester.

Weinstock, M. (2010). Self-organization and material constructing. In Corser, R., editor, *Fabricating Architecture*, pages 140–151. Princeton Architectural Press, New York, N.Y.

Weisstein, E. W. (n.d.). *Limaçon: From MathWorld–A Wolfram Web Resource.* http://mathworld.wolfram.com/Limacon.html.

Wengzinek, J. (2016). *CrowdCreating: Using web techniques and agent-based systems for architectural design*. M. Sc. thesis, Institute for Computational Design and Construction (ICD).

Wester, T. (1989). Structures of nature in modern buildings. *Seisan-Kenkyu, Monthly Journal of Inst. of Industrial Science, University of Tokyo.*, 41(9):694–700.

Wilensky, U. (1999a). *NetLogo*. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. http://ccl.northwestern.edu/netlogo/.

Wilensky, U. (1999b). *NetLogo Segregation model.* Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. http://ccl.northwestern.edu/netlogo/models/Segregation.

Willis, D. and Woodward, T. (2010). Diminishing difficulty: mass customization and the digital production of architecture. In Corser, R., editor, *Fabricating Architecture*, pages 178–213. Princeton Architectural Press, New York, N.Y.

Wilson, E. O. (1975). *Sociobiology*. The New Synthesis. Belknap Pr, Cambridge, Mass.

Wolfram, S. (2002). *A New Kind of Science*. Wolfram Media and London : Turnaround, Champaign, Ill.

Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: theory and practice. *Knowledge Engineering Review*, 10(02):115. doi:10.1017/S0269888900008122.

Yablonina, M. (2015). *Mobile Robotic Fabrication System for Filament Structures*. M. Sc. thesis, Institute for Computational Design and Construction (ICD), University of Stuttgart.

Zhou, L., Atmosukarto, I., Leow, W. K., and Huang, Z. (2002). Reconstructing surface discontinuities by intersecting tangent planes of advancing mesh frontiers. In Vince, J. and Earnshaw, R., editors, *Advances in Modelling, Animation and Rendering*, pages 183–199. Springer London, London. doi:10.1007/978-1-4471-0103-1_12.

Zimmer, H., Campen, M., Herkrath, R., and Kobbelt, L. (2013). Variational tangent plane intersection for planar polygonal meshing. In Hesselgren, L., Sharma, S., Wallner, J., Baldassini, N., Bompas, P., and Raynaud, J., editors, *Advances in Architectural Geometry 2012*, pages 319–332. Springer Vienna, Vienna. doi:10.1007/978-3-7091-1251-9_26.

List of Figures

2.1.1 The Gielingh's classification of the building lifecycle; redrawn by author based on: Eastman (1999, p. 6). 11

2.1.2 Kieran and Timberlake (2004) propose the development of a model to manage the knowledge of different sectors to reinstate architects as a "twenty-first century maestro," rather than a "master builder"; redrawn by author based on: Kieran and Timberlake (2004, pp. 12-22). 12

2.1.3 Relating building elements to object-oriented programming; redrawn by author based on: Björk (1989, p. 73). 12

2.1.4 Top-down relation between ultimate forms and constituents' elements. 13

2.1.5 The schematic diagram of the design computation process; redrawn by author based on: Mitchell (1990, p. 180). 16

2.1.6 A schematic diagram to define relation between form and context with defined boundary. 18

2.2.1 The development of a "theoretical morphology" in bilateral relation between the simulation of morphogenesis and the construction of a hypothetical morphospace (McGhee 1999, pp. 1-2). 24

2.2.2 The Geometrical differentiations of Limacon (Weisstein n.d.) denoted to the equation $r = b + a \times \cos(\Theta)$ in which a and b are parameterized to d. 25

2.2.3 The empirical morphospace articulates the theoretical morphology in two areas of existent and nonexistent forms; adapted from McGhee (1999, p. 25). 25

2.2.4 A schematic diagram of developing the theoretical morphospace; redrawn by author based on: McGhee (2006, p. 61). 26

2.2.5 The development of a Hyper-dimensional morphospace; redrawn by author based on: McGhee (2006, pp. 110-116). 27

2.2.6 Classification of theoretical morphospace into extrinsic and intrinsic constraints; redrawn by author based on: McGhee (2006, p. 116). 28

2.2.7 The diagram of constructional morphology developed by Seilacher (1970). Seilacher (1973) considered these approaches as "traditional," "functional," and "fabricational" aspects (Seilacher 1973, p. 451); redrawn by author based on: Seilacher (1970, p. 394). 29

2.2.8 The schematic diagram of morphodynamics or biomorphodynamics illustrates the relation between the triangle of constructional morphology and the tetrahedron of morphodynamics, which results in the face of "inclusive organism" (Seilacher 1991a,b); redrawn by author based on: Seilacher (1991a, p. 252), Seilacher (1991b, p. 6), and Seilacher and Gishlick (2014, p. 13). 29

2.2.9 Inclusive design computation is shown as the core of interactions among fabricational morphogenesis (fabrication), environment effects, and performative criteria. 31

2.2.10 Inclusive design computation represents the inclusion of different active agencies in the form evolvements. 36

2.3.1 Conducting CGPs through two primitive mechanisms as f and g; redrawn by author based on: Holland (2000, p. 135). 40

2.3.2 The simple mechanism of Bertalanffy's feedback loop; redrawn by author based on: Von Bertalanffy (1968, p. 42). 42

2.3.3 A schematic diagram of population growth with two negative and positive feedback loops; redrawn by author based on: Camazine et al. (2003, p. 17). 42

2.3.4 A simple diagram of a lever emphasized within Complex Adaptive System (CAS); redrawn by author based on: Holland (2000, p. 127). 45

2.3.5 A diagram of modeling relation, where N is computation and F is the Turing machine; redrawn by author based on: Casti (1994, p. 274). 47

2.3.6 A "formal model of models," where S is the state of the model, $F(S)$ is a transition function, $f(s)$ is a simpler transition function and $E(S)$ is the equivalence class (Miller and Page 2007, p. 38-40); redrawn by author based on: Miller and Page (2007, p. 38). 48

2.3.7 A system classification, developed by Boulding (1968); redrawn by author based on: Van Gigch (1991, p. 66). 50

2.3.8 The classification of behavior; redrawn by author based on: Rosenblueth et al. (1943, p. 21). 51

2.3.9 A schematic diagram of a layered control system; redrawn by author based on: Brooks (1986, p. 17). 52

2.3.10 The schematic illustration, developed by De Wolf and Holvoet (2005), represents the self-organization of a system (a), the emergence of a system (b), and the combination of self-organization and the emergence (c); redrawn by author based on: De Wolf and Holvoet (2005, p. 10). 54

2.4.1 A schematic representation of an adaptive agent in a Complex Adaptive System (CAS); adapted from Holland (1995, p. 6). 57

2.4.2 Taxonomy of autonomous agents; redrawn by author based on: Franklin and Graesser (1997, p. 31). 58

2.4.3 A classification of Autonomous agents; redrawn by author based on: Pfeifer and Scheier (2001, p. 26). 58

2.4.4 Braitenberg (1984) developed mobile agents with two sensory systems and two motors with different connections of a, b, and c; redrawn by author based on: Braitenberg (1984). 59

2.4.5 A schematic diagram of interaction between agents and environments; redrawn by author based on: Russell and Norvig (2003, p. 33). 60

2.4.6 Patch systems and distributed agents (turtles). 61

2.4.7 Information space including different layers to enhance agents' behaviors. A schematic diagram of overlapping various layers; developed based on: Macal and North (2009, p. 94), Macal and North (2010, p. 152), and Torrens (2010, p. 430). 61

2.4.8 Two different mechanisms for finding neighbors; redrawn by author based on: Epstein and Axtell (1996, pp. 24, 39-40). 62

2.4.9 A simple algorithm developed by Alvarez and Martinez (2015) represents the process of exploring the environment to find food resources. Embedding signals within the environment is abstracted from the stigmergic communications; source: Alvarez and Martinez (2015). 63

2.4.10 The schematic diagrams of Conway's game of life. 65

2.4.11 A Segregation model; based on the model developed by Wilensky (1999b). 66

2.4.12 A Sugarscape model; based on the model developed by Li and Wilensky (2009). 66

2.4.13 Top row: Euclidean space used to find closest agents via a Cartesian coordinate system; bottom row: Topological space used to indicate adjacent agents through diagonal and orthogonal methods of Moore neighborhood. 67

2.4.14 Three behaviors of "cohesion," "alignment," and "separation" achieved through locomotive behaviors (Reynolds 1999); redrawn by author based on: Reynolds (1999, pp. 783-785). 68

2.4.15 An example of formalizing ABM, such as Conway' game of life through CGP; redrawn by author based on: Holland (2000, p. 137). 71

2.4.16 Comparing two different agencies: Left diagram represents the top-down organization of an agency; redrawn by author based on: Minsky (1988, p. 23). Right diagram illustrates the bottom-up organization (self-organization) processes among agents to erect agency. 71

2.4.17 A diagram of simulation method; redrawn by author based on: Gilbert and Troitzsch (2005, p. 17). 74

3.2.1 The simulation of the diagram of spaces; source: Ireland (2008). 81

3.2.2 The behavioral rules of interacting “bubble diagrams”; redrawn by author based on: Hao and Jia (2010). 81

3.2.3 The generated traces based on interactions between agents and the environment; source: Carranza and Coates (2000). 83

3.2.5 The development of an agent-based system to simulate the Zollinger system; source: Tamke et al. (2010b,a). 84

3.2.4 The behavioral rules for column-like-agents; redrawn by author based on: Scheurer (2007). 84

3.2.6 The generation of fibers on the target surface based on performative criteria; source: Tsiliakos (2012). 85

3.2.7 The implication of a flocking algorithm and their responses to cellular automata rules; source: Anzalone and Clarke (2004). 85

3.2.8 The simulation of the adaptive growth based on the robotic constraints; source: Narahara (2008, 2013a,b). 86

4.1.1 Two steps: The random distribution of agents (left image) and the relaxation process of agents (right image). 91

4.1.2 Increasing the force values among type B agents starts to generate rings around the type B agent with isolated force value. 92

4.1.3 Increasing the number of type A agents simulates cellular morphologies. 93

4.1.4 Schematic diagrams of an autonomous agent (left image) and a turtle agent (right image). 93

4.1.5 Relaxation behaviors of agents. 94

4.1.7 Cellular behaviors of agents. 94

4.1.6 Simple interaction between an agent and its adjacent agents. 94

4.1.8 Repulsion and attraction forces between types A and B. 95

4.1.9 Repulsion forces among type B agents. 95

4.1.10 Attraction forces among type B agents. 96

4.1.11 A schematic diagram of a behavioral layer. 97

4.1.12 The effect of environment circumferences on macro-regularities; left image: The environment with reflecting boundary; right image: The environment with periodic boundary. 97

4.1.13 The attraction and repulsion behaviors of circumferences; left image: Repulsion behaviors; right image: Attraction behaviors. 98

4.1.14 Topographical and topological connectivity; left image: Euclidean space; right image: Topological space. 99

4.1.15 Connectivity effect on emergence behaviors; left image: Interaction between agents and their closest neighbor; middle image: Random distribution of agents; right image: Interaction between agents and all other agents. 100

4.1.16 The sorting mechanism that finds the closest agents within the specific radius. 100

4.1.17 The development of self-organization at the micro and the macro levels. At the micro-level, agents avoid any overlapping boundaries. At the macro-level, agents fill the manifold with the agents. 101

4.1.18 The diagram of mean displacements that are analyzed over time. 101

4.1.19 Assembling behaviors, including inhibitory mechanisms, intrinsic and extrinsic behavioral layers, and classifier actions. 102

4.1.20 Parameterizing repulsion and adhesion behaviors; left image: Increasing repulsion behaviors; middle image: Random distribution of agents; right image: Increasing adhesion behaviors. 103

4.2.1 A schematic diagram of intertwined micro-macro levels. 105

4.2.2 A simple representation of inter-relations among agents, meta-agents, and meta-meta-agents. 108

4.2.3 A conceptual diagram comparing micro-macro levels with different level of aggregation. 108

4.2.4 Conducting an aggregation technique via the transition of lower-level elements to higher-level orders. 109

4.2.5 Three types of n-gons with different configurations of nodes and inter-nodes. 109

4.2.7 A schematic Diagram of a valence with five connections. 110

4.2.8 Agent-Agent interactions at shared valences and available edges. 110

4.2.9 A morphology of one meta-agent. 110

4.2.6 An aggregation process of a hexagon, which includes sub-agents, agents, and meta-agents. 110

4.2.10 Different assemblies of agents and meta- agents. 110

4.2.12 Combination of two different connections in generating one meta-agent. 111

4.2.11 The comparison between two different capacities of valences. 111

4.2.14 Assembly procedures demonstrate the importance of edges and the valences. 112

4.2.13 Indicating micro-macro levels within a behavioral layer. 112

4.2.15 A schematic representation of the assembly process. 113

4.2.16 Results of the self-assembly process within an environment with an anchor position. 114

4.2.17 Assembly processes on target surfaces (synclastic and anticlastic). 114

4.2.18 Assembly behaviors utilized with inhibitory mechanisms. 115

4.2.19 Calculating permutations and probabilities for assembly sequences. 116

4.2.20 Agent's adaptation to the user intervention: i.e., user can control agents assembly by fixing individual agents (A/B). 118

4.3.1 Conducting the tangent plane intersection algorithm on synclastic surface (left image) and anticlastic surface (right image). 121

4.3.2 The Dupin duality for three intersecting planes. 122

4.3.3 ICD/ITKE research pavilion 2011; source: ICD/ITKE University of Stuttgart. 123

4.3.4 A theoretical morphospace based on plywood systems and fabrication tools; source: Institute for Computational Design and Construction (ICD), University of Stuttgart. 124

4.3.5 The fabrication setup: Milling process of timber plates; source: Institute for Computational Design and Construction (ICD), University of Stuttgart. 124

4.3.7 Different types of polygonal geometry based on Gaussian curvature. 126

4.3.6 Geometrical definition of agents at the level of plane geometry and the level of polygonal geometry. 126

4.3.8 Agent-Agent interactions; left image: Adhesion behaviors; right image: Repulsion behaviors. 127

4.3.9 Agent-Environment interactions; left image: Adhesion behaviors; right image: Repulsion behaviors. 128

4.3.10 A conceptual relation among agents' morphology and surrounding agents' curvatures, where K represents Gaussian curvature. 128

4.3.11 The intersection of naked agents with surrounding plate-like agents TP_a, TP_b, TP_c and the bounding plane BPa. 129

4.3.12 The clipping algorithm; bottom row represents the sequence of clipping the base geometry to develop the final form determined by the agents. 129

4.3.13 The implementation of a clockwise-ordered algorithm to sort intersected points. 130

4.3.14 The implementation of the Dupin duality algorithm to control intersected point U. 131

4.3.15 The robotic milling process: An interplay between material and fabrication setups; source: Institute for Computational Design and Construction (ICD), University of Stuttgart. 132

4.3.16 Fabrication and material constraints: "Polygon angle," "polygon radius," and "connection angle" (Menges 2013); source: Institute for Computational Design and Construction (ICD), University of Stuttgart. 133

4.3.17 An analysis of surface curvature. 134

4.3.18 The controlling mechanism of connection angles. 134

4.3.19 The simulation of a connection angle mechanism; left-top image: The initial state; left-bottom image: The final result; right image: The process of approaching to the appropriate angle. 134

4.3.20 A simulation of the polygonal radii mechanism via the agent division mechanism. 135

4.3.21 The analyses of implementing the polygonal radii mechanism. 136

4.3.23 Overlapping topological and topographical spaces on tagging procedures. 137

4.3.22 Generative agents-based design computation for plate-like agents; top row images: Initiated with five agents, connection angle $[40^\circ - 140^\circ]$ and polygonal radii $[25\ units - 550\ units]$; bottom row images: Initiated with four agents, connection angles $[30^\circ - 150^\circ]$ and polygonal radii $[50\ units - 350\ units]$. 137

4.3.24 Conducting an inhibitory mechanism and a coordination system to interrelate topological connectivity (left-top image), morphospace or parametric space (left-bottom image), and Euclidean space (right image). 138

4.3.25 A schematic illustration of agent-agent and agent-environment interactions via two mechanisms of motion behaviors and geometrical mechanisms. 139

4.3.26 The simulation model of the prototype. 139

4.3.27 Details of the physical prototype; source: Institute for Computational Design and Construction (ICD), University of Stuttgart. 140

4.3.28 The final physical prototype; source: Institute for Computational Design and Construction (ICD), University of Stuttgart. 141

4.3.29 Landesgartenschau Exhibition Hall 2014; source: ICD/ITKE/IIGS University of Stuttgart. 141

4.4.1 A schematic diagram of an agent with a parametrized field vision and a sensory field tentacle. 147

4.4.2 The simulation of an agent's perception mechanisms. 148

4.4.3 Overlapping Topological and Topographical levels to find related patches. 149

4.4.4 A schematic diagram of interaction between a sensory fixed tentacle and the environment anomaly. 150

4.4.5 Two main behaviors; left image: Reinforcing the structural elements; right image: Covering the surface target. 150

4.4.6 Embedding structural principals on the formwork. The example of ICD/ITKE research pavilion 2014-15. 151

4.4.7 The details of a selected patch system with stored data that contains the trails of agents and the external cables. 151

4.4.8 Agent-Patch system interactions: The simulation of the agent's behaviors on the environment and the process of storing its behaviors into the environment. 152

4.4.9 Agent's trail system interactions: The simulation of agent's behaviors on the environment and the process of extracting previously stored behaviors from the environment. 153

4.4.10 A gradient system that informs the agent about the position of the anchor points. 153

4.4.11 The inhibitory mechanisms with three procedures: Perception, "IF/THEN," and Tagging. 154

4.4.12 The details of a tagging mechanism with two steps: Approaching and leaving anchor points. 154

4.4.13 The interaction between robot end-effectors and the formwork. 155

4.4.14 The schematic illustration of behavioral layers with their stimuli S and responses R. 156

4.4.15 The schematic illustration of a coordination mechanism, including the classifier responses R with relative strengths G. 157

4.4.17 The biological role-model, a water spider; source: ICD/ITKE University of Stuttgart. 158

4.4.18 The relation between the industrial robot and the composite shell; source: ICD/ITKE University of Stuttgart. 158

4.4.16 The research pavilion 2014-15; source: ICD/ITKE University of Stuttgart. 158

4.4.20 The process of laying fibers on the membrane. 159

4.4.19 Overlapping behavioral layers on the final fiber layers; source: ICD/ITKE University of Stuttgart. 159

4.4.21 The fabrication constraints; left image: The robot Axis 1 limitation to 185 degrees; right images: The robot Axis 6 limitations; source: ICD/ITKE University of Stuttgart. 159

4.4.22 Behavioral categorization, including biomimetic, fabrication, and environment. 160

4.4.23 Over-layering fibers in response to different behavioral layers; source: ICD/ITKE University of Stuttgart. 161

5.1.1 Formal model of inclusive agencies, where $F(s)$ defines the transition laws, $f(s)$ is transition algorithms, and $E(S)$ is the observation for abstraction procedures. The development of this diagram is adjusted to the models developed by Holland (2000); Miller and Page (2007). 166

5.1.3 The formal model of fabrication. Dashed areas indicate the hyper-dimensional morphospaces. Dotted lines represent the equivalence classes, which map the observation of real fabrication setups. 167

5.1.2 The transition model of two generative mechanisms of $f_1(s)$ and $f_2(s)$. 167

5.1.4 The customized end-effector for layering fibers on the membrane; source: ICD/ITKE University of Stuttgart. 168

5.2.1 The schematic illustration of micro-macro effects with the process of self-organization. Adapted from De Wolf and Holvoet (2005). 170

5.2.2 The correlations between local-global levels and micro-macro effects. 170

5.3.1 The schematic illustration of the fabricational morphogenesis that includes material agencies and fabrication agencies. 173

5.3.2 The schematic diagram for generating plate-like-agents with one setup of behaviors. Inspired by Arkin (1998, p. 119). 178

5.3.3 The extrinsic and intrinsic levels of behaviors. 179

5.3.4 The external and internal levels of agent-agent interactions for generating a plate-like-agent. 180

5.3.5 The based behavior definitions for interacting the fibrous-like agent with the environment and the laid fibers. 180

5.3.6 The mechanisms of assembly sophisticated behaviors onto the primitive behaviors. 182

5.3.7 The intervening mechanism of behaviors by activating and deactivating the behavioral layers. 182

5.3.9 A patch system with embedded trail paths and the external cable. 184

5.3.8 The communication network; left image: The topological connectivity of von Neumann neighborhoods with the radius three; right image: The Euclidean space with the specific radius. 184

6.2.1 The design data for constraining the agent's behaviors. 191

6.2.2 The process of constraining the connection angles, where $t_1,\ t_2,\ \cdots,\ and\ t_n$ indicate the time steps for constraining the generated connection angles. 192

6.3.1 The global-local effects on the plate-like agents behaviors. 193

6.3.2 The global-local effects on the plate-like agents behaviors. 194

6.3.3 A theoretical morphospace with two variable polygonal radii and connection angles. The agent behaves within the acceptable area, and it requires mechanism to adjust its behaviors at the critical point c to remain in the boundary. Inspired by Meyer and Guillot (1991, p. 2) and Pfeifer and Scheier (2001, p. 93). 195

6.4.1 The sophisticated behavioral layers, which navigate the agent's behavior from one anchor point to the other anchor point. 197

6.4.2 The inter-node length adjustment in assembly agents experiment. Inspired by Wengzinek (2016). 199

6.4.3 The intersected vertices arrangements in plate-like-agents. 199

6.4.4 The laid fibers on the formwork; source: ICD/ITKE University of Stuttgart. 202

6.4.5 The sequence of laying fiber on the formwork. 202

7.4.1 The cyber-physical system, research pavilion 2014-15; source: ICD/ITKE University of Stuttgart. 216

7.4.2 The woven structure developed through the behavioral fabrication process; source: Brugnaro (2015). 216

7.4.3 The filament structures developed via overlapping virtual and physical agents; source: Yablonina (2015). 217

7.4.4 The transfer of a computational simulation to a physical model; source: Rusenova (2015). 217

Curriculum Vitae

Personal data

	Ehsan Baharlou
1981	Born in Tehran, Iran

Education

2011-2017	Doctoral Candidate (ICD) University of Stuttgart, Baden-Wuerttemberg, Germany Faculty of Architecture and Urban Planning Institute for Computational Design and Construction (ICD) Dr.-Ing. Thesis: “Generative Agent-Based Architectural Design Computation-Behavioral Strategies for Integrating Material, Fabrication and Construction Characteristics in Design Processes” Advisor: Prof. Achim Menges
1999- 2007	Master of Science in Architecture Islamic Azad University, Central Tehran Branch, Iran Faculty of Art and Architecture Advisor: Prof. Mansour Falamaki

Professional Practice

2010-2017	Tutor and Lecturer (ICD) University of Stuttgart, Baden-Wuerttemberg, Germany Faculty of Architecture and Urban Planning Institute for Computational Design and Construction (ICD)
2007-2010	a-matters – Studio Design and Construction , Tehran, Iran
2003-2004	Fluid Motion Architects, Tehran, Iran
2002-2003	Group Five Design Studio, Tehran, Iran