

STUTTGARTER BEITRÄGE ZUR PRODUKTIONSFORSCHUNG

RAPHAEL BUCK

Entwurfsmuster für den Aufbau von Baukästen für das Funktionale Engineering



Universität Stuttgart



Fraunhofer

IPA

STUTTGARTER BEITRÄGE ZUR PRODUKTIONSFORSCHUNG BAND 41

Herausgeber:

Univ.-Prof. Dr.-Ing. Thomas Bauernhansl

Univ.-Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl

Univ.-Prof. a. D. Dr.-Ing. Prof. E.h. Dr.-Ing. E.h. Dr. h.c. mult. Engelbert Westkämper

Raphael Buck

**Entwurfsmuster für den Aufbau von Baukästen für das
Funktionale Engineering**

FRAUNHOFER VERLAG

Kontaktadresse:

Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, Stuttgart
Nobelstraße 12, 70569 Stuttgart
Telefon 07 11 9 70-00, Telefax 07 11 9 70-13 99
info@ipa.fraunhofer.de, www.ipa.fraunhofer.de

STUTTGARTER BEITRÄGE ZUR PRODUKTIONSFORSCHUNG**Herausgeber:**

Univ.-Prof. Dr.-Ing. Thomas Bauernhansl
Univ.-Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl
Univ.-Prof. a. D. Dr.-Ing. Prof. E.h. Dr.-Ing. E.h. Dr. h.c. mult. Engelbert Westkämper

Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, Stuttgart
Institut für Industrielle Fertigung und Fabrikbetrieb (IFF) der Universität Stuttgart
Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW)
der Universität Stuttgart

Titelbild: © Adrian Neyrinck

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über www.dnb.de abrufbar.

ISSN: 2195-2892

ISBN (Print): 978-3-8396-0828-9

D 93

Zugl.: Stuttgart, Univ., Diss., 2014

Druck: Mediendienstleistungen des Fraunhofer-Informationszentrum Raum und Bau IRB, Stuttgart
Für den Druck des Buches wurde chlor- und säurefreies Papier verwendet.

© by **FRAUNHOFER VERLAG**, 2015

Fraunhofer-Informationszentrum Raum und Bau IRB
Postfach 80 04 69, 70504 Stuttgart
Nobelstraße 12, 70569 Stuttgart
Telefon 07 11 9 70-25 00
Telefax 07 11 9 70-25 08
E-Mail verlag@fraunhofer.de
URL <http://verlag.fraunhofer.de>

Alle Rechte vorbehalten

Dieses Werk ist einschließlich aller seiner Teile urheberrechtlich geschützt. Jede Verwertung, die über die engen Grenzen des Urheberrechtsgesetzes hinausgeht, ist ohne schriftliche Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Speicherung in elektronischen Systemen.

Die Wiedergabe von Warenbezeichnungen und Handelsnamen in diesem Buch berechtigt nicht zu der Annahme, dass solche Bezeichnungen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und deshalb von jedermann benutzt werden dürften. Soweit in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z.B. DIN, VDI) Bezug genommen oder aus ihnen zitiert worden ist, kann der Verlag keine Gewähr für Richtigkeit, Vollständigkeit oder Aktualität übernehmen.

GELEITWORT DER HERAUSGEBER

Produktionswissenschaftliche Forschungsfragen entstehen in der Regel im Anwendungszusammenhang, die Produktionsforschung ist also weitgehend erfahrungsbasiert. Der wissenschaftliche Anspruch der „Stuttgarter Beiträge zur Produktionsforschung“ liegt unter anderem darin, Dissertation für Dissertation ein übergreifendes ganzheitliches Theoriegebäude der Produktion zu erstellen.

Die Herausgeber dieser Dissertations-Reihe leiten gemeinsam das Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA und jeweils ein Institut der Fakultät für Konstruktions-, Produktions- und Fahrzeugtechnik an der Universität Stuttgart.

Die von ihnen betreuten Dissertationen sind der marktorientierten Nachhaltigkeit verpflichtet, ihr Ansatz ist systemisch und interdisziplinär. Die Autoren bearbeiten anspruchsvolle Forschungsfragen im Spannungsfeld zwischen theoretischen Grundlagen und industrieller Anwendung.

Die „Stuttgarter Beiträge zur Produktionsforschung“ ersetzt die Reihen „IPA-IAO Forschung und Praxis“ (Hrsg. H.J. Warnecke / H.-J. Bullinger / E. Westkämper / D. Spath) bzw. ISW Forschung und Praxis (Hrsg. G. Stute / G. Pritschow / A. Verl). In den vergangenen Jahrzehnten sind darin über 800 Dissertationen erschienen.

Der Strukturwandel in den Industrien unseres Landes muss auch in der Forschung in einen globalen Zusammenhang gestellt werden. Der reine Fokus auf Erkenntnisgewinn ist zu eindimensional. Die „Stuttgarter Beiträge zur Produktionsforschung“ zielen also darauf ab, mittelfristig Lösungen für den Markt anzubieten. Daher konzentrieren sich die Stuttgarter produktionstechnischen Institute auf das Thema ganzheitliche Produktion in den Kernindustrien Deutschlands. Die leitende Forschungsfrage der Arbeiten ist: Wie können wir nachhaltig mit einem hohen Wertschöpfungsanteil in Deutschland für einen globalen Markt produzieren?

Wir wünschen den Autoren, dass ihre „Stuttgarter Beiträge zur Produktionsforschung“ in der breiten Fachwelt als substanziell wahrgenommen werden und so die Produktionsforschung weltweit voranbringen.

Alexander Verl

Thomas Bauernhansl

Engelbert Westkämper

**Entwurfsmuster für den Aufbau von Baukästen
für das Funktionale Engineering**

**Von der Fakultät Konstruktions-, Produktions- und
Fahrzeugtechnik der Universität Stuttgart
zur Erlangung der Würde eines Doktor-Ingenieurs
(Dr.-Ing.) genehmigte Abhandlung**

**Vorgelegt von
Dipl.-Ing. Raphael Buck
aus Riedlingen**

Hauptberichter: Univ.-Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl
Mitberichter: Univ.-Prof. Dr.-Ing. Michael Weyrich

Tag der mündlichen Prüfung: 24. Oktober 2014

Institut für Steuerungstechnik der Werkzeugmaschinen
und Fertigungseinrichtungen der Universität Stuttgart

2014

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW) der Universität Stuttgart, in Kooperation mit der Firma Eplan Software und Service GmbH & Co. KG (ehemals Mind8 GmbH & Co. KG).

Herrn Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl danke ich herzlich für seine Unterstützung und die wertvollen Anregungen, die zum Gelingen dieser Arbeit wesentlich beigetragen haben, sowie für die Übernahme des Hauptberichtes. Herrn Prof. Dr.-Ing. Michael Weyrich danke ich sehr für seine Bereitschaft, den Mitbericht zu übernehmen.

Weiterhin danke ich allen aktuellen und ehemaligen Mitarbeitern des Institutes für die kollegiale und offene Zusammenarbeit und die interessanten Diskussionen, insbesondere den Kolleginnen und Kollegen aus Gruppe 1 „Planungssysteme und Engineeringmethoden“. Besonderer Dank gilt auch den Gründern der Firma Mind8, Hr. Dr.-Ing. Jörg Lewek und Hr. Dr.-Ing. Marco Litto, für das mir entgegenbrachte Vertrauen, die fachliche Unterstützung sowie die gute Zusammenarbeit. Ebenso gilt dieser Dank den Entwicklern und Consultants im Engineering-Center-Team der Firma Eplan.

Den Herren Dr.-Ing. Jörg Lewek und Dr.-Ing. Armin Lechler danke ich sehr für die kritische Durchsicht meiner Arbeit.

Ein besonderer Dank gilt meinen Eltern Elisabeth und Anton, denen ich diese Arbeit widme, für die Ermöglichung des Studiums und ihre moralische Unterstützung. Meinen Freunden sowie ganz besonders meiner Frau José danke ich für die vielen aufmunternden und bestärkenden Worte und dafür, dass sie in mir den Glauben daran aufrechterhalten haben, diese Arbeit erfolgreich fertig zu stellen.

November 2014

Raphael Buck

Kurzfassung

Unternehmen des Maschinen- und Anlagenbaus bewegen sich heute in einem stark globalisierten Markt, was zu einem hohen Wettbewerbsdruck führt. Daher setzen viele Unternehmen heutzutage neben Innovation, Energieeffizienz und Qualität immer mehr auf die Individualisierung ihrer Produkte, um dem Kunden die für ihn optimale Lösung anbieten und dadurch ein gutes Preis-Leistungs-Verhältnis erreichen zu können. Die Individualisierung von Maschinen und Anlagen führt in den Unternehmen jedoch zu einer erhöhten Komplexität, was sich vor allem im Engineering-Bereich deutlich zeigt, wo explodierende Kosten, Qualitätsprobleme und verlängerte Projektlaufzeiten zu beobachten sind.

Diesen Problemen versuchen viele Unternehmen mit baukastenbasierten Funktionalen Engineering Systemen (FES) entgegenzuwirken. Durch Modularisierung und Wiederverwendung von mechatronischen Komponenten ist es möglich, variantenreiche Maschinen und Anlagen zu konfigurieren und Daten und Projektunterlagen automatisch zu generieren. Die Erstellung der Baukästen für FES ist jedoch zeit- und kostenintensiv. Zudem finden viele Engineering-Prozesse nicht mehr zentral in einem Unternehmen statt, sondern werden firmenübergreifend in Kooperationen hoch spezialisierter Zulieferer und Dienstleister abgewickelt.

Für die reproduzierbare Erstellung leicht zu wartender Baukästen wurde deshalb in dieser Arbeit ein Katalog aus mechatronischen Entwurfsmustern erarbeitet. Um Optimierungspotenziale in vorhandenen Baukästen transparent und die Anwendung der entwickelten Entwurfsmuster messbar zu machen, wurden Metriken zur Analyse von FES-Baukästen ermittelt, die in Form eines „Model-Analysers“ umgesetzt wurden. Die mechatronischen Entwurfsmuster wurden zum einen benutzt, um Konzepte für firmenübergreifend wiederverwendbare Basisbaukästen zu entwickeln, womit sich Aufwände bei der Erstellung von Baukästen reduzieren lassen. Zum anderen wurden mit Hilfe dieser Entwurfsmuster Konzepte erstellt, um den Einsatz von FES in Kooperationen zwischen Firmen zu ermöglichen.

Short Summary

The mechanical and plant engineering industry today finds itself in an increasingly globalized market which results in a very high competitive pressure. Therefore, many companies now focus not only on innovation, energy efficiency and quality but also on customization of their products, in order to be able to offer their customers the optimum solution and consequently a good price/performance ratio. However, the customization of machines and plants in the industry results in a higher level of complexity which can be seen in particular in the field of engineering where cost explosion, quality problems and prolonged project durations can be observed.

Many companies try to avoid these problems by means of modular Functional Engineering Systems (FES). Modularization and reuse of mechatronic components enables the configuration of versatile machines and plants and the automatic generation of data and project documents. But the creation of libraries for FES is time- and cost-intensive. In addition, many engineering processes do not take place centrally in a company, but are cross-company processes in cooperation with highly specialized suppliers and service providers.

For this reason, a catalogue of mechatronic design patterns for the reproducible generation of easy to maintain libraries has been elaborated in this thesis. In order to make optimization potentials in existing libraries transparent and the application of developed design patterns measurable, metrics for the analysis of FES modules have been identified that have been implemented as a „model analyzer“. On one hand, the mechatronic design patterns were used to develop concepts for cross-company reusable basic libraries, so that the effort of producing the libraries can be reduced. On the other hand, concepts have been developed with the help of those design patterns for enabling the application of FES in cooperations between companies.

Inhaltsverzeichnis

Vorwort	III
Kurzfassung	V
Short Summary	VII
Abkürzungen	XIII
Glossar	XIV
Symbole	XVII
1 Einleitung	1
2 Grundlagen	3
2.1 Funktionales Engineering	4
2.2 Baukästen	8
2.3 Entwurfsmuster	9
3 Problemstellung	11
3.1 Qualitätsunterschiede von Baukästen	12
3.2 Fehlende Wiederverwendbarkeit von Baukästen	16
3.3 Fehlende Konzepte für Kooperationen zwischen Firmen	18
3.4 Resümee	22
4 Anforderungen an Baukästen für FES	23
4.1 Qualitätssicherung bei der Erstellung von Baukästen	23
4.2 Wiederverwendbarkeit von Baukästen	24
4.3 Unterstützung von Kooperationen zwischen Firmen	24
5 Stand der Technik und der Forschung	27

5.1	Methoden für die Analyse von Baukästen	27
5.1.1	Software-Metriken.....	27
5.2	Methoden für die Erstellung von Baukästen und Bibliotheken.....	28
5.2.1	Parametrische Baugruppen.....	28
5.2.2	Entwurfsmuster	29
5.2.3	Variantenmanagement.....	30
5.2.4	Funktionales Engineering.....	31
5.3	Firmenübergreifende Basisbaukästen	31
5.3.1	Mechanik-Normteibibliotheken	31
5.3.2	Norm-Symbolbibliotheken für die Elektr(on)ik und Fluidik, frei verfügbare Schaltplanmakros	32
5.3.3	Softwarebibliotheken.....	32
5.4	Datenformate und Systeme für den Austausch von Informationen.....	33
5.4.1	Übergabe von Handbüchern mit Konstruktions- / Entwicklungsrichtlinien.....	34
5.4.2	Manuelle Übergabe von Konfigurationsdaten in Form von Tabellen	34
5.4.3	Disziplinspezifische, herstellerübergreifende Austauschformate .	35
5.4.4	Mechatronische, herstellerübergreifende Austauschformate.....	35
5.4.5	Native Daten.....	37
5.4.6	Zusammenarbeit mit Hilfe von PDM-/PLM-Systemen.....	37
5.5	Defizite und Handlungsbedarf	38
5.5.1	Qualitätssicherung bei der Erstellung von Baukästen.....	38
5.5.2	Wiederverwendbarkeit von Baukästen.....	38
5.5.3	Unterstützung von Kooperationen zwischen Firmen.....	38
5.5.4	Handlungsbedarf	40
6	Lösungsansatz	41
6.1	Analyse vorhandener Baukästen.....	42
6.2	Übertragung und Entwicklung mechatronischer Entwurfsmuster.....	43

6.3	Entwicklung von Konzepten für firmenübergreifende Basisbaukästen	43
6.4	Entwicklung von Konzepten für die Kooperation zwischen Firmen	44
7	Analyse vorhandener Baukästen	45
7.1	Berechnungsmethoden	45
7.1.1	Maximum	46
7.1.2	Median	46
7.2	Kategorisierung der Metriken	47
7.3	Metriken	49
7.3.1	Mechatronik	50
7.3.2	Disziplinen	52
7.3.3	Projekte	53
7.4	Grenzwerte	53
7.5	Realisierung	57
7.6	Fazit	60
8	Entwurfsmuster	61
8.1	Kategorisierung der Entwurfsmuster	61
8.2	Erzeugungsmuster	63
8.2.1	Komposition	63
8.2.2	Kopiervorlage	65
8.2.3	Disabler	67
8.2.4	Erweiterungspunkt	69
8.2.5	Erweiterungspunkt mit Auswahlformel	71
8.2.6	Erweiterungspunkt mit Auswahlformel und Anzahl	73
8.3	Strukturmuster	75
8.3.1	Trennung Architektur – Baukasten	75
8.3.2	4-Ebenen-Komponentenmodell	77
8.3.3	Herstellerbibliothek	80
8.3.4	Parameter-Bibliothek	83

8.4	Verhaltensmuster.....	85
8.4.1	Schnittstellenparameter	85
8.4.2	Dynamische Parametervererbung	87
8.4.3	Zentralsteuerung	89
8.4.4	Entscheidungstabelle	92
8.4.5	Memento.....	95
9	Firmenübergreifende Basisbaukästen	99
9.1	Einbindung firmenübergreifender Basisbaukästen.....	100
9.1.1	Einbindung kompletter Basisbaukästen	102
9.1.2	Einbindung einzelner Komponenten.....	103
9.1.3	Einbindung eigener Ressourcen	104
9.2	Anwenderberichte	105
9.3	Fazit.....	110
10	Baukastenstrukturen für Kooperationen zwischen Firmen	111
10.1	Kooperation auf Baukastenebene.....	111
10.2	Kooperation auf Projektebene.....	113
10.3	Kooperation zwischen Baukasten- und Projektebene.....	115
10.4	Übersichtstabelle	116
10.5	Anwenderberichte	117
10.6	Fazit	121
11	Zusammenfassung und Ausblick	123
	Abbildungsverzeichnis	126
	Tabellenverzeichnis	128
	Literaturverzeichnis	129

Abkürzungen

AQUIMO	Adaptierbares Modellierungswerkzeug und Qualifizierungsprogramm für den Aufbau firmenspezifischer mechatronischer Engineeringprozesse
AML	AutomationML, Automation Markup Language
CAD	Computer Aided Design
CAEX	Computer Aided Engineering Exchange
COLLADA	Collaborative Design Activity
CSV	Comma Separated Values
BMK	Betriebsmittelkennzeichen
ECAD	Electrical Computer Aided Design
FB	Function Block oder Funktionsbaustein
FC	Function
FES	Funktionales Engineering System
MCAD	Mechanical Computer Aided Design
PDM	Product Data Management
PLM	Product Lifecycle Management
STEP	Standard for the Exchange of Product Model Data
UML	Unified Modeling Language
XML	Extensible Markup Language

Glossar

Änderbarkeit

Reziprokes Maß des Aufwands, einen vorhandenen Baukasten oder ein vorhandenes Baukastenelement an neue Anforderungen anzupassen.

Baukasten

Gesamtheit der mechatronischen Komponenten und Module sowie der zugehörigen Architektur innerhalb eines Funktionalen Engineering Systems.

Bibliothek

Ordnungselement zur Strukturierung innerhalb von Baukästen eines Funktionalen Engineering Systems. Die Gesamtheit aller gemeinsam verwendeten Bibliotheken ergibt den Baukasten.

Disziplin

- (a) Entwicklungsbereich des klassischen Maschinen- und Anlagenbaus, z.B. Mechanik, Elektr(on)ik, Fluidik oder Software.
- (b) Ebenfalls verwendet für die zu einem Zielsystem gehörende Datenstruktur innerhalb eines Projektes im Funktionalen Engineering System.

Entwurfsmuster

Abstraktes Lösungskonzept für die Lösung eines immer wiederkehrenden Problems. Neben einer Beschreibung des eigentlichen Kerns der Lösung enthält das Entwurfsmuster meist auch eine Problembeschreibung sowie Hinweise darüber, wann und in welcher Form es angewandt werden sollte.

Funktionales Engineering

Entwicklung einer Maschine oder Anlage mit Hilfe einer Datenstruktur, die sich nicht an einer der Disziplinen Mechanik, Elektr(on)ik, Fluidik oder Software orientiert, sondern an den vom Kunden bestellten Wertschöpfungseinrichtungen („Funktionen“), z.B. der Funktion „Späne fördern“.

Generierung

Automatische Erzeugung von Daten oder Dokumenten auf Basis einer durchgeführten Konfiguration durch ein Software-System (hier Funktionales Engineering System).

Klasse

In der objektorientierten Softwaretechnik verwendeter Begriff für ein Softwareobjekt, das Parameter, weitere Klassen und ein bestimmtes Verhalten in sich vereinen kann.

Baukastenelement

Wiederverwendbare (mechatronische) Einheit einer Maschine oder Anlage, die im Baukasten eines Funktionalen Engineering Systems abgelegt ist, z.B. ein einzelner Sensor oder Aktor.

Konfiguration

Ähnlich der „Projektierung“ eine Erstellung und Parametrierung einer konkreten Maschine oder Anlage aus vorhandenen Baukastenelementen, oftmals unterstützt durch Software mit benutzerfreundliche Bedien-Oberflächen.

Mechatronik

Übergreifende, funktionale Datenstruktur innerhalb eines Projektes im Funktionalen Engineering System.

Metrik

Ein Algorithmus zur Herleitung einer Kennzahl, die es erlaubt, beispielsweise Aussagen zu Performanz, Übersichtlichkeit und Grad der Wiederverwendung eines Baukastens oder Baukastenelements zu treffen.

Modellierung

Abbildung einer „realen“ Maschine, Anlage, Baugruppe, Funktion o.ä. als Baukastenelement(e) innerhalb eines Funktionalen Engineering Systems.

Parameter

In der Softwaretechnik verwendeter Begriff für ein Objekt, das eine oder mehrere Eigenschaften einer Klasse beinhaltet.

Performanz

Maß für die Beurteilung der Berechnungs-, Generierungs- und Reaktionszeiten innerhalb eines Funktionalen Engineering Systems.

Projekt

Datenstruktur innerhalb eines Funktionalen Engineering Systems, die alle zur Erstellung der Daten und Dokumente für eine konkrete Maschine oder Anlage benötigten Instanzen von Baukastenelementen enthält.

Projektierung

Erstellung und Parametrierung einer konkreten Maschine oder Anlage aus vorhandenen Baukastenelementen mit Hilfe eines Funktionalen Engineering Systems.

Ressource

Daten- oder Dokument-Fragment, das bei der Generierung zur Zusammenstellung eines vollständigen Datensatzes oder Dokumentes verwendet wird, z.B. ein Schaltplan-Fragment oder ein Steuerungs-Baustein.

Wiederverwendbarkeit

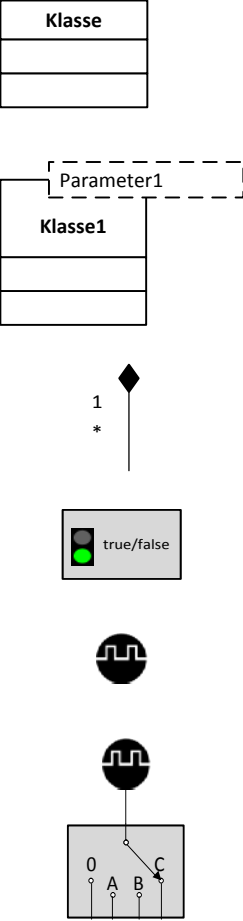
Reziprokes Maß des Aufwands, einen Baukasten oder ein Baukastenelement eines Funktionalen Engineering Systems innerhalb eines neuen Projekts, für einen neuen Anwendungsfall oder in einem anderen Kontext zu nutzen.

Zielsystem

Softwaresystem, für das aus dem Funktionalen Engineering System heraus Daten oder Dokumente generiert werden. Ein ECAD-System ist beispielsweise das Zielsystem, in das oder mit Hilfe dessen vom Funktionalen Engineering System Schaltpläne generiert werden.

Symbole

	Symbol	Bedeutung
Baukastenelemente		<p>Mechatronisches Baukastenelement</p> <p>Mechanisches Baukastenelement</p> <p>Elektrisches Baukastenelement</p> <p>Software-Baukastenelement</p>
Datenablage		<p>Baukasten</p> <p>Projekt</p> <p>Datenbank</p> <p>Black-Box-Baukasten</p> <p>Dokument</p>
Bewertungssystem		<p>Kriterium vollständig erfüllt</p> <p>Kriterium teilweise erfüllt</p> <p>Kriterium nicht erfüllt</p>

	Symbol	Bedeutung
(Erweiterte) UML		<p>Klasse</p> <p>Klasse mit Parameter</p> <p>Komposition 1 zu n</p> <p>Disabler</p> <p>Erweiterungspunkt</p> <p>Erweiterungspunkt mit Auswahlformel</p>

1 Einleitung

Unternehmen des Maschinen- und Anlagenbaus bewegen sich heute in einem stark globalisierten Markt. Neben hohem Wettbewerbsdruck, gekennzeichnet durch sinkende Preise selbst bei hochtechnologischen Industriegütern, stehen sie vor der Herausforderung immer kürzer werdender Produktlebenszyklen. Der Klimawandel und ehrgeizige Ziele der nationalen und internationalen Politik verlangen nach immer effizienteren Produkten in Bezug auf Ressourcen- und insbesondere den Energiebedarf [vgl. Eversheim 2003; Gausemeier 2009; Schmelzer 2008; Westkämper 2009].

Um sich in den internationalen Märkten eine führende Position zu sichern, setzen viele Unternehmen auf vier Standbeine: Innovation, zunehmende Individualisierung von Produkten, hohe Qualität und schnelle Lieferzeiten. Durch kundenindividuelle Lösungen ist es möglich, Produkte und Prozesse exakt auf die Bedürfnisse und Anforderungen des Kunden anzupassen. Dadurch, dass nur die vom Kunden benötigten Funktionen eingebaut werden, lassen sich ressourcen- und energieeffiziente Maschinen und Anlagen zu günstigen Preisen anbieten. Die Individualisierung der Produkte führt jedoch häufig zu Sonderlösungen und einem immer breiter und weniger beherrschbar werdenden Produktportfolio. In der Folge nehmen Engineeringaufwände zu, die Komplexität steigt an, Liefertermine können nicht gehalten werden, Fehler schleichen sich ein und das Qualitätsniveau der Produkte sinkt [vgl. Tomasz 2010; Schuh 2005; Franke 2002; Nyhuis 2013].

Kundenindividuelle Maschinen und Anlagen effizient, schnell und in hoher Qualität zu entwickeln, das ist das Ziel heute verfügbarer Funktionaler Engineering Systeme (FES). Mit Hilfe mechatronischer Baukästen werden Maschinen und Anlagen kon-

figuriert und anschließend alle benötigten Daten und Dokumente (wie z.B. Angebote, Stücklisten, Fertigungszeichnungen, Schaltpläne, SPS-Code oder Dokumentation) automatisch generiert [vgl. Litto 2004; Eplan 2012; Neyrinck 2013; Weyrich 2012].

Die Erstellung der in FES verwendeten mechatronischen Baukästen ist jedoch Stand heute zeit- und kostenintensiv, da bisher noch keine (wiederverwendbaren) Basisbaukästen zur Verfügung stehen. Daher werden Baukästen in jedem Unternehmen von Grund auf neu erstellt [vgl. Buck IV 2010]. Weiterhin ist die Qualität der Baukästen stark vom Erfahrungswissen des jeweiligen Entwicklers abhängig, weshalb sich diese stark in Bezug auf Performanz und Änderbarkeit unterscheiden. Auch gibt es momentan keine Möglichkeit, die Qualität eines vorhandenen Baukastens hinsichtlich der genannten Aspekte zu messen und zu bewerten. Außerdem werden Konzepte für den Einsatz von FES in Kooperationen zwischen Unternehmen benötigt, die einen automatischen und sicheren Austausch von Daten erlauben. Dadurch können Kosten und Fehler, die bei der manuellen Übernahme oder der doppelten Pflege von Daten entstehen, eliminiert werden [vgl. Buck II 2012].

Aufgabenstellung dieser Arbeit ist es deshalb, für die Messung der Qualität vorhandener Baukästen geeignete Metriken zu finden, die eine Analyse von Problemursachen und Verbesserungspotenzialen erlauben. Darauf aufbauend sollen Konzepte entwickelt werden, die eine methodische Unterstützung beim Aufbau von mechatronischen Baukästen für FES bieten, die Aufwände beim Aufbau reduzieren und Kooperationen zwischen Unternehmen ermöglichen.

2 Grundlagen

In diesem Kapitel werden die Grundlagen erläutert, auf denen diese Arbeit aufbaut. Dies sind insbesondere FES, die darin verwendeten Baukästen sowie das bei der Erstellung von Baukästen angewandte Konzept der Entwurfsmuster (siehe Bild 2.1). Dabei werden wichtige und immer wieder vorkommende Begriffe im Kontext dieser Dissertation definiert.

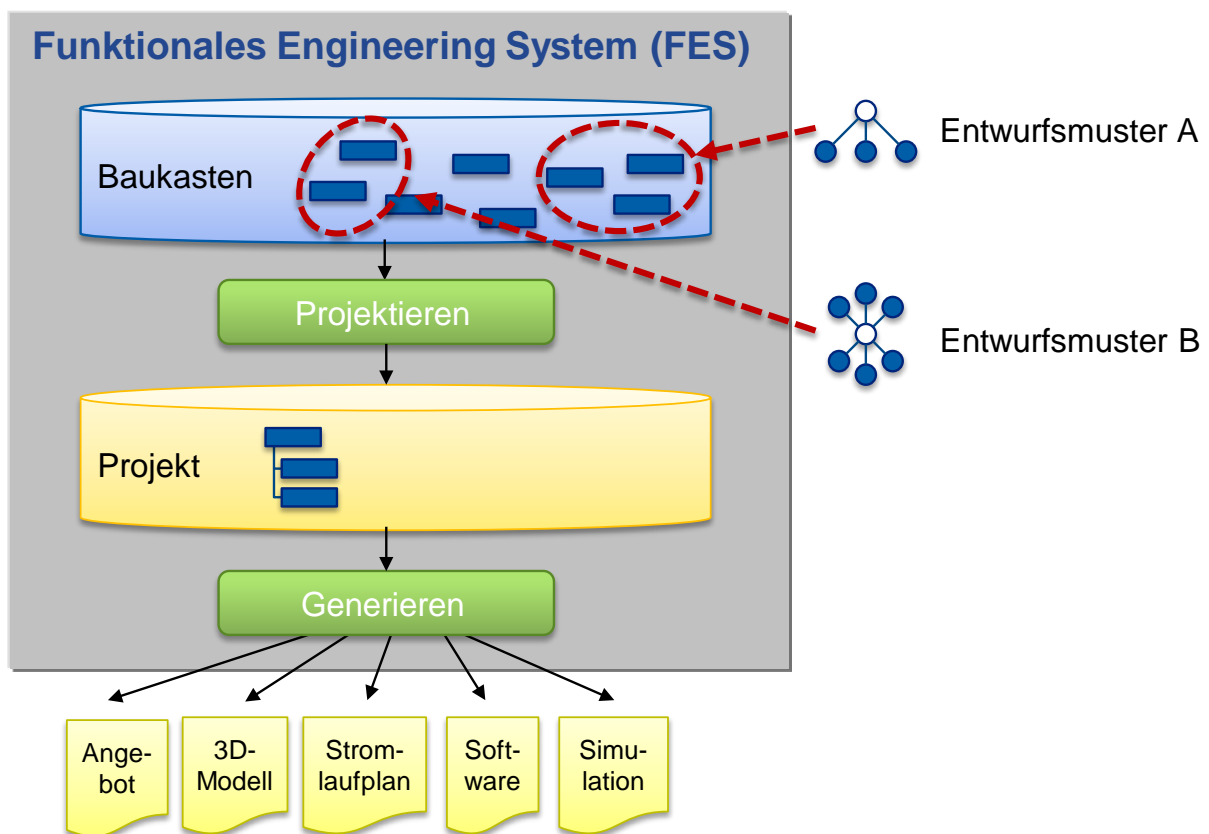


Bild 2.1: Überblick FES – Baukästen – Entwurfsmuster

2.1 Funktionales Engineering

Bei vielen Unternehmen des Maschinen- und Anlagenbaus findet bis heute das Engineering von Produkten mit hohem Sonderanteil sequenziell in den verschiedenen Abteilungen (Mechanik-, Elektro-, Softwareabteilung) statt (siehe Bild 2.2). Für gewöhnlich erfolgt im ersten Schritt die Mechanik-Konstruktion, aus der 2D- oder 3D-CAD-Daten sowie Fertigungszeichnungen und Stücklisten hervorgehen. Gleichzeitig wird eine Sensor-/Aktor-Liste erstellt, die als Grundlage für die nachfolgende Elektro- bzw. Fluid-Konstruktion dient.

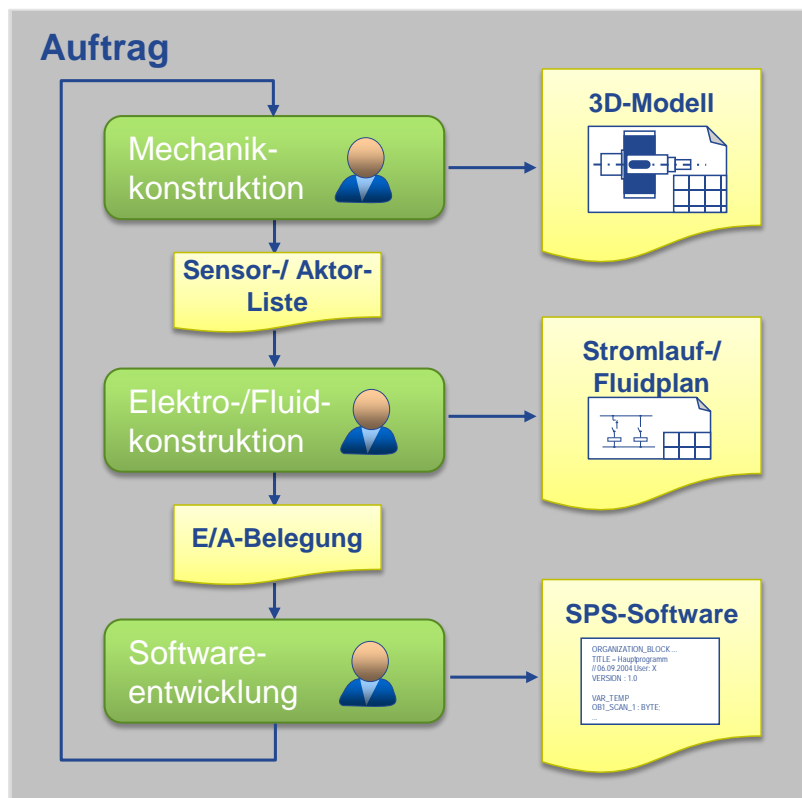


Bild 2.2: Sequenzielles Engineering [angelehnt an Litto 2004]

In der Elektro- bzw. Fluid-Konstruktion erfolgen u.a. die Auslegung der Steuerung sowie die Verdrahtung aller Aktoren und Sensoren zur Steuerung, die Auslegung der hydraulischen Komponenten und der Aufbau des Schaltschranks. Daraus resul-

tieren Stromlauf- und Hydraulikpläne als Grundlage für die Fertigung und als Dokumentation sowie eine Belegungsliste aller Ein-/Ausgänge. Diese Liste ist Voraussetzung für die Entwicklung der (Steuerungs-)Software.

Nachteile dieses sequenziellen Engineering-Prozesses sind hohe Durchlaufzeiten, hohe Aufwände durch manuelle Tätigkeiten bei der Übernahme von Daten von einer Abteilung in die nächste sowie Qualitätsprobleme durch unzureichende Abstimmung zwischen den einzelnen Gewerken. Ebenfalls nachteilig wirkt sich die geringe Möglichkeit der nachgelagerten Disziplinen zur Einflussnahme auf die Gestaltung der Maschine aus. Hierdurch gehen oftmals Potenziale verloren, die erst durch das Zusammenspiel von Mechanik, Elektr(on)ik, Fluidik und Software im Sinne der Mechatronik entstehen [vgl. Verl 2009].

Um Engineeringaufwände zu senken und Durchlaufzeiten zu verkürzen, werden oftmals Teile von alten Projekten kopiert und manuell angepasst (siehe Bild 2.3).

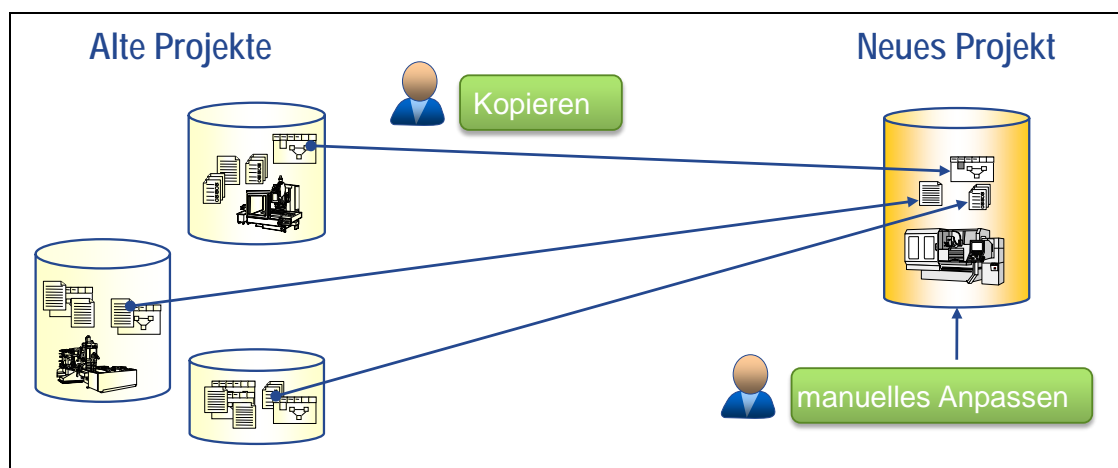


Bild 2.3: Erstellen von neuen Projekten durch Kopieren und Anpassen [vgl. Litto 2004]

Nachteil bei dieser Vorgehensweise ist jedoch, dass Fehler mitkopiert werden. Ebenso sind als langweilig empfundene, oft wiederkehrende manuelle Tätigkeiten

2.1 Funktionales Engineering

fehleranfällig. Dadurch entsteht der Bedarf an ausführlichen Tests, um hohe Fehlerfolgekosten bei der Inbetriebnahme beim Kunden zu vermeiden. Unnötige Aufwände entstehen durch Mehrfachentwicklungen, da beispielsweise aus Projekten mit unterschiedlichen Softwareständen kopiert wird.

Diesen Nachteilen wird versucht, mit Hilfe baukastenbasierter Funktionaler Engineering Systeme (FES) entgegenzuwirken (siehe Bild 2.4). Funktional bedeutet in diesem Zusammenhang, dass die Strukturierung und Modularisierung einer Maschine oder Anlage nicht gemäß einer der Disziplinen Mechanik, Elektr(on)ik, Fluidik oder Software, sondern an Hand funktionaler Einheiten stattfindet. Eine Funktion ist dabei eine in sich abgeschlossene, mechatronische, wiederverwendbare Einheit, die etwas ausführt, wofür der Kunde bezahlt, z.B. Funktion „Späne fördern“. Der funktionale Ansatz ist somit die Sicht des Kunden auf eine Maschine oder Anlage und gleichzeitig die Struktur, die sich in allen Kundendokumenten wiederfindet. Die Sichten und Strukturen der einzelnen Disziplinen hingegen unterscheiden sich häufig deutlich voneinander, was oft zu Kommunikationsproblemen führt [vgl. Hackel 2011].

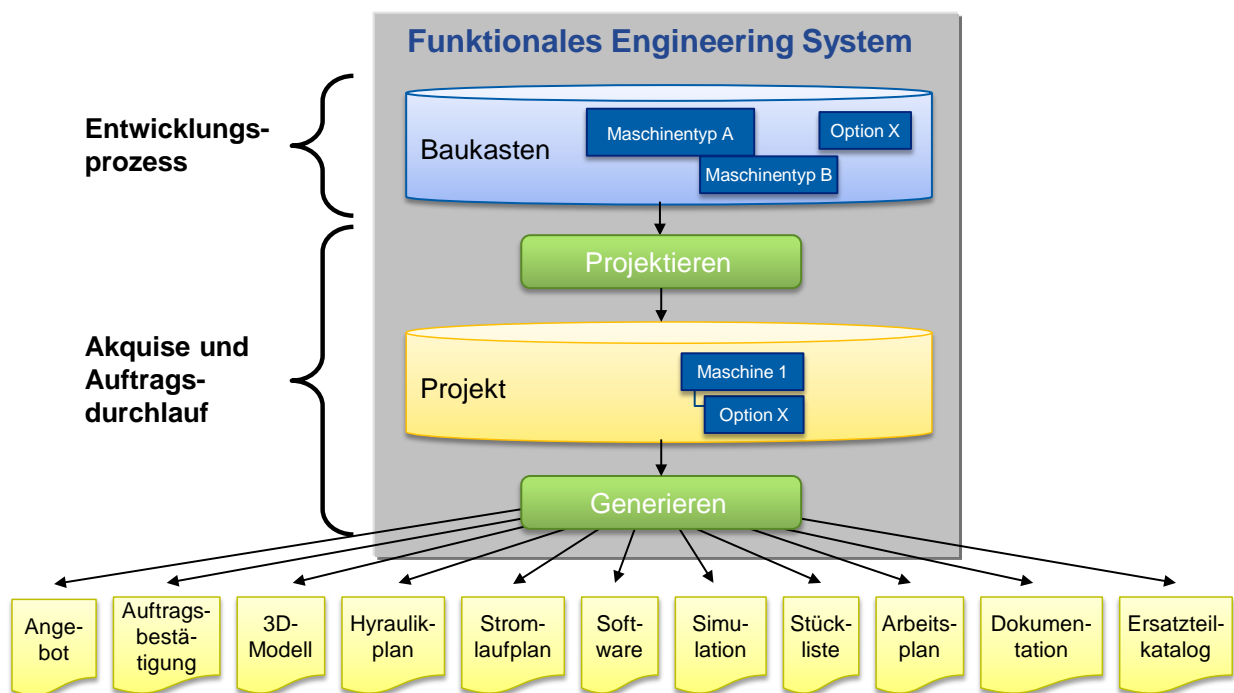


Bild 2.4: Funktionales Engineering Systeme (FES)

FES sind heute Stand der Technik [vgl. Eplan 2012]. Grundlage solcher Systeme sind Baukästen mit standardisierten mechatronischen Komponenten und Modulen, die mit Hilfe von Konfiguratoren zu Projekten zusammengestellt und parametrisiert (projektiert) werden. Anschließend können Daten und Dokumente wie z.B. Angebot, 3D-Modell, Stromlaufplan oder Stückliste automatisch generiert werden (siehe Bild 2.4). Die dadurch gewonnene Durchgängigkeit hilft, die Kommunikation zwischen den Disziplinen Vertrieb, Mechanik, Elektr(on)ik, Fluidik und Software zu verbessern und Inkonsistenzen zwischen unterschiedlichen Daten und Dokumenten zu vermeiden [vgl. Buck I 2012; Buck IV 2010; Lewek 2005].

Durch die Entkopplung von Entwicklungsprozess einerseits und Akquise und Auftragsdurchlauf andererseits wird das Kopieren und Anpassen von Teilen alter Projekte unterbunden. Stattdessen werden standardisierte und gut getestete Module in einem Baukasten abgelegt und in verschiedenen Projekten immer wieder verwendet. Durch die möglichst frühe und vollständige Konfiguration einer Maschine oder Anlage im Vertrieb wird die Definition des Produktes zeitlich nach vorne verschoben.

2.2 Baukästen

Dadurch steigt die Qualität der Produkte bei gleichzeitiger Verkürzung der Durchlaufzeit [vgl. Buck I 2012].

2.2 Baukästen

Ein **Baukasten** besteht [nach Lewek 2005] aus einer Baukastenarchitektur und einem Baukastensystem (siehe Bild 2.5). In der **Baukastenarchitektur** sind die Hierarchieebenen der zu erstellenden Maschinen oder Anlagen festgelegt, wie z.B. Anlage – Maschine – Funktionsgruppe - Funktionseinheit – Station – Komponente. Diese Ebenen definieren die Struktur des Baukastensystems.

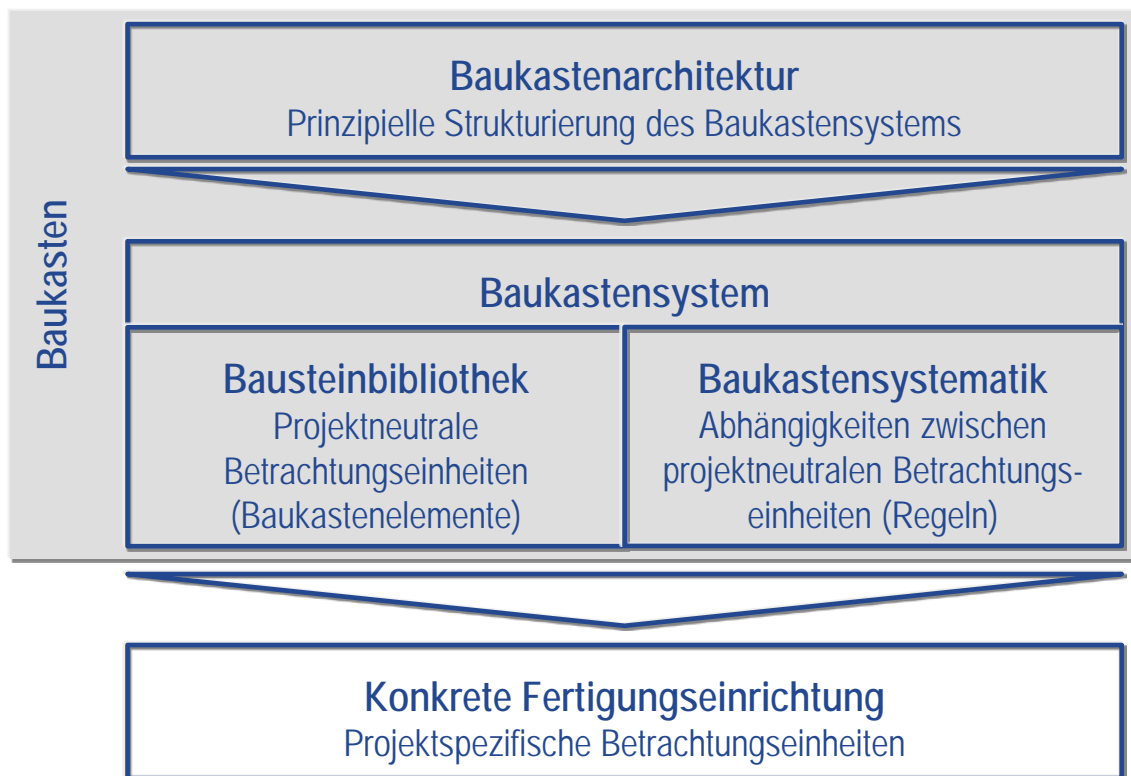


Bild 2.5: Abgrenzung von Baukasten-Begriffen [angelehnt an Lewek 2005, S. 21]

Das **Baukastensystem** wiederum besteht aus der Bausteinbibliothek und der Baukastensystematik. Die **Bausteinbibliothek** enthält projektneutrale Betrachtungsein-

heiten (**Baukastenelemente**), die beim Projektieren später instanziiert werden. Zwischen den Baukastenelementen bestehen Abhängigkeiten, die durch **Regeln** abgebildet werden (**Baukastensystematik**). Diese Regeln definieren beispielsweise, unter welchen Bedingungen eine Komponente in einer anderen verbaut werden kann.

Kann ein Baukastenelement ein anderes Baukastenelement im Sinne einer Komposition enthalten, das jedoch nicht zwingend erforderlich ist, sondern auf Kundenwunsch hinzugefügt oder weggelassen werden kann, so spricht man von einem optionalen Baukastenelement innerhalb einer Komposition, kurz von einer **Option**. Die dabei entstehenden unterschiedlichen Kompositionen werden als **Varianten** bezeichnet. Der Begriff Varianten wird jedoch ebenfalls für ähnliche Baukastenelemente verwendet, die als gleichberechtigte Alternativen an einer bestimmten Stelle innerhalb eines Produktes verbaut werden können und sich in mindestens einer Eigenschaft unterscheiden.

Beim Projektieren werden Baukastenelemente unter Berücksichtigung der vorgegebenen Regeln gemäß der definierten Baukastenarchitektur in konkreten Fertigungseinrichtungen verwendet. Parameterwerte in diesen projektspezifischen Betrachtungseinheiten (**Instanzen** der Baukastenelemente) können je nach Bedarf überschrieben werden.

2.3 Entwurfsmuster

Das Konzept der Entwurfsmuster entstand ursprünglich in den 1970er Jahren. Der Architekt Christopher Alexander entwickelte eine Muster-Sprache mit 253 Mustern, die jeweils zunächst ein immer wieder auftretendes Problem beschreiben, gefolgt vom Kern der Lösung. Ziel bei der Entwicklung der Muster war es, diese so allgemeingültig wie möglich zu halten, um sie millionenfach anwenden zu können, ohne sich je zu wiederholen [vgl. Alexander 1995].

1987 entwickelten Kent Beck und Ward Cunningham nach dem Vorbild Alexanders Entwurfsmuster für die Erstellung grafischer Benutzeroberflächen in Smalltalk.

Zum Durchbruch hat den Entwurfsmustern letztendlich Erich Gamma in seiner Dissertation „Objektorientierte Software-Entwicklung am Beispiel von ET++“ und dem späteren Standardwerk der Softwareentwicklung „Entwurfsmuster – Elemente wiederverwendbarer objektorientierter Software“ verholfen [vgl. Gamma 1992; Gamma 2004]. In letzterem Werk beschreiben die Autoren um Erich Gamma (Gang of Four) 23 Entwurfsmuster für die Lösung häufig vorkommender Probleme der Softwaretechnik (siehe auch Kapitel 5.2.2). Dabei enthält jedes Entwurfsmuster vier grundlegende Elemente: einen Musternamen, einen Problem-, einen Lösungs- und einen Konsequenzenabschnitt [vgl. Gamma 2004].

Die Erstellung von Baukästen für FES ähnelt aufgrund des objektorientierten Ansatzes von FES stark der Entwicklung von objektorientierter Software. Beispielsweise finden in beiden Bereichen die Paradigmen Abstraktion, Klassen und Instanzen oder Vererbung Anwendung. Daher lässt sich das Konzept der Entwurfsmuster aus der Softwareentwicklung auf die Erstellung von Baukästen für FES übertragen, um beispielsweise die Qualität der Baukästen zu erhöhen.

3 Problemstellung

In diesem Kapitel werden Probleme und Herausforderungen bei der Entwicklung von Baukästen für FES erläutert und in diesem Kontext Begriffe definiert, um anschließend Anforderungen für neue Methoden und Konzepte abzuleiten.

Die Entwicklung und Pflege von Baukästen für FES ist Stand heute zeitaufwendig und kostenintensiv. Die Ursachen für dieses Problem lassen sich in drei Hauptaspekte untergliedern (siehe Bild 3.1, rote Pfeile).

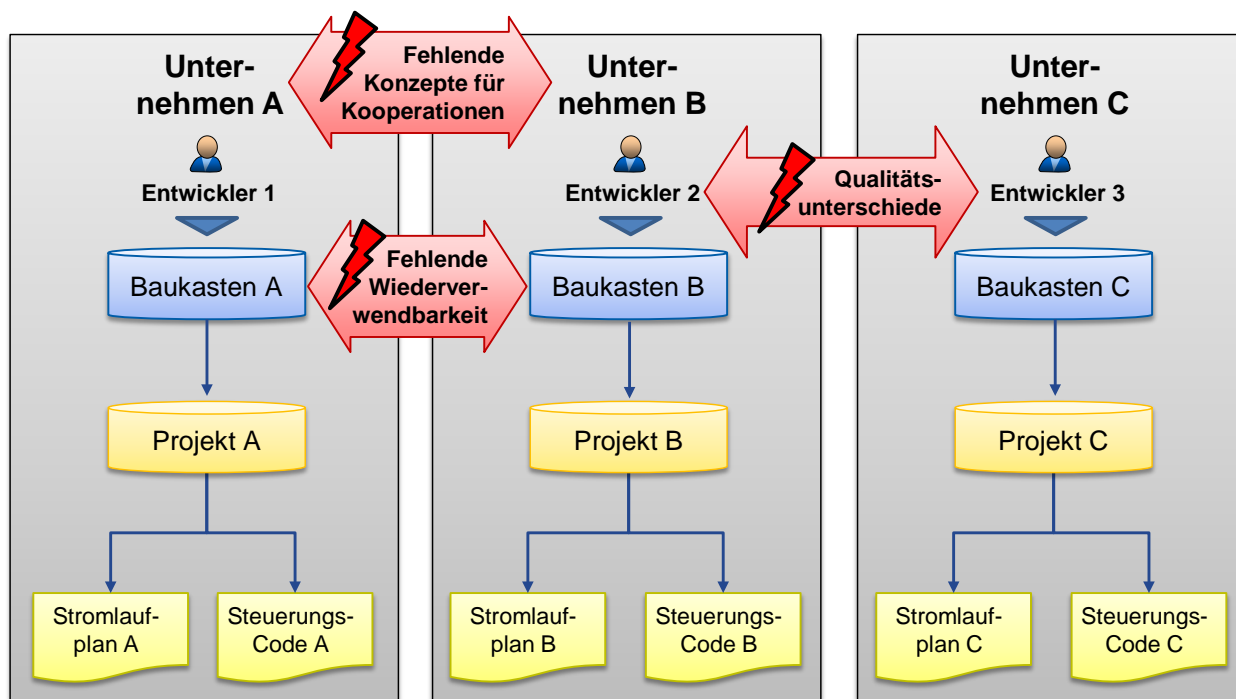


Bild 3.1: Problemstellung bei der Entwicklung, Nutzung und Pflege von Baukästen für FES

3.1 Qualitätsunterschiede von Baukästen

Diese Hauptaspekte lassen sich im Einzelnen wie folgt beschreiben:

- Auf Grund unterschiedlichen Erfahrungswissens der Baukastenentwickler kommt es zu Qualitätsunterschieden der Baukästen, was später beispielsweise zu Problemen bei Änderungen innerhalb der Baukästen führen kann.
- Baukästen werden heute üblicherweise von Grund auf neu erstellt, trotz immer wiederkehrender Komponenten und Baugruppen existieren keine wiederverwendbaren Baukästen.
- Bisher existieren keine Konzepte, wie Firmen unter Einsatz von FES kooperieren können, um die Vorteile von FES (siehe Kapitel 2.1) auch bei Zusammenarbeit mit Arbeitsteilung nutzen zu können.

Die drei genannten Hauptprobleme bei der Erstellung von Baukästen für FES werden im Folgenden genauer erläutert.

3.1 Qualitätsunterschiede von Baukästen

Bevor ein FES dazu genutzt werden kann, Maschinen oder Anlagen zu konfigurieren und anschließend Daten und Dokumente zu generieren, muss ein Baukasten mit allen dafür benötigten Baukastenelementen entwickelt werden. Diese Entwicklung des Baukastens wird von einem oder mehreren Baukastenentwicklern durchgeführt. Der bzw. die Baukastenentwickler haben – ähnlich wie in der klassischen Softwareentwicklung – viele Entscheidungen bei der Modellierung der Baukastenelemente zu treffen, beispielweise hinsichtlich Struktur, Hierarchisierung oder Granularität [vgl. Balzert 2001]. Ein Beispiel hierfür ist in Bild 3.2 zu sehen. Drei verschiedene Servomotoren können beispielsweise als konkrete Klassen einer abstrakten Oberklasse modelliert werden (oben) oder alternativ als eine einzige Klasse mit unterschiedlichen Parametersätzen.

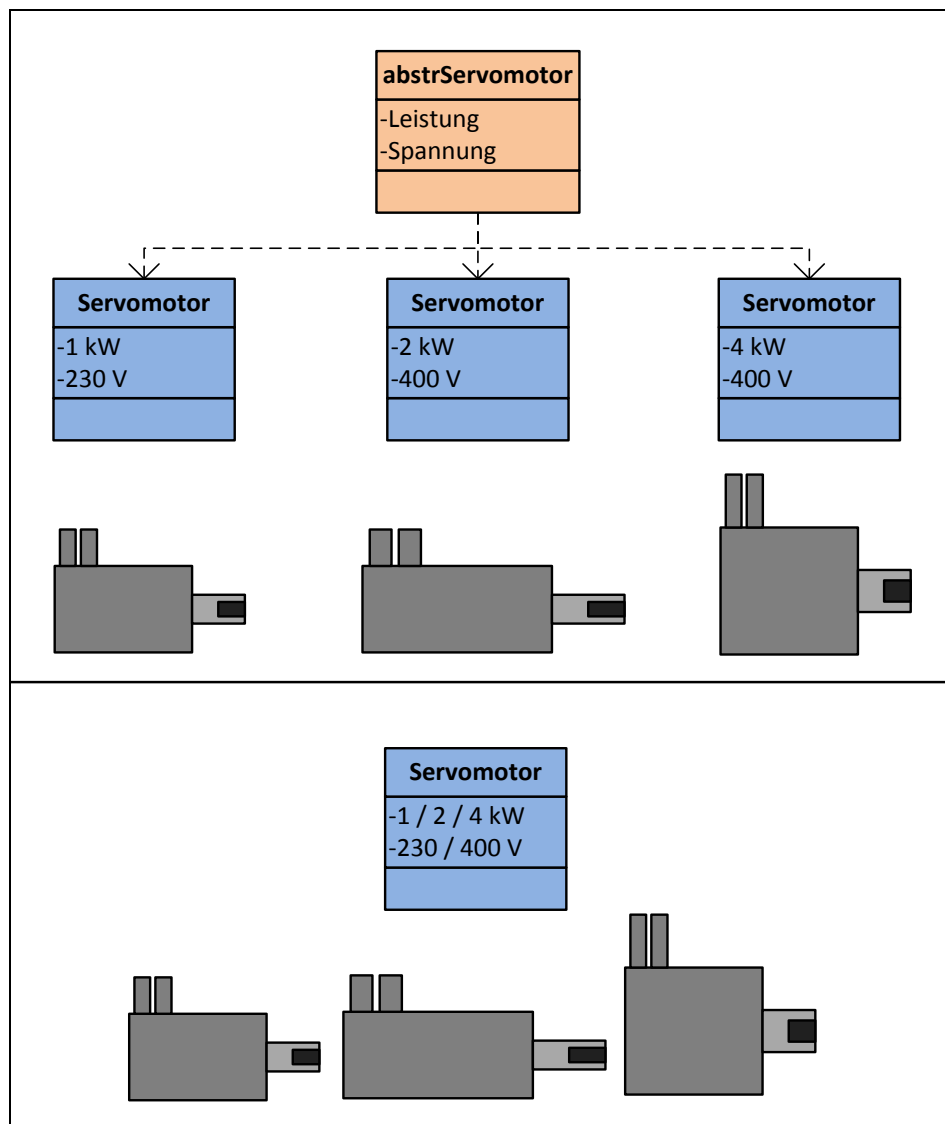


Bild 3.2: Unterschiedliche Modellierungen von Servomotoren

Je nach Erfahrungswissen und Vorlieben der Baukastenentwickler werden Baukästen folglich bei ähnlicher Problemstellung unterschiedlich in Bezug auf z.B. Struktur, Hierarchisierung oder Granularität erstellt. Das Ergebnis sind individuell erstellte Baukästen, die unterschiedlich leicht zu ändern, performant, übersichtlich und verständlich sind. Folglich gibt es keinen einheitlichen, reproduzierbaren Qualitätsstandard der Baukästen. Daher ist es für einen Baukastenentwickler schwierig, den Baukasten eines anderen Baukastenentwicklers zu übernehmen oder zu warten. Dafür ist vorher eine Übergabe bzw. eine zeitintensive Einarbeitung notwendig.

3.1 Qualitätsunterschiede von Baukästen

Bestehende Baukästen ändern sich im Laufe der Zeit auf Grund von Neuentwicklungen oder Änderungen im Produktportfolio. Bei grundlegenden Änderungen (bspw. Austausch der Steuerung einer Maschine oder Änderung der Schweißnetzspannung bei einer Anlage) ist jedoch häufig ein Neuaufbau des Baukastens notwendig, weil ein Umbau zu aufwendig wäre. Implementierungsprojekte von FES während der letzten Jahre haben gezeigt, dass erfahrene Baukastenentwickler bei Änderungen weniger Aufwand und somit geringere Änderungskosten haben als weniger erfahrene Kollegen.

Ein schematisches Beispiel hierfür wird in Bild 3.3 dargestellt. Bei einer Änderung in einem Baukasten, beispielsweise wenn eine Formel angepasst werden soll, muss im besten Fall nur eine einzige Klasse geändert werden, da die Formel auf alle anderen Klassen vererbt wird (oben dargestellt). Im schlechtesten Fall (unten dargestellt) muss die Formel in jeder nicht abstrakten Klasse geändert werden, was wesentlich aufwendiger und fehleranfälliger ist, weil Klassen übersehen oder beim wiederholten Ändern Flüchtigkeitsfehler gemacht werden könnten.

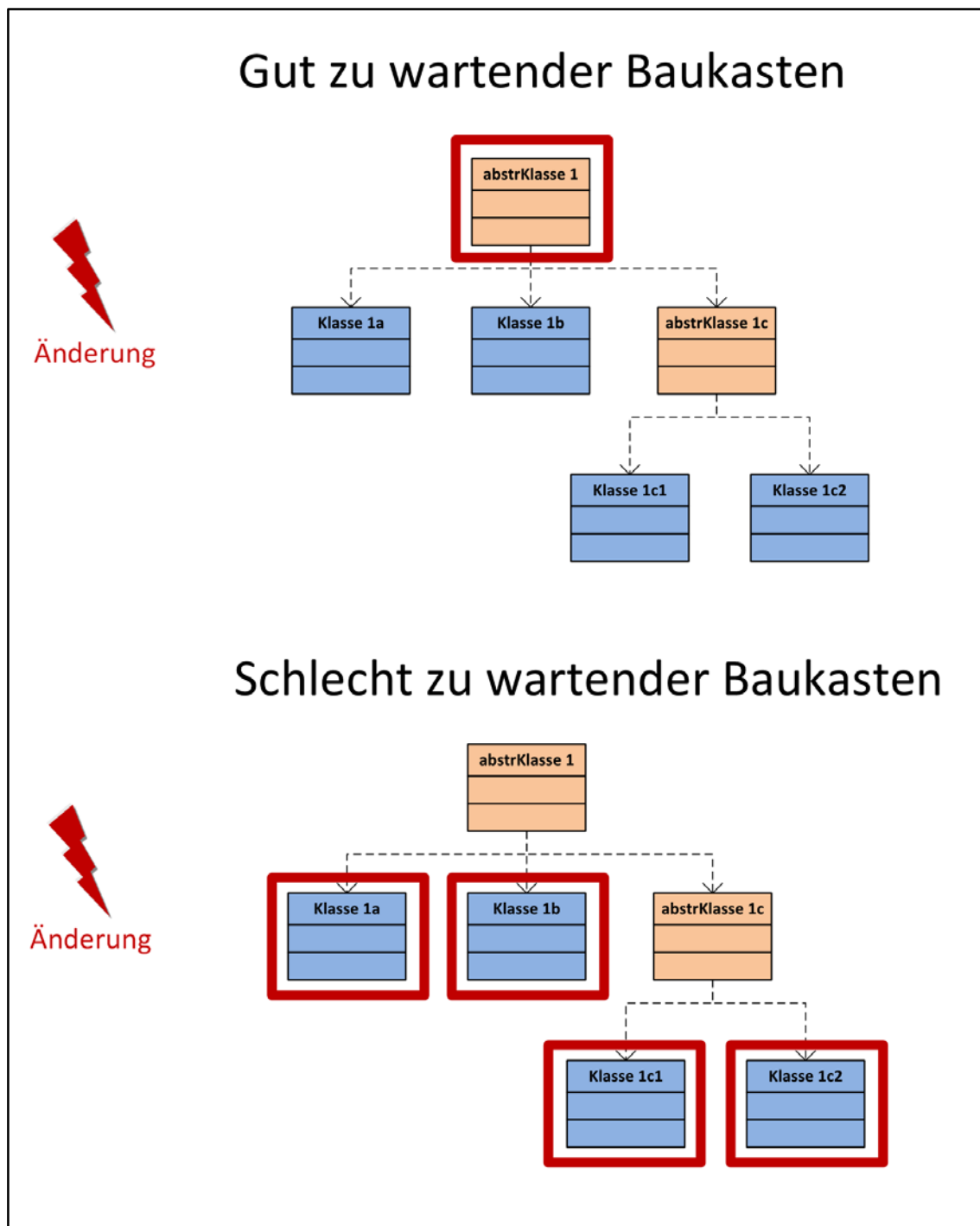


Bild 3.3: Beispiel eines gut (oben) bzw. schlecht (unten) zu wartenden Baukastens

Diese Qualitätsunterschiede bei Baukästen sind stand heute nicht messbar. Zwar kann festgestellt werden, dass manche Baukästen – ähnlich wie Bibliotheken in der

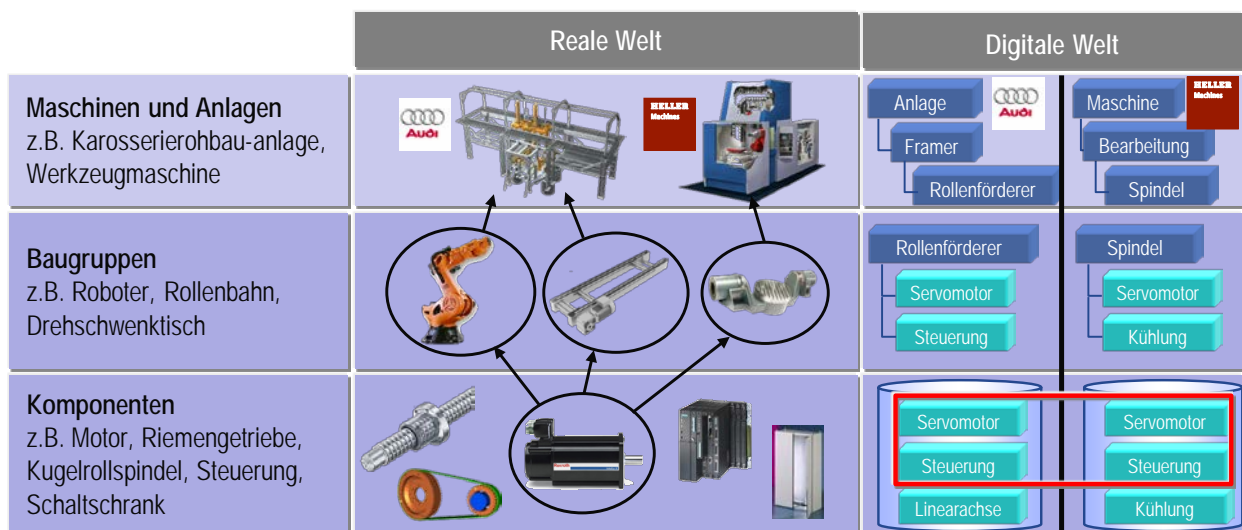
3.2 Fehlende Wiederverwendbarkeit von Baukästen

klassischen Softwareentwicklung – weniger performant sind als andere, oder größere Änderungsaufwände auftreten [vgl. Balzert 1998]. Eine quantitative Bewertung und eine Analyse der Ursachen ist jedoch nicht oder nur mit großen manuellen Aufwänden möglich.

3.2 Fehlende Wiederverwendbarkeit von Baukästen

Die Entwicklung von Baukästen für FES und der enthaltenen Komponenten ist zeitaufwendig und damit kostenintensiv, da sie auf der Basis vorhandener Daten und Strukturen stattfindet. Sämtliche Strukturen müssen disziplinübergreifend analysiert und umgesetzt sowie anschließend mit disziplinspezifischen (z.B. mechanischen, elektr(on)ischen, softwaretechnischen) Daten gefüllt werden. Auf Grund von Unterschieden, resultierend aus den jeweiligen Produkten der Firmen und der in der jeweiligen Branche üblichen Terminologie, erfolgt die Erstellung der Baukästen heute in jedem Unternehmen von Grund auf neu. Dies führt immer zu 100% firmenspezifischen Baukästen [vgl. Lewek 2005]. Ähnliches ist insbesondere auch im Bereich der Simulation zu beobachten, wo häufig disziplin- und domänenspezifische (z.B. Maschinenbau, Luft- und Raumfahrt, Automobilbau) Modelle entstehen [vgl. Voss 2012].

Vergleicht man die Produkte (reale Welt) und die FES-Baukästen (digitale Welt) mehrerer Unternehmen des Maschinen- und Anlagenbaus miteinander (siehe Bild 3.4), so wird ersichtlich, dass die gleichen mechatronischen Komponenten und Grundelemente (z.B. Servomotor oder Steuerung) in unterschiedlichen Produkten der realen Welt eingesetzt werden. In den Baukästen der digitalen Welt werden diese jedoch unabhängig voneinander mehrmals teilweise ähnlich oder identisch erstellt (roter Kasten rechts unten im Bild).



Bildquellen: Audi, Bosch-Rexroth, Heller, Kessler, Kuka, Rittal, Schnaitmann, Siemens

Bild 3.4: Ebenen der Wiederverwendung von Komponenten

Dadurch werden Arbeiten teilweise in verschiedenen Unternehmen mehrfach durchgeführt, was zu hohen Kosten bei der Erstellung von Baukästen führt. Die hohen Aufwände enden jedoch nicht mit dem Abschluss der Erstellung der Baukästen. Bei Produktneuerungen der Komponentenhersteller liegt der Pflegeaufwand der Baukästen bislang bei den Maschinen- und Anlagenbauern.

Unterschiede zwischen den Baukästen von Unternehmen haben ihre Ursache meist

- in der funktionalen Architektur der Produkte auf Grund unterschiedlicher Modulstrukturen
- der Namensgebung (Terminologie) von z.B. Funktionen, Komponenten, Signalen und Symbolen oder Örtlichkeiten
- der Gestaltung der Steuerungsprogramme
- der Dokumentation der Produkte in MCAD-/ECAD-/Fluid-Plänen, Maschindokumentationen bzw. Maschinenrichtlinien

3.3 Fehlende Konzepte für Kooperationen zwischen Firmen

Als Folge dieser Unterschiede gibt es Stand heute keine firmenübergreifend wiederverwendbaren Baukästen für FES, obwohl bei den in Maschinen und Anlagen verwendeten Komponenten ein hoher Gleichanteil festzustellen ist [vgl. Buck IV 2010].

3.3 Fehlende Konzepte für Kooperationen zwischen Firmen

Bei der Entwicklung von Maschinen oder Anlagen werden häufig Engineering-Anteile an Kooperationspartner ausgelagert [vgl. Bracht 2013], sodass mehrere Partner in unterschiedlichen FES am selben Projekt arbeiten (siehe Bild 3.5, Baukästen A und B, Projekte 1a und 1b). Dabei entstehen zusätzliche Datenflüsse in Form von Übergabedokumenten (z.B. Excel-, Word- oder PowerPoint-Dokumente). Das Beispiel zeigt einen Automobilhersteller, der bei der Planung einer Karosserierohbauanlage die Auswahl, Programmierung und Inbetriebnahme der Roboter sowie die Erstellung der Applikation (z.B. Schweißen, Kleben) an einen Zulieferer vergibt.

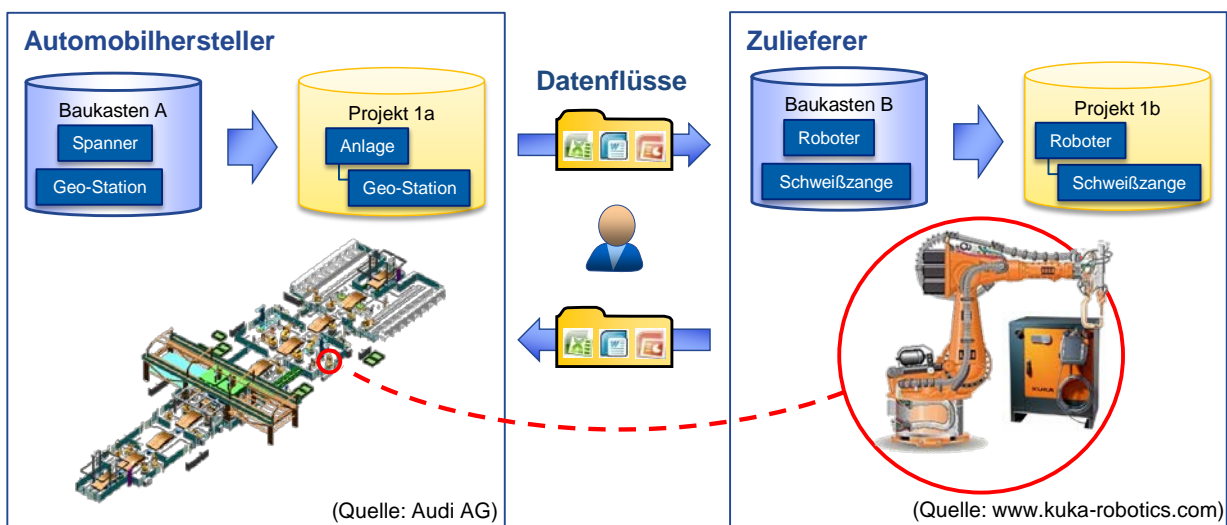


Bild 3.5: Manuelle Datenübergabe zwischen Automobilhersteller und Zulieferer

Das während der Kooperation entstehende zusätzliche Daten- und Kommunikationsaufkommen im Engineering-Prozess hat unterschiedliche Mehraufwände zur Folge:

- bei der **Festlegung von Schnittstellen** zwischen den Baukästen der Partner, die bei jeder Änderung eines der Baukästen angepasst werden müssen;
- beim Export aus dem Ausgangs-FES, der Übergabe von Konfigurationen und Parametersätzen in Form von Office-Dokumenten und der **manuellen Übernahme** ins Ziel-FES; auf Grund von Know-How-Schutz werden meist keine kompletten Baukästen oder Projekte weitergegeben, sondern es findet eine starke **Vereinfachung bzw. Befreiung von Details** (z.B. Regel- und Auslegungswissen) statt;
- bei der **manuellen Zusammenführung** der von verschiedenen Zulieferern erstellten Daten und Dokumente zu einem Gesamtmodell bzw. –Dokument.

Neben den erhöhten Kosten resultieren aus den oben genannten Mehraufwänden auch Zeitverzögerungen sowie Fehler, da Daten mehrfach manuell kopiert und in den Zielsystemen eingefügt werden.

Die Art der Zusammenarbeit und die Probleme, die bei den jeweiligen firmenübergreifenden Engineering-Prozessen entstehen, sind abhängig von der jeweiligen Ebene, in der die Kooperationspartner interagieren. Die heute üblichen Arbeitsweisen innerhalb überbetrieblicher Kooperationen mit FES und die daraus resultierenden Aufwände bei Kommunikation und Datenzusammenführung sind in Bild 3.6 zu sehen [vgl. Buck II 2012].

3.3 Fehlende Konzepte für Kooperationen zwischen Firmen

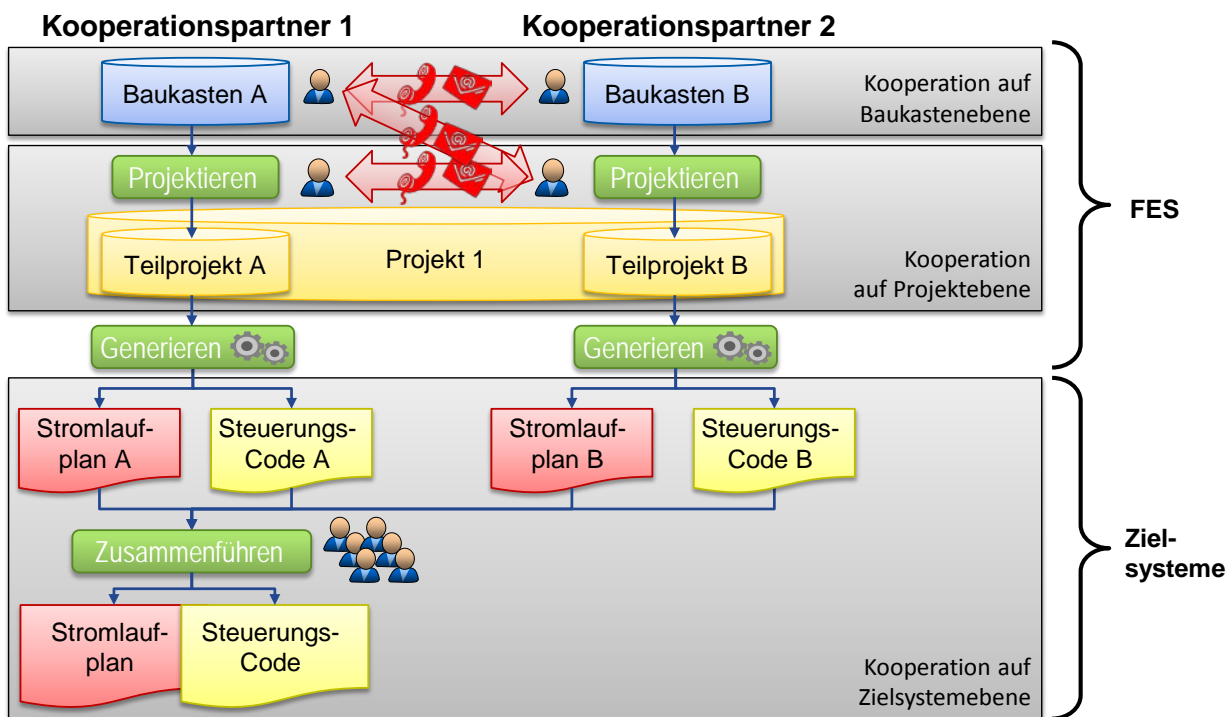


Bild 3.6: Aufwände für Kommunikation und Datenzusammenführung bei Kooperationen

Innerhalb der FES kann zwischen Baukasten- und Projektebene unterschieden werden. Auf Baukastenebene werden Baukästen mit mechatronischen Komponenten entwickelt, die auf Projektebene zu Maschinen oder Anlagen zusammengestellt (projektiert) werden. Anschließend werden Zieldaten wie Stromlaufplan oder Steuerungs-Code generiert, die in den Zielsystemen (z.B. ECAD-System oder Programmierumgebung) ergänzt oder zu Gesamtdokumenten bzw. –Daten zusammengeführt werden. Dabei ist grundsätzlich zwischen folgenden, typischerweise vorzufindenden Arbeitsweisen zu unterscheiden (rote Pfeile im Bild):

- Bei **Kooperationen auf Baukastenebene** werden bei der Erstellung der Baukästen Schnittstellen definiert, was häufig mit hohen Kommunikationsaufwänden zwischen den Kooperationspartnern und redundanter Erstellung und Pflege von Regelwissen verbunden ist.

- Werden Baukästen unabhängig voneinander erstellt, während der Projektierung jedoch Daten über Konfigurationen und Parametrierungen ausgetauscht, so spricht man von einer **Kooperation auf Projektebene**. Hierbei werden üblicherweise Daten manuell in Form von Excel-Tabellen übertragen, der damit verbundene Kommunikationsaufwand zwischen den Kooperationspartnern ist hoch.
- Der dritte Fall, die **Kooperation auf Zielsebene**, ist meist der ineffektivste und fehleranfälligste Prozess: Daten wie Stromlaufplan oder Steuerungs-Code werden unabhängig voneinander erzeugt und im letzten Arbeitsschritt manuell zusammengeführt. Dabei sind oftmals Rückfragen an alle partizipierenden Projektoren notwendig, weil Schnittstellen in der Regel vorher unzureichend definiert werden und die Integration von Teildokumenten in Gesamtdokumente dadurch unnötig kompliziert wird.
- Zusätzlich gibt es noch eine weitere Arbeitsweise, die **Kooperation zwischen Baukasten- und Projektebene** (siehe diagonaler roter Pfeil), bei der ein Kooperationspartner die Erstellung des Baukastens vornimmt und diesen dem anderen für die Projektierung zur Verfügung stellt. Aus Gründen des fehlenden Know-How-Schutzes findet diese Art der Kooperation bislang jedoch selten Anwendung.

Für Kooperationen auf Baukasten- bzw. Projektebene sowie dazwischen werden Lösungskonzepte benötigt, um manuelle und damit kostenintensive Datenübernahmen so weit wie möglich zu eliminieren und Kooperationen auf Zielsebene überflüssig zu machen.

3.4 Resümee

Für die Kategorisierung der Konzepte im Weiteren lassen sich Kooperationen in drei Arten untergliedern (siehe Bild 3.7).

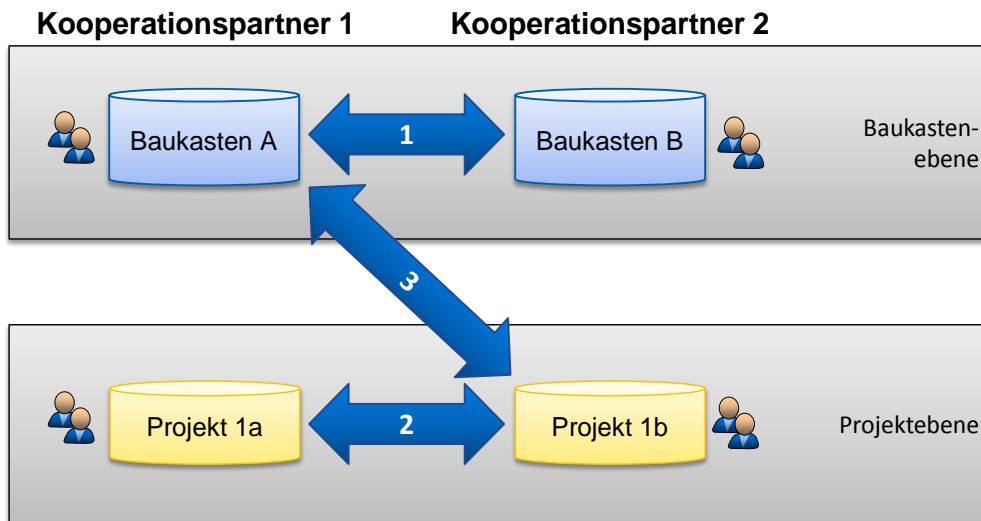


Bild 3.7: Kooperationsarten

Diese drei Kooperationsarten werden im Folgenden benannt als:

- Kooperationen auf Baukastenebene (1)
- Kooperationen auf Projektebene (2)
- Kooperationen zwischen Baukasten- und Projektebene (3)

Für alle drei Kooperationsarten werden im nächsten Kapitel Anforderungen definiert, auf deren Grundlage dann Lösungskonzepte erarbeitet werden.

3.4 Resümee

Bei der Erstellung, Nutzung und Pflege von Baukästen für FES können drei Hauptprobleme beobachtet werden, die zu einer Erhöhung der Kosten, zu zeitlichen Verzögerungen und zu Qualitätsproblemen führen können. Aus diesen drei oben beschriebenen Problemen werden im Folgenden Anforderungen an mögliche Lösungskonzepte abgeleitet.

4 Anforderungen an Baukästen für FES

Die aus der Problemstellung abgeleiteten Anforderungen lassen sich wie folgt zusammenfassen und gliedern.

4.1 Qualitätssicherung bei der Erstellung von Baukästen

An die Erstellung von Baukästen für FES werden folgende Anforderungen gestellt:

- Es soll eine Bewertung vorhandener Baukästen in Hinblick auf Performanz, Änderbarkeit und Grad der Wiederverwendung möglich sein. Diese Eigenschaften definieren sich im Umfeld von FES analog zum Bereich der Softwaretechnik folgendermaßen:
 - **Performanz:** Angemessene Leistung relativ zu den benötigten Ressourcen, wobei die volle Funktionalität unter festgelegten Bedingungen bereitgestellt wird [vgl. Balzert 2011].
 - **Änderbarkeit:** Erforderlicher Aufwand für die Durchführung vorgegebener Änderungen an Baukästen. Dabei sind die Aspekte Analysierbarkeit, Modifizierbarkeit, Stabilität und Testbarkeit von Bedeutung [vgl. Grechenig 2010].
 - **Grad der Wiederverwendung:** Wiederverwendung von Baukastenelementen ist das erneute Anwenden von bei der Entwicklung eines Baukastens entstandenen Artefakten und angesammeltem Wissen bei der Entwicklung beispielsweise eines neuen Moduls oder Maschinentyps, um den Aufwand zur Erstellung und Pflege dieses neuen Moduls oder Maschinentyps zu reduzieren [angelehnt an Biggerstaff 1989].

4.2 Wiederverwendbarkeit von Baukästen

- Die heute oftmals vorhandene hohe Komplexität von FES-Baukästen soll so weit reduziert werden, dass es möglich ist, einen Baukasten ohne große Einarbeitungsaufwände an einen anderen Baukastenentwickler zu übergeben.
- Die Erstellung von Baukästen soll standardisiert werden, um die Reproduzierbarkeit und die gleichbleibende Qualität von Baukästen sicherzustellen.
- FES-Baukästen sollen so erstellt werden, dass später die Änderbarkeit gewährleistet ist bei
 - Erweiterung des Baukastens um neue Disziplinen (Bsp. ECAD)
 - Wechsel der Technologie (Bsp. Kleben statt Schweißen)
 - Änderungen in der Infrastruktur (Bsp. anderes Bussystem)

4.2 Wiederverwendbarkeit von Baukästen

Um den Aufbau von Baukästen schneller und kostengünstiger zu realisieren, bedarf es der Senkung der Aufwände bei der Erstellung eines Baukastens. Dafür ist es notwendig, das Einbinden firmenübergreifender Basisbaukästen zu ermöglichen und solche Baukästen zur Verfügung zu stellen.

4.3 Unterstützung von Kooperationen zwischen Firmen

Für die Unterstützung von Kooperationen zwischen Firmen auf den drei unterschiedlichen Kooperationsebenen existieren jeweils unterschiedliche Anforderungen. Bei der gemeinsamen Entwicklung von Baukästen sind dies

- die Gewährleistung von **Know-How-Schutz**, um firmeninternes Wissen vor dem Zugriff durch Mitarbeiter anderer Unternehmen zu schützen;
- die Ermöglichung von **räumlich getrenntem und zeitversetztem Arbeiten**, da die Kooperationspartner oftmals über Ländergrenzen hinweg und in unterschiedlichen Zeitzonen arbeiten;
- die Ermöglichung von **parallelem Arbeiten**, falls mehrere Entwickler zur selben Zeit am selben Baukasten arbeiten.

Beim Austausch von Projektdaten ist es darüber hinaus notwendig, dass

- diese **ohne manuelle Tätigkeiten oder Konvertierungen zusammengeführt** werden können, um Prozesse robust gegenüber Fehlern und gleichzeitig effizient gestalten zu können;
- die Verwendung **unterschiedlicher Engineering-Systeme** möglich ist, da nicht davon ausgegangen werden kann, dass bei jedem Kooperationspartner dasselbe FES vorhanden ist.

Bei der Bereitstellung von Baukästen für die Projektierung ist außerdem darauf zu achten, dass

- **Updates automatisch** verbreitet werden, damit alle Partner stets auf dem aktuellen Stand der Baukästen arbeiten;
- je nach Anwendungsfall eine **Online- oder Offline-Zusammenarbeit** oder beides ermöglicht wird, um ein effizientes Arbeiten mit und ohne Internet-Zugang zu gewährleisten.

Im Folgenden wird ein Überblick über den Stand der Technik gegeben und dieser den hier genannten Anforderungen gegenübergestellt.

5 Stand der Technik und der Forschung

Da im Bereich des Funktionalen Engineerings bislang nur wenige Ansätze zur Erfüllung der genannten Anforderungen existieren, werden an dieser Stelle auch Konzepte aus den Disziplinen Mechanik, Elektr(on)ik, Fluidik und Software aufgeführt. Dabei erstrecken sich die Recherchen von Methoden zur Analyse und Erstellung von Baukästen über Verfügbarkeit von Basisbaukästen bis hin zu Systemen und Austauschformaten für Kooperationen zwischen Firmen.

5.1 Methoden für die Analyse von Baukästen

5.1.1 Software-Metriken

Im Bereich der Softwareentwicklung sind Metriken weit verbreitet. Nach Balzert definiert eine Software-Metrik, wie eine Kenngröße eines Software-Produkts oder eines Software-Prozesses gemessen wird. Software-Metriken werden dazu verwendet, um quantitative Aussagen über die Qualität einer Software tätigen zu können. Sie sollen helfen, Entwicklungsprozesse zu kontrollieren und planbar zu machen [vgl. Balzert 1998].

Beispiele für Metriken für objektorientierte Software sind Vererbungstiefe im Hierarchiebaum, Anzahl der Unterklassen (nach Chidamber und Kemerer, siehe Bild 5.1) oder gewichtete Menge an Variablen (nach Sharble und Cohen) [vgl. Ebert 1996; Sneed 2010].

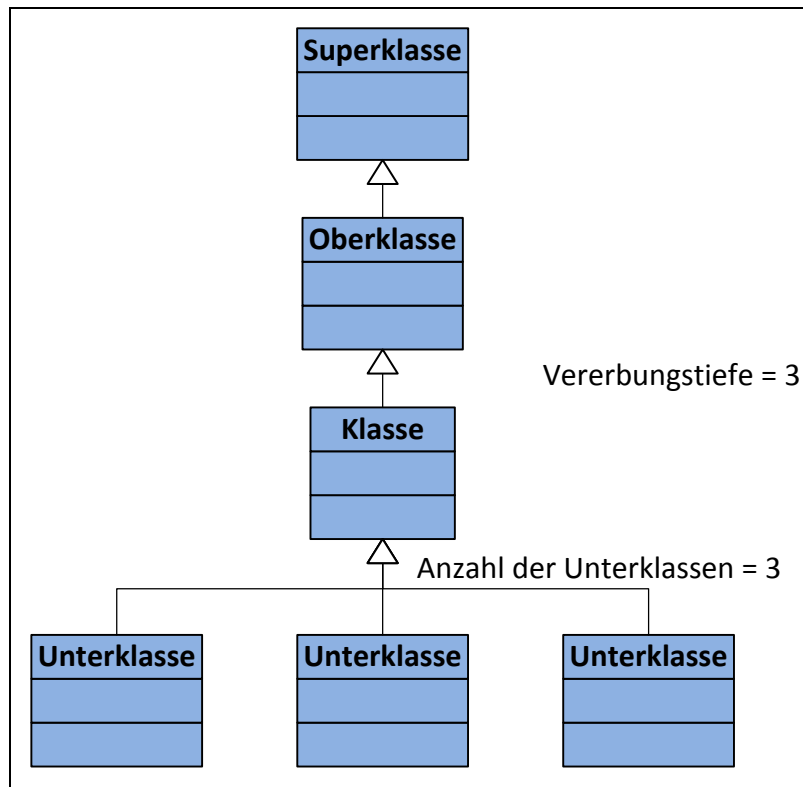


Bild 5.1: Objektorientierte Entwurfsmetriken nach Chidamber und Kemerer [angelehnt an Sneed 2010]

Die in der Softwareentwicklung üblichen Metriken sind zwar nicht eins zu eins auf FES übertragbar. Dadurch, dass FES häufig die Konzepte der Objektorientierung verwenden, ist es jedoch möglich, vorhandene Metriken auf FES anzupassen bzw. neue Metriken zu entwerfen, die die Anforderungen bei der Beurteilung von Baukästen erfüllen.

5.2 Methoden für die Erstellung von Baukästen und Bibliotheken

5.2.1 Parametrische Baugruppen

Das Konzept der Baukastensysteme wird in der Mechanik schon seit vielen Jahren erfolgreich angewandt. Kennzeichen der Baukästen sind dabei

- der Bausteininventar, bestehend aus gleichen und verschiedenen Bausteinen;

- die Vielzahl von Kombinationsmöglichkeiten;
- die Zerlegbarkeit der aus Bausteinen aufgebauten Gebilde;
- die Wiederverwendbarkeit der Bausteine;
- die einheitlichen Passflächen oder Passstellen an jedem Baustein [vgl. Borowski 1961].

Moderne MCAD-Werkzeuge unterstützen verschiedene Mechanismen wie z.B. parametrische Bauteile oder Baugruppen. Abhängig von den eingegebenen Parameterwerten (z.B. Länge und Breite) verändern sich beispielsweise Position und Anzahl von Bohrungen. Die parametrischen Bauteile oder Baugruppen sind dabei nur einmal in der Datenablage (z.B. im PDM-System) vorhanden. Werden sie im Rahmen einer Konstruktion verwendet, wird lediglich eine Referenz zu der Baugruppe oder dem Bauteil gespeichert. Dadurch sinkt das Datenvolumen von MCAD-Daten bei gleichzeitig besserer Änderbarkeit und Übersichtlichkeit.

Diese Mechanismen sind zwar die Grundlage für das baukastenbasierte Funktionale Engineering, sie geben jedoch keine methodische Hilfestellung für die Qualitätssicherung bei der Erstellung von Baukästen, die Wiederverwendbarkeit oder die Unterstützung von Kooperationen.

5.2.2 Entwurfsmuster

In der Softwaretechnik ist das Konzept der Entwurfsmuster weit verbreitet. Wie in Kapitel 2.3 beschrieben, handelt es sich bei Entwurfsmustern um allgemein formulierte Lösungsansätze, die dabei helfen, immer wiederkehrende Problemstellungen zu lösen. Neben den von Erich Gamma et al. entwickelten 23 Entwurfsmustern für den Entwurf objektorientierter Software gibt es inzwischen eine Vielzahl weiterer Entwurfsmuster, die von einer großen Anzahl von Entwicklern angewendet, diskutiert und weiterentwickelt werden [vgl. Eric 2008; Hillside 2012; Gamma 2004]. Dies führt dazu, dass Entwurfsmuster heute in der Softwareentwicklung unerlässlich sind und dabei helfen, Software ingenieurmäßig zu entwickeln [vgl. Reuse 2008].

Entwurfsmuster unterstützen den Softwareentwickler dabei, Programmcode so zu erstellen, dass er gleichzeitig wiederverwendbar und dennoch flexibel und wartungsarm ist. Durch die Anwendung weit verbreiteter Entwurfsmuster ist es für Entwickler ohne großen Aufwand möglich, sich im Code eines Kollegen zurechtzufinden. Bestimmte Muster gewährleisten den nachträglichen Austausch von Klassen und Methoden und ermöglichen ebenfalls verteiltes Arbeiten an ein und derselben Software [vgl. IBM 2012].

Das Konzept der Entwurfsmuster erfüllt zwar beinahe alle Anforderungen, durch das Fehlen mechatronischer Entwurfsmuster für das Funktionale Engineering und der völlig unterschiedlichen Begriffswelt der vorhandenen Entwurfsmuster aus dem Bereich der Softwaretechnik ist die direkte Anwendung innerhalb von FES jedoch nicht möglich.

5.2.3 Variantenmanagement

Ein anderer Ansatz, der für sich beansprucht, mechatronisch zu sein ist das Variantenmanagement. Es umfasst die Entwicklung, Gestaltung und Strukturierung von Produkten und Dienstleistungen bzw. Produktsortimenten im Unternehmen. Dadurch wird angestrebt, die vom Produkt ausgehende Komplexität (z.B. Anzahl Teile, Komponenten oder Varianten) wie auch die auf das Produkt einwirkende Komplexität (z.B. Marktdiversifikation oder Produktionsabläufe) mittels geeigneter Instrumente zu bewältigen [vgl. Schuh 2005].

Der Ansatz des Variantenmanagements liefert zwar Methoden zur Bewältigung der Komplexität; jedoch gibt er keine Antworten auf die Fragen, wie Baukästen erstellt werden müssen, damit sie änderbar, wiederverwendbar oder in Kooperationen nutzbar sind.

5.2.4 Funktionales Engineering

Ein weiterer Ansatz zur Lösung der genannten Anforderungen ist das Funktionale Engineering. Dabei handelt es sich ebenfalls um einen mechatronischen Ansatz mit dem Ziel, Komplexität von Produkten beherrschbar zu machen. Dazu wird ein Baukasten mit standardisierten, wiederverwendbaren Baukastenelementen erstellt, die dann in der Auftragsbearbeitung in verschiedenen Projekten benutzt werden (siehe Kapitel 2.1) [vgl. Litto 2004].

Doch auch dieser Ansatz liefert keine Antwort auf die Fragen, wie Baukästen zu erstellen sind, damit sie später änderbar und firmenübergreifend einsetzbar sind.

5.3 Firmenübergreifende Basisbaukästen

5.3.1 Mechanik-Normteilbibliotheken

Viele Hersteller von MCAD-Systemen bieten Normteilbibliotheken in offenen Datenformaten an. Bei der Erstellung neuer Maschinen und Anlagen können vorhandene Bauteile und Baugruppen verwendet werden. Dabei werden keine Daten kopiert, sondern lediglich Referenzen auf die Originaldaten angelegt. Dadurch werden in Projekten verwendete Bauteile oder Baugruppen bei Änderungen automatisch aktualisiert.

Die oben genannten Konzepte können nicht vollständig auf die Mechatronik übertragen werden. In der Mechatronik hat erfahrungsgemäß jede Disziplin ihre eigene Sicht auf ein und dasselbe Produkt [vgl. Hackel 2011]. Daher ist die Darstellung eines mechatronischen Produkts in unterschiedlichen Strukturen und Hierarchien sowie eine ständige Transformation zwischen den einzelnen Sichten notwendig. Aus diesem Grund bieten die genannten Ansätze der Mechanik keine Lösung auf die Anforderung an den Aufbau mechatronischer Baukästen.

5.3.2 Norm-Symbolbibliotheken für die Elektr(on)ik und Fluidik, frei verfügbare Schaltplanmakros

Konstrukteure der Elektrotechnik und Fluidik nutzen in ihren Softwarewerkzeugen häufig einen Makromechanismus. In Makrobibliotheken werden unternehmensweit standardisierte Symbole oder Teilschaltungen (Makros) gesammelt und gepflegt. Diese Makros werden entweder firmenspezifisch erstellt oder aber von im Internet verfügbaren Portalen wie beispielsweise dem Eplan-Dataportal bezogen. Des Weiteren stellen ECAD-System-Hersteller umfangreiche Norm-Symbolbibliotheken zum Download bereit.

Da auch der hier beschriebene Ansatz der Elektr(on)ik und Fluidik analog zu dem der Mechanik zu sehen ist und keinerlei Lösungen für die Belange der Mechatronik (unterschiedliche Strukturen/Hierarchien/Sichten und Transformationen dazwischen) bietet, ist auch hier eine einfache Übertragung auf die Mechatronik nicht möglich.

5.3.3 Softwarebibliotheken

Aufbauend auf den Entwurfsmustern existieren in der Softwaretechnik umfangreiche Open-Source-Bibliotheken. Im Framework Eclipse können beispielsweise so genannte Plugins genutzt werden [vgl. Eclipse 2012], im Bereich der SPS-Programmierung existieren umfangreiche Bausteinbibliotheken (z.B. PLC-Open [vgl. PLCopen 2012]). Die Erstellung und Pflege dieser Bibliotheken geschieht firmenübergreifend durch Communities.

Da die Begrifflichkeiten und Denkweisen in der Softwaretechnik sich jedoch stark von denen im Bereich der Mechatronik beispielsweise im Maschinen- und Anlagenbau unterscheiden, ist es nicht möglich, die vorhandenen Entwurfsmuster und bestehende Systeme für wiederverwendbare Softwarebibliotheken in der Mechatronik zu verwenden.

5.4 Datenformate und Systeme für den Austausch von Informationen

In diesem Kapitel werden die heute verbreiteten Ansätze dargestellt, die üblicherweise bei Kooperationen zwischen Unternehmen beim Informationsaustausch im Engineering vorgefunden werden können. Neben Ansätzen unter Einsatz von FES werden auch der Informationsaustausch mit Hilfe von Handbüchern und Listen, disziplinspezifische Baukastensysteme sowie PDM-/PLM-Systeme (PDM = Product Data Management, PLM = Product Lifecycle Management) betrachtet, um ein möglichst breites Spektrum firmenübergreifender Engineering-Prozesse analysieren zu können.

Die Ansätze lassen sich den in Kapitel 3.3 definierten Kooperationsarten (siehe Bild 5.2, Pfeile 1-3) wie folgt zuordnen:

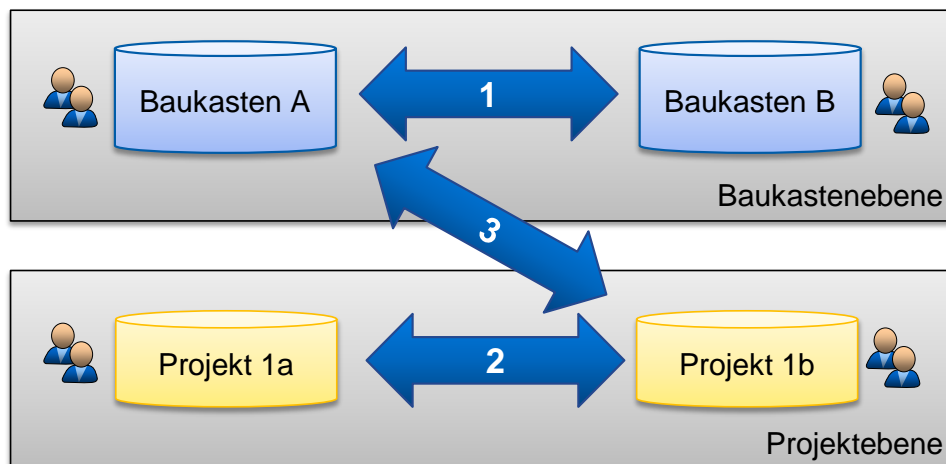


Bild 5.2: Kooperationen auf Baukastenebene (1), auf Projektebene (2) sowie zwischen Baukasten- und Projektebene (3)

- Übergabe von Handbüchern mit Konstruktions-/Entwicklungsrichtlinien (3)
- Manuelle Übergabe von Konfigurationsdaten in Form von Tabellen (Word, Excel, CSV, ProList) (2)
- Weitergabe von disziplinspezifischen herstellerübergreifenden Austauschformaten (1 & 3)

- Übergabe von Projektdaten in Form von mechatronischen, herstellerübergreifenden Austauschformaten (CAEX, AutomationML) (2)
- Übergabe von nativen Daten (1 & 3)
- Zusammenarbeit mit Hilfe von PDM-/PLM-Systemen (1)

Diese Ansätze werden im Folgenden genauer erläutert. Die Reihenfolge spiegelt den Grad der Formalität bzw. der Möglichkeit zu einer automatischen Übernahme der Daten wieder, beginnend mit informellen Daten wie Handbüchern bis hin zu nativen Datenformaten. Zuletzt werden PDM-/PLM-Systeme betrachtet.

5.4.1 Übergabe von Handbüchern mit Konstruktions- / Entwicklungsrichtlinien

Bei der Auslagerung von Engineering-Tätigkeiten an externe Dienstleister werden oftmals Handbücher mit Konstruktions- bzw. Entwicklungsrichtlinien weitergegeben. Diese Richtlinien schreiben dem Dienstleister vor, nach welchen Vorgaben oder Standards Daten und Dokumente zu erstellen sind.

Bei dieser Art der Informationsweitergabe zwischen Entwicklung (des Auftraggebers) und Projektierung (des Dienstleisters) (3) handelt es sich nicht um ein Datenformat, über das Informationen automatisiert weitergegeben werden können. Dies hat zur Folge, dass Wissen informell weitergegeben und von den Entwicklern wiederum manuell auf die Zielsysteme übertragen werden muss, was in der Regel zeitintensiv und fehleranfällig ist.

5.4.2 Manuelle Übergabe von Konfigurationsdaten in Form von Tabellen

Etwas formaler und konkreter ist der Austausch auf Projektebene (2) von Daten wie Aufträgen, Pflichtenheften, Konfigurationen und Parametersätzen beispielsweise in Form von Excel-, Word-, CSV-Dateien oder im ProList-Format. Die enthaltenen Daten werden meist manuell in die jeweiligen Zielsysteme übernommen.

Die Übergabedaten sind zwar mehr oder weniger formal festgehalten, durch die oftmals manuelle Übernahme in nachgelagerte Systeme entstehen jedoch neben hohen Zeitaufwänden häufig auch Fehler.

5.4.3 Disziplinspezifische, herstellerübergreifende Austauschformate

In den einzelnen Disziplinen Mechanik, Elektr(on)ik, Fluidik und Software gibt es häufig Ansätze, die den Kooperationspartnern die Zusammenarbeit auf Baukastenebene (1) und zwischen Entwicklung und Projektierung (3) ermöglichen. Hierbei werden herstellerübergreifende Bibliotheken mit in Projekten zu verwendenden Komponenten und Modulen zur Verfügung gestellt. In der Mechanik sind dies beispielsweise Bibliotheken mit standardisierten Bauteilen und –Gruppen; in der Elektr(on)ik und Fluidik gibt es die Möglichkeit, standardisierte Makrobibliotheken zu erstellen; in der Softwareentwicklung können z.B. Bibliotheken mit standardisierten FBs und FCs weitergegeben werden.

Die Bibliotheken werden in offenen Datenformaten gespeichert, wie z.B. IEC 61131 oder PLC-Open [vgl. PLCopen 2012] im Bereich von Steuerungsprogrammen oder STEP und IGES im Bereich von MCAD-Daten. Da es sich dabei jedoch nicht um native Datenformate handelt, ist der Export und Import der Daten von einem System in ein anderes mit aufwendigen manuellen Nacharbeiten und Verlust von Details behaftet. Know-How-Schutz wird dadurch erreicht, dass Bauteile oder Baugruppen so stark vereinfacht werden, dass nur noch die äußere Hülle existiert und somit der Bauraum die einzige verbleibende Information ist.

5.4.4 Mechatronische, herstellerübergreifende Austauschformate

Einen mechatronischen Ansatz für Kooperationen auf Projektebene (2) liefern CAEX und das AutomationML-Konsortium: die Übergabe von Daten erfolgt in CAEX oder im auf CAEX basierenden AutomationML-Format und der Import findet mit Hilfe von Mapping-Dateien statt, gefolgt von einer oftmals manuellen Integration der Daten in Gesamtmodelle [vgl. Fedai 2006; Drath 2010].

5.4 Datenformate und Systeme für den Austausch von Informationen

Die Organisation AutomationML stellt zwar keine wiederverwendbaren Baukästen zur Verfügung, unterstützt jedoch durch die Bereitstellung standardisierter Austauschformate beim Datentransfer von Projekten zwischen verschiedenen Softwaresystemen unterschiedlicher Disziplinen.

Das Datenformat basiert auf XML und beinhaltet im Bereich der 3D-Geometrie- und Kinematik-Daten das Format COLLADA und für die Weitergabe von SPS-Code das Format PLCopenXML (siehe Bild 5.3). Für den Austausch weiterer Daten wie z.B. Elektro- und Hydraulikdaten sind Erweiterungen in Arbeit [vgl. Drath 2010].

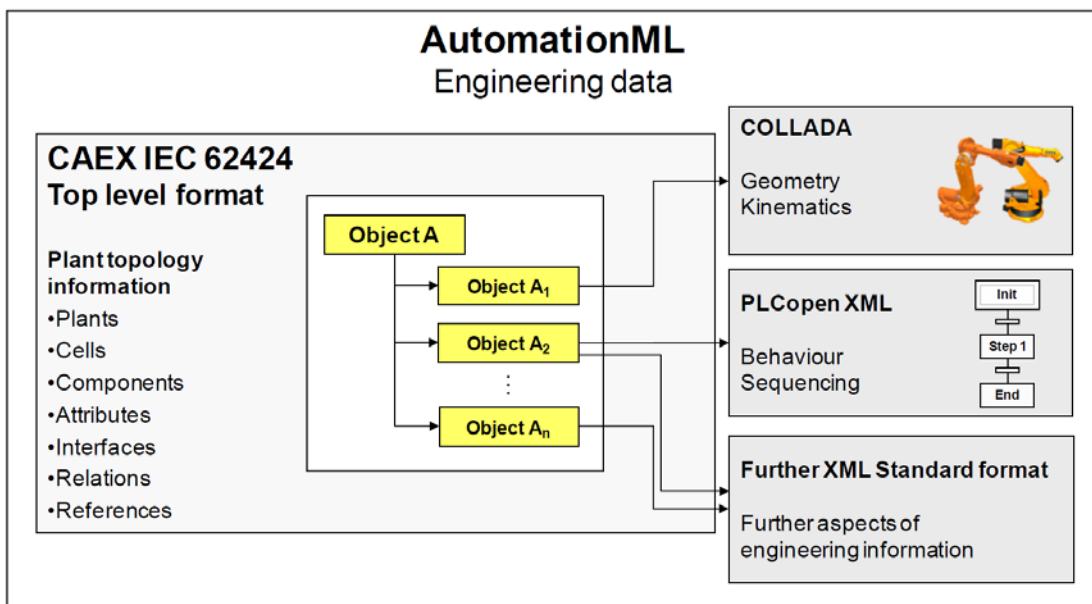


Bild 5.3: Übersicht über das AutomationML Datenformat [vgl. Drath 2010]

Die herstellerübergreifenden disziplinspezifischen bzw. mechatronischen Austauschformate zeigen Defizite bei der automatischen Weiterverarbeitung der Daten in den Zielsystemen. Weiterhin ist ihre Anwendung auf die Projektebene begrenzt, da lediglich konkrete Instanzen ausgetauscht werden können. Oftmals sind manuelle Nacharbeiten nach der Übernahme der Daten erforderlich, da die Daten in ein Übergabeformat konvertiert wurden oder Schnittstellenprobleme (beispielsweise beim

Mapping von Daten) auftreten. Teilweise ist darüber hinaus auch der Know-How-Schutz nicht gewährleistet.

5.4.5 Native Daten

Werden im Maschinen- oder Anlagenbau Teilprojekte an Dienstleister vergeben, so werden nach Fertigstellung der Engineering-Pakete häufig die Schalt- und Fluidplan-Fragmente in Gesamtdokumente eingefügt. Meist geschieht dies als White-Box, deren Schnittstellen manuell mit den entsprechenden Anschlüssen im umgebenden Dokument verbunden werden. Spätere Iterationen während der Projektbearbeitung sind immer mit hohen manuellen Pflegeaufwänden verbunden.

Der Austausch nativer Daten hat zwar den Vorteil, dass diese größtenteils ohne Konvertierungsaufwände in die Zielsysteme übernommen werden können, jedoch geht damit der Know-How-Schutz verloren.

5.4.6 Zusammenarbeit mit Hilfe von PDM-/PLM-Systemen

Weiterhin existieren vor allem im Bereich der Mechanik Lösungen auf Basis von PDM-/ PLM-Systemen, die eine koordinierte Zusammenarbeit mehrerer Personen auf Baukastenebene (1) ermöglichen; für andere Disziplinen ist meist eine strukturierte Ablage von Dokumenten sowie eine Versions- und Freigabeverwaltung auf Zieldokumentenebene implementiert [vgl. Siemens 2012].

PDM-/PLM-Systeme unterstützen die Zusammenarbeit in der Mechanik, für andere Disziplinen bewegen sie sich meist auf der Ebene der Zieldaten. Für den firmenübergreifenden Austausch und die automatisierte Weiterverarbeitung von Daten sowie für den Einsatz in FES bieten diese Systeme bislang keine Ansätze. Des Weiteren besteht zwar die Möglichkeit der Zugangskontrolle zu bestimmten Daten, bei freigegebenen Baugruppen ist jedoch beispielsweise kein Know-How-Schutz mehr möglich.

5.5 Defizite und Handlungsbedarf

5.5.1 Qualitätssicherung bei der Erstellung von Baukästen

Keiner der aufgeführten wissenschaftlichen Ansätze bzw. industriellen Umsetzungen erfüllt die Anforderungen, die bei der Erstellung mechatronischer Baukästen vorliegen, vollständig. Die Defizite der einzelnen Ansätze hinsichtlich der Anforderungen aus Kapitel 4 sind in Tabelle 5.1 zusammengefasst dargestellt. Dabei wird deutlich, dass die beiden Konzepte der Metriken und der Entwurfsmuster im Bereich der Softwaretechnik beinahe alle Anforderungen erfüllen, jedoch auf Grund der völlig unterschiedlichen Begriffswelt nicht eins zu eins auf die Mechatronik angewendet werden können.

5.5.2 Wiederverwendbarkeit von Baukästen

In den einzelnen Disziplinen existieren bereits zahlreiche Baukästen und Bibliotheken, im Bereich der Mechatronik gibt es jedoch bislang keine Lösungen, die die Einbindung standardisierter Baukästen innerhalb von FES ermöglichen.

5.5.3 Unterstützung von Kooperationen zwischen Firmen

Für keine der vorgestellten Kooperationsebenen existieren Konzepte, die eine Unterstützung von Kooperationen unter Verwendung von FES dahingehend gewährleisten, dass eine akzeptabel effiziente und komfortable Zusammenarbeit unter Wahrung des Know-How-Schutzes ermöglicht wird.

Anforderungen		Stand der Technik und der Forschung																							
		Datenformate und Systeme für den Austausch von Informationen						Verfügbare Basisbaukästen			Methoden für die Erstellung von Baukästen oder Bibliotheken		Analyse von Baukästen												
Qualitätssicherung bei der Erstellung von Baukästen	Analyse von Baukästen	○	○	○	○	○	○	○	○	●	◐	○	Zusammenarbeit mit Hilfe von PDM-/PLM-Systemen	○	○	○	○	○	○	○	●	◐	○		
	Reduktion der Komplexität von Baukästen	○	○	○	○	○	○	○	○	○	○	○		Native Daten (Bsp. Autodesk-Inventor, P8, Step7 usw.)	○	○	○	○	○	◐	○	●	○	○	
	Standardisierte Erstellung von Baukästen	○	○	○	○	○	○	○	○	○	○	○		Mechatronische, herstellerübergreifende Austauschformate (Bsp. CAEX, AutomationML)	○	○	○	○	○	◐	○	●	○	●	
	Änderbarkeit von Baukästen	○	○	○	○	○	○	○	○	○	○	○		Disziplinspezifische, herstellerübergreifende Austauschformate (Bsp. Step, PLC-Open, IEC61131 usw.)	○	○	○	○	○	○	○	◐	●	○	○
	Verfügbarkeit / Firmenübergreifender Basisbaukästen	○	○	○	○	○	○	○	○	○	○	○		Konfigurationsdaten in Form von Tabellen (Bsp. Word, Excel, CSV, ProList)	○	○	○	○	○	○	○	○	○	○	○
	Automatische Weiterverarbeitung der Übergabedaten	○	○	○	○	○	○	○	○	○	○	○		Handbücher mit Konstruktions-/Entwicklungsrichtlinien	○	○	○	○	○	○	○	○	○	○	○
Unterstützung von Kooperationen zwischen Firmen	Verfügbare Basisbaukästen	○	○	○	○	○	○	○	○	○	○	Softwarebibliotheken (Bsp. PLC-Open-Bibliotheken, Eclipse Pulgins)	○	○	○	○	○	○	○	○	○	○	○		
	Unterstützung von Know-How-Schutz / Sicherheit der Daten	○	○	○	○	○	○	○	○	○	○		Norm-Symbolbibliotheken, frei verfügbare Schaltplanmakros (Bsp. Eplan Electric P8)	○	○	○	○	○	○	○	○	○	○		
	Räumlich getrenntes, zeitversetztes Arbeiten	○	○	○	○	○	○	○	○	○	○		Mechanik-Normteillbibliotheken in offenen Datenformaten (Bsp. STEP, IGES)	○	○	○	○	○	○	○	○	○	○		
	Paralleles Arbeiten im selben Projekt (automatische Updates)	○	○	○	○	○	○	○	○	○	○		Methoden für die Erstellung von Baukästen oder Bibliotheken	○	○	○	○	○	○	○	○	○	○	○	
Interdisziplinärer Ansatz	○	○	○	○	○	○	○	○	○	○	Funktionales, mechatronisches Engineering (Föderal)	○		○	○	○	○	○	○	○	○	○			
Mechatronik	Variantenmanagement (Schuh)	○	○	○	○	○	○	○	○	○	○	Variantenmanagement (Schuh)		○	○	○	○	○	○	○	○	○	○		
	Entwurfsmuster (Gamma)	○	○	○	○	○	○	○	○	○	○	Entwurfsmuster (Gamma)		○	○	○	○	○	○	○	○	○	○		
Legende:	Baugruppen, Referenzen, Parametrierung (Bsp. Autodesk Inventor)	○	○	○	○	○	○	○	○	○	○	Software-Metriken	○	○	○	○	○	○	○	○	○	○	○		
	Software-Metriken	○	○	○	○	○	○	○	○	○	○		●	○	○	○	○	○	○	○	○	○			

● Anforderung vollständig erfüllt; ◐ Anforderung teilweise erfüllt; ○ Anforderung nicht erfüllt

Tabelle 5.1: Übersicht über die Anforderungen und den Stand der Technik

5.5.4 Handlungsbedarf

Im Bereich Funktionaler Engineering Systeme (FES) existieren zum Zeitpunkt der Betrachtung weder für die Bewertung und Erstellung von Baukästen, noch für firmenübergreifende Basisbaukästen oder die Unterstützung von Kooperationen zwischen Firmen Konzepte, die alle gestellten Anforderungen erfüllen. Solche Konzepte werden jedoch benötigt, um die Vorteile von FES bei geringen Kosten und auch in firmenübergreifenden Engineering-Prozessen nutzen zu können.

Daher werden im Folgenden neue Lösungsansätze vorgestellt, mit Hilfe derer die gestellten Anforderungen erfüllt werden können.

6 Lösungsansatz

In der Einleitung wurde ausführlich dargelegt, dass bei vielen Unternehmen des Maschinen- und Anlagenbaus die Optimierung von Engineering-Prozessen und des Auftragsdurchlaufs stark im Fokus steht, um im Hinblick auf Qualität, Zeit und Kosten wettbewerbsfähig zu bleiben. Aus diesem Grund werden häufig FES eingesetzt, die die Konfiguration von Maschinen und Anlagen sowie die anschließende Generierung von Daten und Dokumenten ermöglichen.

Beim Einsatz von FES sehen sich die Unternehmen jedoch Stand heute oftmals den in Kapitel 3 beschriebenen Problemen gegenüber gestellt. Zum einen sind dies Qualitätsunterschiede der erstellten Baukästen, was sich auf den Grad der Wiederverwendung von Baukastenelementen, die Performanz und die Änderbarkeit der Baukästen auswirkt. Um die Erstellung von Baukästen zu standardisieren und eine reproduzierbare Qualität der Baukästen zu gewährleisten, soll das aus dem Bereich der Softwareentwicklung bekannte Konzept der Entwurfsmuster (siehe Kapitel 2.3) auf die Mechatronik übertragen werden. Die Entwurfsmuster sollen allgemein gehaltene Lösungsansätze darstellen, die wiederkehrende Problemstellungen bei der Entwicklung von Baukästen und den darin enthaltenen Baukastenelementen lösen.

Voraussetzung für die Qualitätssicherung und -Kontrolle von Baukästen ist jedoch, dass geeignete Metriken zur Analyse von Baukästen hinsichtlich Grad der Wiederverwendung von Baukastenelementen, Performanz und Änderbarkeit zur Verfügung stehen. Da in heutigen FES solche Analysewerkzeuge größtenteils fehlen, sollen geeignete Metriken gefunden und beispielhaft in einem FES als Analyseroutine implementiert werden.

Die Entwicklung der FES-Baukästen ist insbesondere deshalb zeit- und kostenaufwendig, da bei jedem Unternehmen wieder mit einem leeren Baukasten gestartet wird. Während für andere Werkzeuge bereits firmenübergreifende Basisbibliotheken existieren (siehe Tabelle 5.1), fehlen im Bereich der FES bislang Konzepte, wie solche Basisbaukästen in firmenspezifische Baukästen eingebunden werden können. Die oben genannten Entwurfsmuster bilden die Grundlage für solche Konzepte und sollen daher dazu verwendet werden, Strukturen für das Zusammenspiel firmenspezifischer und firmenübergreifender Baukästen abzubilden.

Während bei firmenübergreifenden Basisbaukästen beispielsweise der Steuerungshersteller einen Baukasten liefert, den verschiedene Unternehmen des Maschinen- und Anlagenbaus in ihren Baukästen einbinden, gibt es bei Kooperationen zwischen Unternehmen ganz andere Anforderungen. Soll im Bereich des Engineerings und der Auftragsbearbeitung mit Hilfe von FES komfortabel zusammengearbeitet werden, so müssen die Baukästen der kooperierenden Unternehmen aufeinander abgestimmt sein und Schnittstellen für den Datenaustausch bieten, wobei gleichzeitig beispielsweise der Know-How-Schutz gewährleistet sein muss. Auch hierfür sollen mit Hilfe der oben genannten Entwurfsmuster passende Konzepte entwickelt werden.

Für die Umsetzung der genannten Lösungsansätze wurde im Rahmen dieser Arbeit folgende Vorgehensweise gewählt, an Hand derer die weiteren Kapitel gegliedert sind:

6.1 Analyse vorhandener Baukästen

Um die Qualität eines Baukastens z.B. in Hinblick auf Performanz, Änderbarkeit und Grad der Wiederverwendung bewerten zu können, ist es zunächst notwendig, diesen zu analysieren. Hierfür wird in Kapitel 7 ein Software-Objekt entwickelt, das mit Hilfe zu definierender Metriken einerseits Ursachen für Probleme wie z.B. schlechte Performanz eines Modells oder hohe Änderungsaufwände analysiert und

gleichzeitig auch konkrete Hinweise auf Verbesserungsmöglichkeiten beispielweise durch den Einsatz passender Entwurfsmuster ausgibt.

6.2 Übertragung und Entwicklung mechatronischer Entwurfsmuster

Nachdem vorhandene Baukästen analysiert und bewertet werden können, kann mit der Entwicklung von Entwurfsmustern begonnen werden. Somit kann parallel zur Entwicklung gleichzeitig der Nutzen eines Entwurfsmusters quantitativ beurteilt werden. Wie in Tabelle 5.1 zu sehen, erfüllt das Konzept der Entwurfsmuster aus der Softwaretechnik die meisten Anforderungen in Hinblick auf Reduktion der Komplexität, standardisierte Erstellung und Änderbarkeit von Baukästen. Deshalb wird es in Kapitel 8 auf die Mechatronik, genauer gesagt auf den Maschinen- und Anlagenbau übertragen. Neben begrifflichen Anpassungen bedarf es der grundlegenden Überarbeitung einiger Entwurfsmuster. Für einige Problemstellungen der Mechatronik gibt es keine Lösungen in der Softwaretechnik; für diese Problemstellungen müssen neue Entwurfsmuster entwickelt werden.

6.3 Entwicklung von Konzepten für firmenübergreifende Basisbaukästen

Um neben reproduzierbarer Qualität auch eine Senkung der Aufwände bei der Erstellung neuer Baukästen zu erreichen, soll die Einbindung firmenübergreifend wiederverwendbarer Basisbaukästen ermöglicht werden. Dafür werden Konzepte benötigt, die in Kapitel 9 mit Hilfe der entwickelten Entwurfsmuster realisiert werden. Diese Konzepte sollen gewährleisten, dass die Unternehmen weiterhin Daten und Unterlagen flexibel und nach Kunden- oder Firmenstandard generieren können. Je nach Anwendungsfall sind hierbei die Anforderungen unterschiedlich, je nachdem, wie ausgeprägt Firmenstandards bei der Gestaltung von Daten und Unterlagen (z.B. Vorgabe von Funktionsbausteinen oder Schaltplanmakros) sind. Aus diesem Grund sollen Konzepte für unterschiedliche Szenarien bezüglich des Umfangs firmenübergreifender Basisbaukästen entwickelt werden.

6.4 Entwicklung von Konzepten für die Kooperation zwischen Firmen

Analog zu firmenübergreifenden Basisbaukästen sollen auch Konzepte für Kooperationen zwischen Firmen entwickelt werden, die eine möglichst komfortable Zusammenarbeit unter Verwendung von FES ermöglichen. Hierzu werden in Kapitel 10 Strukturen unter Benutzung der oben genannten Entwurfsmuster entwickelt, innerhalb derer Baukästen unterschiedlicher Firmen über definierte Schnittstellen zusammenarbeiten können. Da in der Praxis drei verschiedene Kooperationsebenen vorkommen (siehe Kapitel 3.3), für die ganzheitliche und effiziente Engineering-Prozesse gestaltet werden sollen, werden unterschiedliche Lösungskonzepte passend zu diesen drei Kooperationsebenen benötigt. Diese Lösungskonzepte sollen die jeweiligen Anforderungen wie automatische Weiterverarbeitung der Übergabedaten, Know-How-Schutz, zeitversetztes oder paralleles Arbeiten erfüllen.

7 Analyse vorhandener Baukästen

Um Aussagen zur Bewertung von Performanz, Änderbarkeit und Grad der Wiederverwendung von Baukästen zu machen, kann das Konzept der Metriken aus dem Bereich der Softwaretechnik auf FES übertragen werden. Mit Hilfe von Metriken können dann einerseits Ursachen für Probleme wie z.B. schlechte Performanz eines Modells oder hohe Änderungsaufwände herausgefunden und gleichzeitig auch konkrete Hinweise auf Verbesserungsmöglichkeiten beispielweise durch den Einsatz von Entwurfsmustern gewonnen werden. Gleichzeitig ist es dadurch möglich, die Auswirkungen von Entwurfsmustern zu visualisieren und Optimierungsschritte innerhalb von Baukästen zu motivieren bzw. nachzuprüfen.

Im Folgenden werden zur Beurteilung von FES-Baukästen geeignete Metriken und zugehörige Grenzwerte vorgeschlagen, kategorisiert und genauer erläutert. Zunächst erfolgt jedoch eine Beschreibung der gewählten Berechnungsmethoden innerhalb der Metriken.

7.1 Berechnungsmethoden

Die im Rahmen dieser Arbeit zur Analyse vorhandener Baukästen eingesetzten Metriken werden auf Projekte oder Baukästen eines FES angewendet. Eine Metrik enthält meist einen Algorithmus, der beispielsweise durch alle Elemente eines Projektbaums läuft und für die jeweils in Betracht zu ziehenden Elemente bestimmte Eigenschaften auswertet. Die dabei entstehenden Ergebnislisten werden mit Hilfe von Berechnungsmethoden zu einer Kennzahl ausgewertet. Die Auswertungen innerhalb

der hier verwendeten Metriken basieren hauptsächlich auf zwei Berechnungsmethoden, dem Maximum und dem Median. Diese beiden Berechnungsmethoden werden im Folgenden genauer erläutert.

7.1.1 Maximum

Als Maximum bezeichnet man den größten Wert bzw. das größte Element einer geordneten Menge.

7.1.2 Median

Um die mittlere Häufigkeit eines Ereignisses oder Zustandes zu errechnen, kommt innerhalb der folgenden Analyse der Median zum Einsatz. Der Median einer Liste von Werten ist die Zahl, welche an der mittleren Stelle steht, wenn man die Werte nach Größe sortiert. Beispielsweise ist der Median für die Werte 5, 0, 36, 2, 0 die Zahl 2, nämlich die mittlere Zahl in 0, 0, 2, 5, 36. Enthält die Liste eine gerade Anzahl von Werten, so wird das arithmetische Mittel der beiden mittleren Zahlen berechnet. Dies ist z.B. bei der Liste 0, 2, 3, 6, 10, 50 der Fall, der Median berechnet sich hier zu 4,5.

Allgemein teilt ein Median eine Stichprobe, eine Anzahl von Werten oder eine Verteilung in zwei Hälften, so dass die Werte in der einen Hälfte kleiner als der Medianwert sind, in der anderen größer. Der Median gehört zur Gruppe der Quantile und kann auch als 0,5-Quantil bezeichnet werden.

Der Median findet für die im Folgenden beschriebene Analyse Anwendung, da er im Vergleich zum arithmetischen Mittel, auch Durchschnitt genannt, robuster gegenüber Ausreißern (extrem abweichenden Werten) ist [vgl. Georgii 2009].

7.2 Kategorisierung der Metriken

Die vorgeschlagenen und im Folgenden genauer erläuterten Metriken lassen sich in drei Kategorien einteilen, die sich aus den Anforderungen in Kapitel 4.1 ergeben und das Bewertungsziel der jeweiligen Metrik ausdrücken. Dadurch ist es möglich, die Metriken in einer Übersicht tabellarisch darzustellen und je nach Anwendungsfall an Hand der Kategorien schnell die passende Metrik zu finden. Diese drei Kategorien sind im Einzelnen:

- **Performanz:** die Performanz ist ein Maß dafür, wie schnell Modelle im FES berechnet bzw. Daten und Dokumente generiert werden. Eine gute Performanz bringt außerdem Vorteile bei Reaktionszeiten von Eingabemasken, Dateigrößen von Austauschdaten und Übermittlungszeiten bei webbasierten Anwendungen.
- **Übersichtlichkeit:** die Übersichtlichkeit trägt zusammen mit dem Grad der Wiederverwendung stark zur Änder- und Wartbarkeit von FES-Baukästen bei. Ist die Übersichtlichkeit nicht gegeben, steigt der Aufwand zur Einarbeitung eines neuen Baukastenentwicklers erheblich an.
- **Grad der Wiederverwendung:** ist der Grad der Wiederverwendung von Baukastenelementen innerhalb FES-Baukästen hoch, sinkt der Änderungs- und Wartungsaufwand, da Regelwissen und Dokumentfragmente konzentriert an einigen wenigen Stellen gepflegt werden können.

Tabelle 7.1 zeigt eine Übersicht der im Folgenden zur Analyse von FES-Baukästen eingesetzten Metriken. Diese sind zum einen geordnet an Hand des Anwendungsbereichs:

- **Mechatronik:** die mechatronische Struktur eines Projekts
- **Disziplinen:** die disziplinspezifische(n) Struktur(en) eines Projekts
- **Projekte:** alle innerhalb des FES geladenen Projekte

7.2 Kategorisierung der Metriken

Zum anderen wird an Hand der oben beschriebenen Kategorien das Bewertungsziel der Metrik dargestellt (Performanz/Übersichtlichkeit/Grad der Wiederverwendung). Einige der Metriken verfolgen jedoch mehr als eines der Bewertungsziele, weshalb sie in mehreren Kategorien genannt werden.

		Metrik	Kategorie		
			Performanz	Übersichtlichkeit	Grad der Wiederverwendung
Anwendungs- bereich	Mechatronik	Projektgröße	●	◐	○
		Projekttiefe	●	◐	○
		Feingranularität	○	●	○
		Verästelung	○	●	○
		Schlankheit	●	◐	○
		Wiederverwendung	○	○	●
		Ballastfreiheit	●	◐	○
		Dynamik	◐	●	○
		Parameter- verwendung	○	●	●
	Disziplinen	Anzahl Disziplinen	●	◐	○
		Anzahl Objekte	●	◐	○
		Projekttiefe	●	◐	○
		Feingranularität	●	◐	○
		Wiederverwendung	○	◐	●
	Projekte	Anzahl Projekte	●	◐	○

● Bewertungsziel vollständig erfüllt, ◐ teilweise erfüllt, ○ nicht erfüllt

Tabelle 7.1: Kategorisierung der Metriken

7.3 Metriken

Um Aussagen im Hinblick auf Performanz, Übersichtlichkeit und Grad der Wiederverwendung von Baukästen treffen zu können, wurden mehrere Metriken bestimmt, die innerhalb der mechatronischen und der disziplinspezifischen Strukturen sowie im Überblick der Projekte berechnet werden.

Zur Veranschaulichung der im Folgenden aufgeführten Metriken zeigt Bild 7.1 ein Beispiel für eine Projektübersicht innerhalb eines FES.

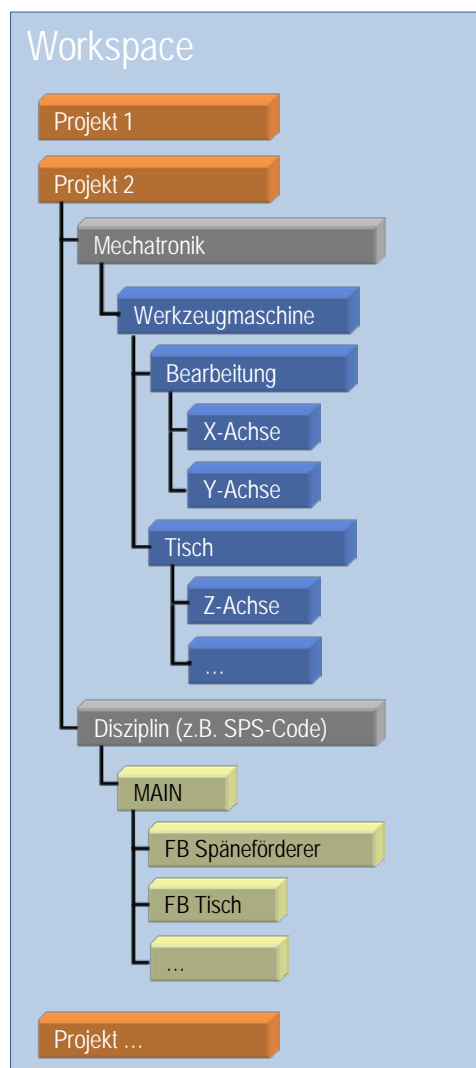


Bild 7.1: Beispiel einer Projektübersicht innerhalb eines FES

Je nach Konfiguration des FES können ein oder mehrere Projekte gleichzeitig im Speicher (Workspace) vorgehalten werden. Ein Projekt gliedert sich auf in eine mechatronische und eine Disziplin-Struktur, worunter die jeweiligen mechatronischen und disziplinspezifischen Projektbäume aufgehängt sind.

Einige Metriken zielen u.a. auf die Performanz eines Modells. Ob es notwendig ist, ein Modell diesbezüglich zu optimieren, hängt von der Größe der Projekte (z.B. große Anlagen) sowie von der bereitgestellten Rechenleistung (z.B. Anzahl Prozessoren, Arbeitsspeicher) ab. Sinnvoll ist ein schlankes und performantes Modell jedoch immer, da es neben kurzen Reaktionszeiten und geringem Speicherbedarf auch zu einer besseren Übersichtlichkeit und Wartbarkeit beiträgt.

Die vorgeschlagenen Metriken werden im Folgenden hergeleitet und genauer erläutert. Die Gliederung erfolgt nach Anwendungsbereich (Mechatronik/Disziplinen/Projekte).

7.3.1 Mechatronik

Zur Beurteilung der **Performanz** eines mechatronischen Modells kommen vor allem Metriken zum Einsatz, die die Anzahl von Komponenten und die Struktur eines Projektes bewerten:

- **Projektgröße:** Sie stellt die Anzahl aller in einem Projekt verwendeten mechatronischen und disziplinspezifischen Baukastenelemente dar. Werden die Projektgrenzen zu groß gewählt, kann dies zu Performanz- und Übersichtlichkeitseinbußen führen.
- **Projekttiefe (Maximum/Median):** Sie misst das Maximum bzw. den Median der Tiefe des mechatronischen Projektbaums. Ist die Projekttiefe zu groß oder zu klein, beeinträchtigt dies sowohl Performanz als auch Übersichtlichkeit des Modells.
- **Schlankheit (Maximum/Median):** Im Fokus dieser Metrik steht das Maximum bzw. der Median der Anzahl von **fest** in Kompositionen eingebauten Unterkomponenten (fest oder per Disabler). Die Betrachtung erfolgt ohne

Blätter (Komponenten ohne Unterkomponenten). Die Schlankheit ist eine Kenngröße für die Übersichtlichkeit und Wartungsfreundlichkeit von Modellen, da Disabler zu mehr Objekten und redundantem Regelwissen führen können. Die Existenz vieler Objekte beeinflusst wiederum die Performanz des Modells.

- **Ballastfreiheit:** Sie misst das Verhältnis aktiver (nicht disableter) Komponenten zur Anzahl aller Komponenten. Ist ein hoher Anteil an Komponenten nicht aktiv, deutet dies darauf hin, dass an Stelle von Disablern besser Erweiterungspunkte eingesetzt werden sollten, um dadurch die Übersichtlichkeit und Änderungsfreundlichkeit sowie die Performanz der Modelle zu erhöhen.

Zur Bestimmung des **Grad der Wiederverwendung** innerhalb mechatronischer Modelle können folgende Metriken verwendet werden, von denen eine auf Baukastenelemente, die andere auf Parameter zielt:

- **Wiederverwendung (Maximum/Median):** Diese Metrik misst das Maximum bzw. den Median der Anzahl der Instanzen eines mechatronischen Baukastenelements. Ein hoher Grad der Wiederverwendung trägt zur Wartungsfreundlichkeit und Übersichtlichkeit eines Baukastens bei.
- **Parameterverwendung (Absolut/Maximum/Median):** Diese Metrik misst die Anzahl der im Modell verwendeten Parameter sowie das Maximum und den Median der Anzahl der Verwendungen eines Parameters. Je geringer die Anzahl unterschiedlicher Parameter und je höher die Wiederverwendung eines Parameters, desto besser sind Übersichtlichkeit und somit Änderbarkeit des Baukastens.

Alle oben beschriebenen Metriken lassen neben Performanz und Grad der Wiederverwendung auch eine Bewertung der Übersichtlichkeit mechatronischer Modelle zu. Darüber hinaus gibt es jedoch noch folgende weitere Metriken, die explizit zur Beurteilung von Übersichtlichkeit und damit auch Änderbarkeit entworfen wurden:

- **Feingranularität:** Sie stellt den prozentualen Anteil von Blättern (Komponenten ohne Unterkomponenten/Einfügeplätze) bezogen auf die Projektgröße dar. Für die Übersichtlichkeit eines Modells ist es wichtig, dass der Projektbaum ausgeglichen ist, d.h. er ist weder zu tief noch zu flach.

- **Verästelung (Maximum/Median):** Sie misst das Maximum bzw. den Median der Anzahl von in Kompositionen eingebauten Komponenten (fest oder per Disabler/Erweiterungspunkt/manuellem Einfügeplatz). Die Betrachtung erfolgt für alle Komponenten, die Unterkomponenten besitzen, also keine Blätter sind. Das Ergebnis dieser Metrik ist ebenfalls ein Zeichen dafür, ob ein Projektbaum ausgeglichen und somit übersichtlich ist.
- **Dynamik (Absolut/Maximum/Median):** Diese Metrik ermittelt die Anzahl der im Projekt enthaltenen Erweiterungspunkte sowie das Maximum bzw. den Median der Anzahl der auf einen Erweiterungspunkt eingefügten Komponenten. Für eine gute Übersichtlichkeit sollte auf jeden Erweiterungspunkt mindestens eine Komponente instanziiert sein, da ansonsten der Disabler besser geeignet wäre.

7.3.2 Disziplinen

Zur Beurteilung der **Performanz** eines disziplinspezifischen Projektbaums kommen vor allem Metriken zum Einsatz, die die Anzahl von Objekten und die Struktur eines Disziplinbaums bewerten:

- **Anzahl der generierten Disziplinen:** Diese Metrik zählt, wie viele Disziplinen generiert wurden und sich im Speicher befinden. Für eine gute Performanz und Übersichtlichkeit des Modells sollte immer nur die Disziplinstruktur geladen werden, die momentan benötigt wird.
- **Anzahl der Objekte im Disziplinbaum:** Es wird gemessen, wie viele Objekte jede Disziplinstruktur enthält. Werden die Projektgrenzen zu groß gewählt, kann dies auch zu Performanz- und Übersichtlichkeitseinbußen in den Disziplinstrukturen führen.
- **Tiefe des Disziplinbaums (Maximum/Median):** Diese Metrik bestimmt das Maximum bzw. den Median der Tiefe jedes Disziplinbaums. Ist die Tiefe eines Disziplinbaums zu groß oder zu klein, beeinträchtigt dies sowohl Performanz als auch Übersichtlichkeit des Modells.
- **Feingranularität (Maximum/Median):** Sie sagt aus, wie groß das Maximum bzw. der Median der Anzahl von Parametern in den Komponenten der

jeweiligen Disziplinstruktur ist. Die Anzahl der Parameter in einem Disziplinobjekt lässt Rückschlüsse auf die Größe des Fragments des Disziplinobjekts zu. Für eine gute Übersichtlichkeit und Performanz des Modells sollten die Objekte nicht zu klein, aber auch nicht zu groß gewählt werden.

Zur Bestimmung des **Grad der Wiederverwendung** innerhalb eines Disziplinaums kann folgende Metrik verwendet werden:

- **Wiederverwendung (Maximum/Median):** Sie gibt das Maximum bzw. den Median der Anzahl der Instanzen eines Baukastenelements in der jeweiligen Disziplinstruktur an. Ein hoher Grad der Wiederverwendung trägt zur Wartungsfreundlichkeit und Übersichtlichkeit eines Baukastens bei.

Die oben genannten Metriken lassen gleichzeitig auch eine Bewertung der **Übersichtlichkeit** disziplinspezifischer Projektbäume zu.

7.3.3 Projekte

Für **Performanz** und **Übersichtlichkeit** innerhalb eines FES ist noch eine weitere Metrik wichtig, die sich auf die Gesamtheit aller geladenen Projekte bezieht:

- **Anzahl der Projekte:** Diese Metrik zählt, wie viele Projekte sich im Speicher befinden. Für eine gute Performanz und Übersichtlichkeit des Modells sollte immer nur das Projekt geladen werden, das momentan bearbeitet wird.

7.4 Grenzwerte

Die jeweiligen Grenzwerte der einzelnen Metriken verstehen sich als Empfehlungen, die auf Erfahrungswerten aus unterschiedlichen Kundenprojekten basieren. Diese Erfahrungswerte entstanden bei der Analyse von FES-Baukästen im System Eplan Engineering Center in Unternehmen des Maschinen-/Anlagenbaus. Die Darstellung der Ergebnisse mit Hilfe der Ampelfarben (grün – gelb – rot) soll keine Bewertung des Baukastenentwicklers darstellen, sondern dabei unterstützen, Hinweise auf Ursachen unterschiedlicher Probleme und geeignete Optimierungsmöglichkeiten (z.B. Entwurfsmuster) zu geben.

7.4 Grenzwerte

Die zu jeder Metrik genannten Grenzwerte stellen Erfahrungswerte auf Grundlage heute gebräuchlicher FES und Rechnersysteme dar.

Die Farben bedeuten hierbei:

Grün	Optimaler Bereich
Gelb	Leichte Einschränkungen bei Performanz und Übersichtlichkeit, mittlerer Grad der Wiederverwendung
Rot	Kritischer Zustand des Modells, deutliche Einbußen bei Performanz und Übersichtlichkeit, schlechter Grad der Wiederverwendung

Tabelle 7.2: Bedeutung der Farben für die Grenzwerte der Metriken

Bei Metriken, die die Anzahl bestimmter Objekte oder ein Verhältnis der Anzahl unterschiedlicher Objekte innerhalb des Modells angeben, wird das Ergebnis als Absolutwert angezeigt. Diejenigen Metriken, die statistische Auswertungen auf mehreren Objekten vornehmen, werden mit Maximum und Median dargestellt. Das Maximum dient dazu, um eventuelle Ausreißer zu bemerken, während der Median eine Übersicht über die mittlere Anzahl der jeweiligen Vorkommen liefert.

Die ermittelten Grenzwerte zu den einzelnen Metriken sind in Tabelle 7.3 (Absolutwert), Tabelle 7.4 (Maximum) und Tabelle 7.5 (Median) dargestellt.

7 Analyse vorhandener Baukästen

Metrik / Grenzwert	Legende/Einheit	Absolutwert		
		grün	gelb	rot
Mechatronik				
Projektgröße	Anzahl Komponenten	< 10.000	10.000 .. 100.000	> 100.000
Projekttiefe	Anzahl Komponenten vom Kind bis zur Wurzel	-	-	-
Feingranularität	prozentualer Anteil von Blättern bezogen auf Projektgröße	40 .. 90 %	10 .. 40 %	< 10 oder > 90 %
Verästelung	Anzahl von in Kompositionen eingebauten Komponenten	-	-	-
Schlankheit	Anzahl von in Kompositionen fest eingebauten Komponenten	-	-	-
Wiederverwendung	Anzahl Instanzen eines Baukastenelements	-	-	-
Ballastfreiheit	Verhältnis aktiver Komponenten zu allen Komponenten	> 80 %	20 .. 80 %	< 20 %
Dynamik	Anzahl der im Projekt enthaltenen Erweiterungspunkte	> 10	1 .. 10	0
Parameterverwendung	Anzahl der im Modell verwendeten Parameter	< 250	250 .. 1000	> 1000
Disziplinen		grün	gelb	rot
Anzahl der generierten Disziplinen	Anzahl Disziplinen	0 .. 1	> 1	> 2
Anzahl der Objekte im Disziplinbaum	Anzahl Objekte	< 10.000	10.000 .. 100.000	> 100.000
Tiefe des Disziplinbaums	Anzahl Objekte vom Kind bis zur Wurzel	-	-	-
Feingranularität	Anzahl Parameter in einem Objekt	-	-	-
Wiederverwendung	Anzahl Instanzen eines Baukastenelements	-	-	-
Projekte		grün	gelb	rot
Anzahl der Projekte	Anzahl Projekte	0 .. 1	> 1	> 2

Tabelle 7.3: Grenzwerte der Metriken, Teil 1: Absolutwert

7.4 Grenzwerte

Metrik / Grenzwert	Legende/Einheit	Maximum		
		grün	gelb	rot
Mechatronik				
Projektgröße	Anzahl Komponenten	-	-	-
Projekttiefe	Anzahl Komponenten vom Kind bis zur Wurzel	4 .. 20	21 .. 50	< 4 oder > 50
Feingranularität	prozentualer Anteil von Blättern bezogen auf Projektgröße	-	-	-
Verästelung	Anzahl von in Kompositionen eingebauten Komponenten	< 20	20 .. 50	> 50
Schlankheit	Anzahl von in Kompositionen fest eingebauten Komponenten	< 10	10 .. 20	> 20
Wiederverwendung	Anzahl Instanzen eines Baukastenelements	> 20	2 .. 20	< 2
Ballastfreiheit	Verhältnis aktiver Komponenten zu allen Komponenten	-	-	-
Dynamik	Anzahl der auf einen Erw.-punkt eingefügten Komponenten	> 10	1 .. 10	0
Parameterverwendung	Anzahl der Verwendungen eines Parameters	> 50	5 .. 50	< 5
Disziplinen				
Anzahl der generierten Disziplinen	Anzahl Disziplinen	-	-	-
Anzahl der Objekte im Disziplinbaum	Anzahl Objekte	-	-	-
Tiefe des Disziplinbaums	Anzahl Objekte vom Kind bis zur Wurzel	4 .. 20	21 .. 50	< 4 oder > 50
Feingranularität	Anzahl Parameter in einem Objekt	5 .. 20	21 .. 50	< 5 oder > 50
Wiederverwendung	Anzahl Instanzen eines Baukastenelements	> 20	2 .. 20	< 2
Projekte				
Anzahl der Projekte	Anzahl Projekte	-	-	-

Tabelle 7.4: Grenzwerte der Metriken, Teil 2: Maximum

Metrik / Grenzwert	Legende/Einheit	Median		
		grün	gelb	rot
Mechatronik				
Projektgröße	Anzahl Komponenten	-	-	-
Projekttiefe	Anzahl Komponenten vom Kind bis zur Wurzel	4,0 .. 10,0	10,0 .. 20,0	< 4,0 oder > 20,0
Feingranularität	prozentualer Anteil von Blättern bezogen auf Projektgröße	-	-	-
Verästelung	Anzahl von in Kompositionen eingebauten Komponenten	< 10,0	10,0 .. 20,0	> 20,0
Schlankheit	Anzahl von in Kompositionen fest eingebauten Komponenten	< 5,0	5,0 .. 10,0	> 10,0
Wiederverwendung	Anzahl Instanzen eines Baukastenelements	> 5,0	1,5 .. 5,0	< 1,5
Ballastfreiheit	Verhältnis aktiver Komponenten zu allen Komponenten	-	-	-
Dynamik	Anzahl der auf einen Erw.-punkt eingefügten Komponenten	> 2,0	0,5 .. 2,0	< 0,5
Parameterverwendung	Anzahl der Verwendungen eines Parameters	> 10,0	2,0 .. 10,0	< 2,0
Disziplinen				
Anzahl der generierten Disziplinen	Anzahl Disziplinen	-	-	-
Anzahl der Objekte im Disziplinbaum	Anzahl Objekte	-	-	-
Tiefe des Disziplinbaums	Anzahl Objekte vom Kind bis zur Wurzel	4,0 .. 10,0	10,0 .. 20,0	< 4,0 oder > 20,0
Feingranularität	Anzahl Parameter in einem Objekt	3,0 .. 10,0	10,0 .. 20,0	< 3,0 oder > 20,0
Wiederverwendung	Anzahl Instanzen eines Baukastenelements	> 5,0	1,5 .. 5,0	< 1,5
Projekte				
Anzahl der Projekte	Anzahl Projekte	-	-	-

Tabelle 7.5: Grenzwerte der Metriken, Teil 3: Median

Die beschriebenen Metriken mit den jeweiligen Grenzwerten wurden in einem Softwareobjekt implementiert und anschließend an einem Baukasten eines Anlagenherstellers validiert, was im Folgenden genauer erläutert wird.

7.5 Realisierung

Im Rahmen der Dissertation entstand ein „Model-Analyser“. Dabei handelt es sich um ein Objekt, das in ein beliebiges Projekt im Eplan Engineering Center eingebaut wird. Zur Auswertung der Ergebnisse des Model-Analysers werden sämtliche Werte

grafisch als „Ampel-Balken“ dargestellt. Bild 7.2 zeigt beispielhaft die Auswertung, die bei der Analyse eines Baukastens eines Anlagenherstellers entstand. In diesem Fall wurde der Baukasten analysiert, da sich Projektoren über lange Wartezeiten auf Grund von zeitintensiven Modellberechnungen bei der Konfiguration einer Anlage beschwerten.

Der vorliegenden Auswertung kann entnommen werden, dass das Projekt sehr groß ist (Projektgröße), also sehr viele Komponenten enthält. Für diesen Fall ist es meist sinnvoll, so weit möglich in mehreren Stufen zu konfigurieren und die Anzahl der im Projekt enthaltenen Baukastenelemente (Detailierungstiefe) Schritt für Schritt zum spätmöglichen Zeitpunkt zu erhöhen. Dies kann über das Entwurfsmuster Erweiterungspunkt realisiert werden (siehe auch Kapitel 8.2.4), was laut Auswertung bereits teilweise gemacht wurde. Es ist jedoch ersichtlich, dass zwar viele Erweiterungspunkte eingefügt wurden (Anzahl Erweiterungspunkte im Projekt), jedoch nur wenige Erweiterungspunkte tatsächlich belegt sind (Dynamik, Komponenten auf EP Median). Des Weiteren existieren Kompositionen mit bis zu 100 fest verbauten Komponenten (Schlankheit, feste Unterkomponenten maximal).

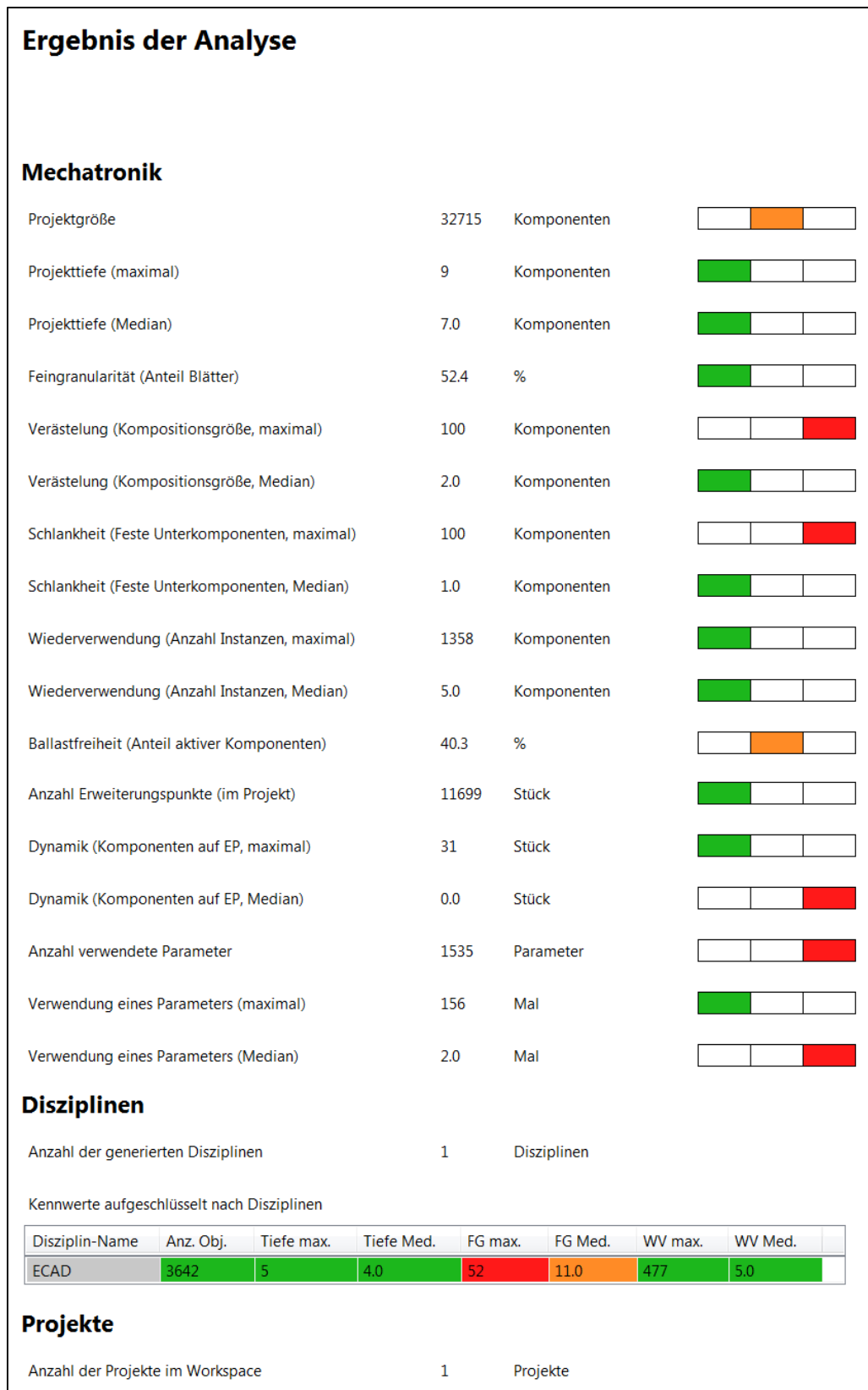


Bild 7.2: Screenshot Ergebnis der Analyse eines Modells

7.6 Fazit

Auf Grund einer Auswertung mit Hilfe des Model-Analysers kann in kurzer Zeit ein Überblick über einen vorliegenden Baukasten und ein Projekt gewonnen werden. Die ermittelten Kennwerte liefern erste Anhaltspunkte, wie Performanz, Übersichtlichkeit und Grad der Wiederverwendung eines Modells verbessert werden können. Nach diesem ersten Überblick ist es möglich, über Strukturen des Baukastens zu diskutieren und schneller die Stellen zu finden, wo Optimierungen durch den Einsatz der im folgenden Kapitel beschriebenen Entwurfsmuster durchgeführt werden können.

8 Entwurfsmuster

Um die in Kapitel 4.1 genannten Anforderungen zur Erreichung eines gleichbleibenden und hohen Qualitätsstandards bei der Erstellung von Baukästen zu erfüllen, wurden im Rahmen dieser Dissertation mechatronische Entwurfsmuster entwickelt. Diese werden im Folgenden zunächst kategorisiert und anschließend genauer erläutert.

8.1 Kategorisierung der Entwurfsmuster

Die Kategorisierung der entwickelten Entwurfsmuster dient der besseren Übersichtlichkeit. Abhängig von der vorliegenden Problemstellung kann an Hand der Kategorien schnell das passende Entwurfsmuster gefunden werden. Die Kategorisierung erfolgt einerseits [analog zu Gamma 2004] an Hand des Kriteriums Aufgabe, andererseits nach Optimierungsziel.

Die Aufgabe fasst zusammen, was das Entwurfsmuster macht. Es kann entweder der Erzeugung von Instanzen dienen (Erzeugungsmuster), sich mit der Zusammensetzung von Klassen und Objekten beschäftigen (Strukturmuster) oder eine Art und Weise beschreiben, wie Klassen und Instanzen zusammenarbeiten und Zuständigkeiten aufteilen (Verhaltensmuster).

Die Optimierungsziele Performanz, Übersichtlichkeit und Grad der Wiederverwendung entsprechen den Kategorien der Metriken (siehe Kapitel 7.2). Dadurch ist direkt ersichtlich, welche Entwurfsmuster angewandt werden können, wenn bei der Anwendung des Model-Analysers bei einer bestimmten Metrik der Wert beispielsweise im roten Bereich liegt.

8.1 Kategorisierung der Entwurfsmuster

Eine Übersicht über die entwickelten Entwurfsmuster, geordnet nach zugehöriger Aufgabe und Optimierungsziel, zeigt Tabelle 8.1.

Aufgabe	Muster	Optimierungsziel		
		Perfor- manz	Übersicht- lichkeit	Grad der Wiederver- wendung
Erzeugungsmuster	Komposition <i>Seite 63</i>	○	●	◐
	Kopiervorlage <i>Seite 65</i>	○	◐	○
	Disabler <i>Seite 67</i>	○	●	●
	Erweiterungspunkt <i>Seite 69</i>	●	◐	●
	Erweiterungspunkt mit Auswahlformel <i>Seite 71</i>	●	◐	●
	Erweiterungspunkt mit Auswahlformel und Anzahl <i>Seite 73</i>	●	◐	●
	Strukturmuster	Trennung Architektur – Baukasten <i>Seite 75</i>	○	●
4-Ebenen-Komponentenmodell <i>Seite 77</i>		○	●	●
Herstellerbibliothek <i>Seite 80</i>		◐	◐	●
Parameter-Bibliothek <i>Seite 83</i>		●	●	●
Verhaltensmuster	Schnittstellenparameter <i>Seite 85</i>	○	●	○
	Dynamische Parametervererbung <i>Seite 87</i>	●	◐	○
	Zentralsteuerung <i>Seite 89</i>	●	●	○
	Entscheidungstabelle <i>Seite 92</i>	◐	●	○
	Memento <i>Seite 95</i>	◐	●	○

● Optimierungsziel vollständig erfüllt, ◐ teilweise erfüllt, ○ nicht erfüllt

Tabelle 8.1: Übersicht über die Entwurfsmuster

Manche der Entwurfsmuster bewirken bei Anwendung eine Optimierung in mehreren Bereichen, weshalb in der Tabelle mehrere Optimierungsziele gleichzeitig genannt werden. Im Folgenden werden die Entwurfsmuster hergeleitet und jeweils genauer erläutert, wann welches Muster eingesetzt werden kann, welche Konsequenzen dies hat und in welcher Beziehung das Muster zu anderen Mustern steht.

8.2 Erzeugungsmuster

Wie der Name bereits sagt, dienen Erzeugungsmuster der Erzeugung von Instanzen. Erzeugungsmuster stellen somit unterschiedliche Möglichkeiten dar, wie eine Komponente, die in verschiedenen Varianten und mit unterschiedlichen Optionen verfügbar sein soll, im Baukasten angelegt werden kann, so dass sich später die entsprechenden Instanzen dieser Komponente im Projekt erzeugen lassen.

Voraussetzung für die Instanziierung größerer, verschachtelter Baugruppen ist das im Folgenden beschriebene Entwurfsmuster Komposition, auf dem alle anderen Erzeugungsmuster basieren.

8.2.1 Komposition

Das Entwurfsmuster Komposition beschreibt die Verschachtelung von Baukastenelementen zu größeren Einheiten, wodurch mehrere Baukastenelemente auf einmal instanziiert werden können. Dazu werden in ein Baukastenelement ein oder mehrere weitere Baukastenelemente einfach oder mehrfach eingebaut (siehe Bild 8.1). Durch das Hinzufügen einer Komposition zu einer anderen Komposition können beliebig tiefe Verschachtelungen geschaffen werden.

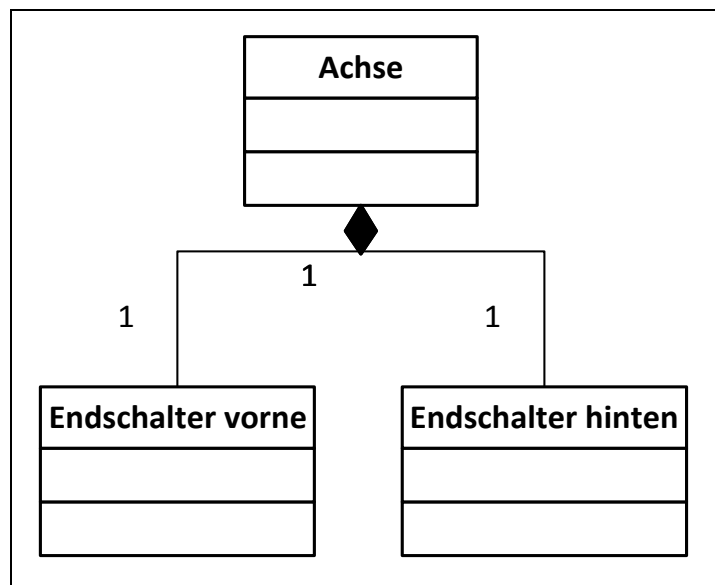


Bild 8.1: Entwurfsmuster Komposition

Durch das Bilden von Kompositionen wird die Standardisierung von Baukastenelementen vorangetrieben und die Übersichtlichkeit des Baukastens wird verbessert. Die Anwendung des Musters ist überall dort sinnvoll, wo größere Einheiten bestehend aus mehreren Baukastenelementen wiederverwendet werden sollen und alle Regeln vollständig bekannt sind.

Da es sich beim Komposition-Muster um das grundlegendste Muster handelt, findet es sehr häufig in allen Bereichen von Baukästen Anwendung. Von der Ebene der Maschine/Anlage bis hinunter zu einzelnen Komponenten kann es eingesetzt werden. Mit Hilfe dieses Musters werden für gewöhnlich die disziplinspezifischen Fragmente (z.B. Schaltplanmakro oder SPS-Baustein) in eine mechatronische Komponente eingebaut.

Durch die Verwendung des Komposition-Musters nimmt die Granularität des Gesamtmodells ab, wodurch der Aufwand bei der Konfiguration gesenkt werden kann. Die primitivste Möglichkeit, eine Komposition im Baukasten zu implementieren ist es, alle möglicherweise vorkommenden Kombinationen aus Varianten und Optionen

explizit zu modellieren. Dieses Vorgehen beschreibt das Entwurfsmuster Kopiervorlage.

Steigt die Anzahl der verschachtelten Baukastenelemente jedoch auf Grund vieler unterschiedlicher Kombinationsmöglichkeiten aus Varianten und Optionen stark an, sollte das Disabled- (siehe Kapitel 8.2.3) oder Erweiterungspunkt-Muster (siehe Kapitel 8.2.4) verwendet werden.

8.2.2 Kopiervorlage

Beim Entwurfsmuster Kopiervorlage werden alle zulässigen Kombinationen aus Varianten und Optionen einer Baugruppe einzeln in einer Bibliothek abgelegt (siehe Bild 8.2). Der Erstellung der Baugruppen liegt dabei das Muster Komposition zu Grunde, das in Kapitel 8.2.1 genauer erläutert wird. Beim Projektieren werden die erstellten Baugruppen dann als eine Art Kopiervorlage instanziiert.

8.2 Erzeugungsmuster

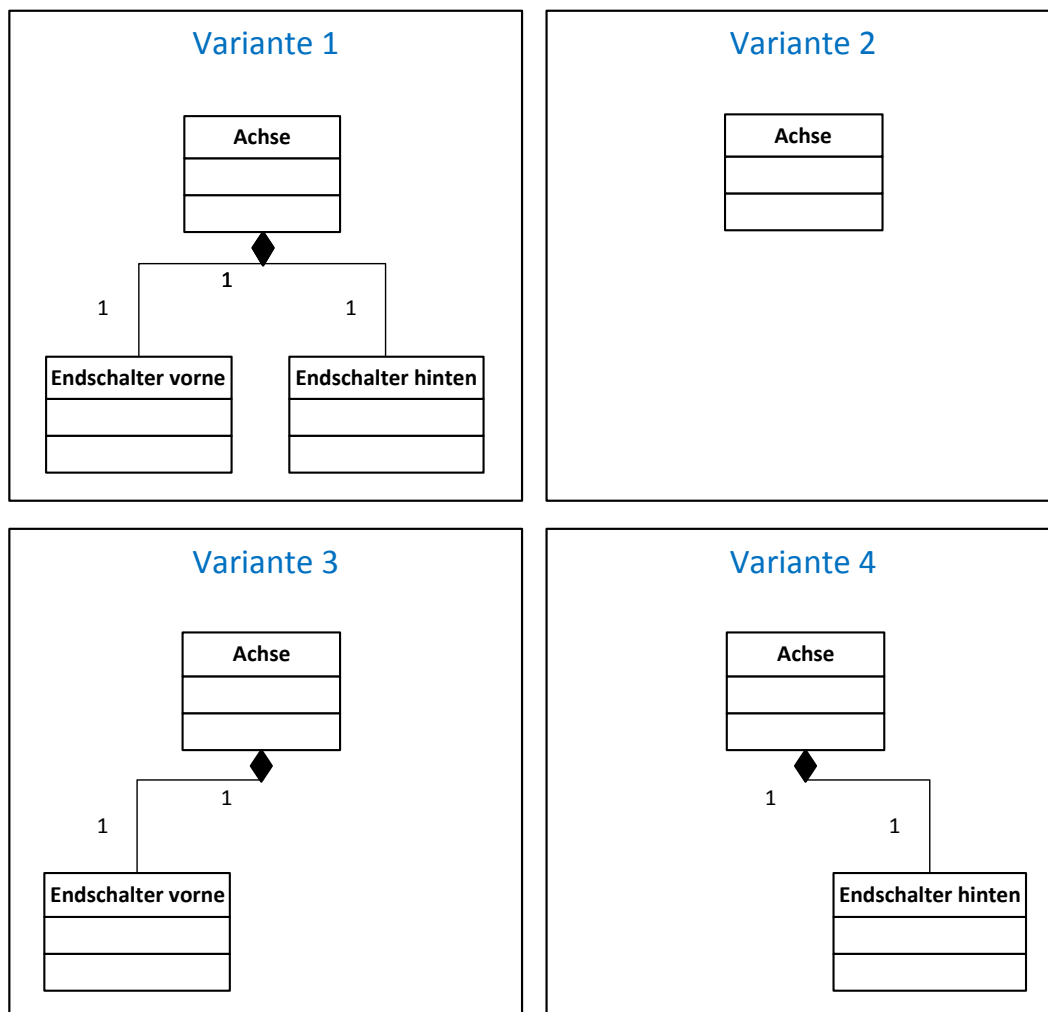


Bild 8.2: Entwurfsmuster Kopiervorlage

Grund für das Anlegen solcher Kopiervorlagen ist häufig, dass zwar ein Baukasten vorliegt, aus dem Varianten und Optionen frei zusammengestellt werden können; die Auswahl zulässiger Kombinationen ist jedoch durch einen Katalog verfügbarer Artikel eingeschränkt. Bei der Konfiguration werden die Anforderungen des Kunden auf eine Datenbank der verfügbaren Artikel gespiegelt und so Schritt für Schritt die jeweils passende Lösung herausgefiltert.

Diese Art der Modellierung eignet sich dann, wenn die Produktpalette eine statische Varianz aufweist. D.h. Varianten und Optionen sind zwar vorhanden, die möglichen

Kombinationen sind jedoch eingeschränkt und vorgedacht. Es ist wenig Regelwissen notwendig, da alle zulässigen Kombinationen statisch abgelegt werden.

Ein Beispiel für die Verwendung dieses Musters ist eine große Anlage im Bereich der Verfahrenstechnik, in der immer wieder dieselben Kombinationen aus z.B. Leitungen, Sensoren, Ventilen und Behältern verbaut werden. Für eine Baugruppe aus Behälter, Füllstandsensoren und Ventilen existieren mehrere zulässige Kombinationen, die keinerlei Regelwissen enthalten. Diese Kombinationen werden als statische – d.h. invariante – Kopiervorlagen im Baukasten abgelegt und in einer Anlage beliebig oft instanziiert.

Der größte Vorteil dieses Musters liegt in seiner Einfachheit. Mögliche Kombinationen werden statisch abgelegt, auf in Formeln abgelegtes Regelwissen wird größtenteils verzichtet. Existieren nur wenige mögliche Kombinationen, so trägt die Anwendung des Entwurfsmusters zur Übersichtlichkeit des Baukastens bei. Der Nachteil dieses Musters ist, dass der Baukasten sehr schnell wächst und ggf. Formeln an verschiedenen Stellen redundant eingetragen sind. Dadurch wird der Pflegeaufwand deutlich erhöht. Ist die Anzahl möglicher Kombinationen zu groß, so kann das Entwurfsmuster Disabler eingesetzt werden, mit Hilfe dessen mehrere Kombinationen als eine einzige Komposition umgesetzt werden können.

8.2.3 Disabler

Mit Hilfe des Disabler-Musters können verschiedene Varianten oder Optionen auf einfache Art und Weise in einer einzigen Komposition (siehe Kapitel 8.2.1) realisiert werden. Dazu wird in eine Komposition ein Baukastenelement als optionale Unterkomponente eingebaut. Über eine Formel wird der Disabler-Mechanismus (ähnlich einem Ein-/Ausschalter) spezifiziert. Ergibt die Formel zur Laufzeit als Ergebnis „true“, wird die Unterkomponente aktiv, ansonsten bleibt sie inaktiv (siehe Bild 8.3, grüne und rote Ampel).

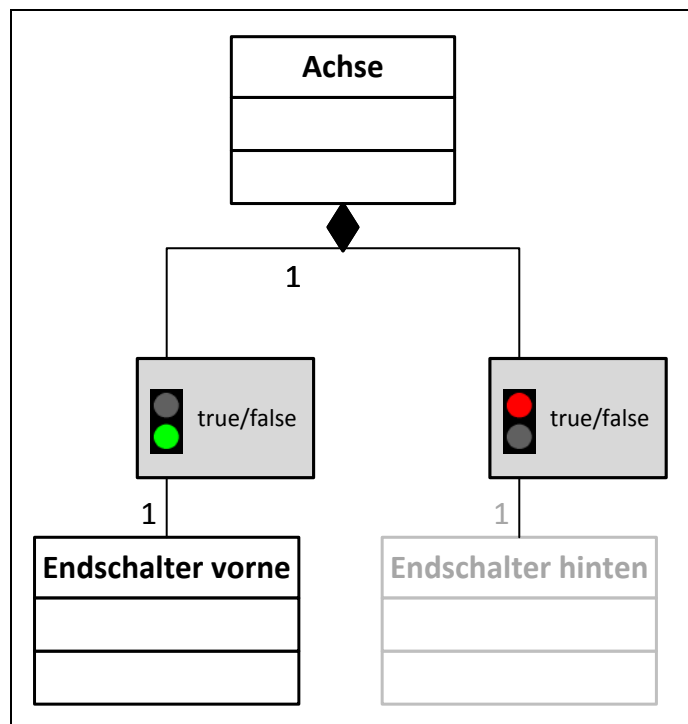


Bild 8.3: Entwurfsmuster Disabler

Das Disabler-Muster eignet sich besonders für Kompositionen mit wenigen Varianten und Optionen. Dabei kann ein geschlossener (endlicher) Konfigurationsraum abgebildet werden, d.h. bei der Entwicklung der Komponente können bereits alle möglichen Konfigurationen der Komponente vorgedacht werden. Bei einem geschlossenen Konfigurationsraum spricht man auch von einer Konfiguration mit Hilfe des Maximalkonzepts. Dies bedeutet, dass bei der Entwicklung der Maximalausbau vorgedacht wird und später Optionen weggelassen werden können bzw. zwischen Varianten gewählt werden kann.

Ein Beispiel für die Anwendung des Entwurfsmusters Disabler ist eine Achse, die immer zwei Anschläge hat, ein dritter kann optional bestellt werden. In diesem Fall bietet es sich an, die beiden ersten Anschläge „fest“ einzubauen, den dritten hingegen mit einem Disabler. Durch die Konfiguration wird der dritte Anschlag dann dynamisch aktiv/inaktiv geschaltet.

Das Disabler-Muster ist leicht verständlich und bei kleinen Modellen sehr übersichtlich. Da im Gegensatz zum Kopiervorlagen-Muster weniger Komponenten im Baukasten angelegt werden, die bei unterschiedlichen Anwendungsfällen benutzt werden können, steigt der Grad der Wiederverwendung. Bei Komponenten mit zahlreichen Optionen, von denen immer nur wenige gleichzeitig aktiv sind, empfiehlt sich jedoch das Erweiterungspunkt-Muster. Der Grund hierfür ist, dass das Modell bei Anwendung des Disabler-Musters auf Grund der Vielzahl der Elemente und dem redundanten Vorkommen von Regelwissen unübersichtlich und aufwendig zu warten wird.

8.2.4 Erweiterungspunkt

Beim Entwurfsmuster Erweiterungspunkt enthält eine Komponente einen Einfügeplatz, dessen Typ ein vorher definiertes Interface ist. Gleich wie beim Disabler existiert auch bei diesem Konzept eine Formel, deren Ergebnis zur Laufzeit darüber bestimmt, ob die Unterkomponente aktiviert wird oder nicht. In dem Moment, in dem durch den Nutzer die Aktualisierung der Erweiterungspunkte angestoßen wird, werden Komponenten eingefügt bzw. wieder entfernt, die als Schnittstelle das im Einfügeplatz definierte Interface besitzen (siehe Bild 8.4, Steckersymbol).

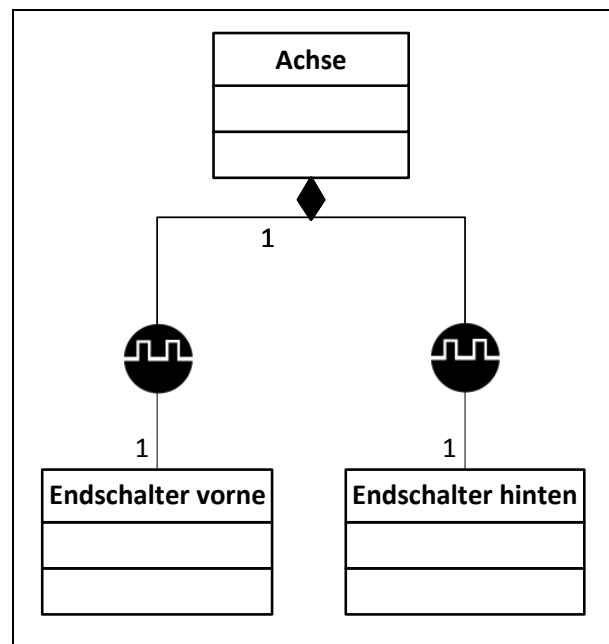


Bild 8.4: Entwurfsmuster Erweiterungspunkt

Durch den Einsatz des Erweiterungspunkt-Musters wird ein Modell skalierbar und performant, da Komponenten jeweils zu dem Zeitpunkt hinzugefügt werden, wenn sie benötigt werden. Deshalb eignet sich das Muster beispielsweise besonders gut für eine mehrstufige Konfiguration. So kann beispielsweise ein Vertriebsmitarbeiter auf einem sehr rudimentären, schlanken Modell arbeiten und dort eine grobe Konfiguration einer Maschine oder Anlage vornehmen. Nach der anfänglichen Vertriebskonfiguration beim Kunden fügt der Elektrokonstrukteur über Erweiterungspunkte z.B. Antriebe, Sensoren und Steuerung hinzu. Erst im eigentlichen Engineering-Prozess wird das Modell dann über Erweiterungspunkte um die disziplinspezifischen Komponenten erweitert, die Fragmente für die anschließende Generierung der Projektunterlagen enthalten.

Analog zum Disabler-Muster werden auch beim Erweiterungspunkt Übersichtlichkeit und Grad der Wiederverwendung erhöht, da mit wenigen konfigurierbaren Komponenten viele Anwendungsfälle erschlagen werden können.

Den genannten Vorteilen des Musters steht ein höherer Implementierungs- und auch Einarbeitungsaufwand als beim Disabler-Muster gegenüber. Daher ist individuell zu entscheiden, ob sich der Aufwand lohnt.

Soll zur Laufzeit nicht nur entschieden werden, ob eine Komponente auf einen Erweiterungspunkt instanziiert wird oder nicht, sondern auch eine Auswahl unter verschiedenen möglichen Komponenten getroffen werden, so kann das Entwurfsmuster Erweiterungspunkt mit Auswahlformel eingesetzt werden.

8.2.5 Erweiterungspunkt mit Auswahlformel

Geht es darum, zur Laufzeit aus verschiedenen passenden Unterkomponenten eine auszuwählen, kann ein Erweiterungspunkt mit Auswahlformel verwendet werden. Das Ergebnis der Auswahlformel spezifiziert den Typ der einzufügenden Komponente (siehe Bild 8.5).

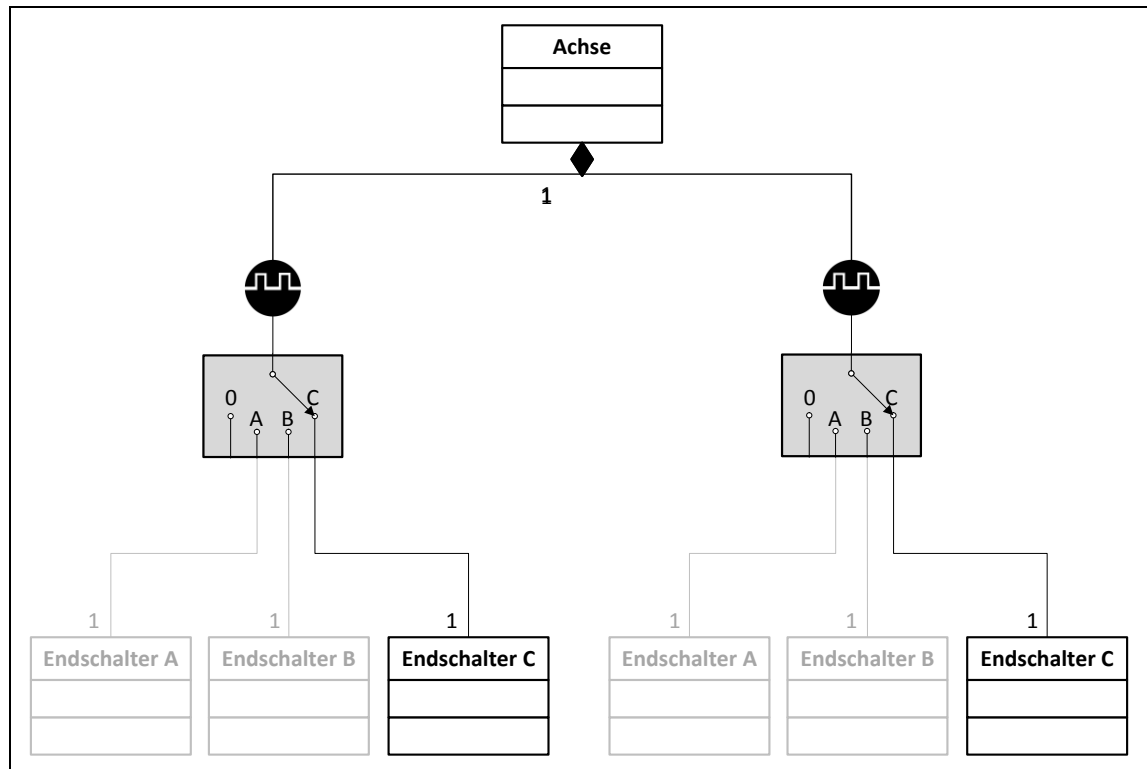


Bild 8.5: Entwurfsmuster Erweiterungspunkt mit Auswahlformel

Durch die Verwendung von Erweiterungspunkten mit Auswahlformel wird vermieden, viele per Disabler eingebaute, inaktive Komponenten im Modell zu haben. Die Auswahlformel ermöglicht es, nicht für jede wählbare Komponente einen eigenen Erweiterungspunkt definieren zu müssen. Mit Hilfe des Zugriffs auf einen Parameterwert können auf einen einzigen Erweiterungspunkt unterschiedliche Komponenten instanziiert werden. Dadurch bleiben das Modell schlank und Berechnungszeiten kurz.

Die Anwendung des Musters ist immer dann sinnvoll, wenn aus einer größeren Auswahl von Komponenten jeweils eine einzelne eingesetzt wird. D.h. es wird aus mehreren sich gegenseitig ausschließenden Varianten eine ausgewählt. Die Auswahlformel kann dabei auf Parameter zugreifen, die komplexe Berechnungen beinhalten. Ebenfalls möglich ist das Auslesen eines zentralen Parameters, so dass mehrere Erweiterungspunkte innerhalb einer Maschine oder Anlage dieselbe Komponente oder beispielsweise eine Komponente desselben Herstellers instanziiieren (siehe auch Kapitel 8.3.3 Herstellerbibliothek und Kapitel 8.4.3 Zentralsteuerung).

Der Vorteil gegenüber dem Disabler-Muster (siehe Kapitel 8.2.3) ist, dass nur die aktuell passende Komponente instanziiert wird, es verbleiben keine inaktiven Komponenten im Modell. Dadurch ist das Modell übersichtlich und performant. Das Muster bietet über die Vorteile des oben beschriebenen Erweiterungspunkts ohne Auswahlformel hinaus die Möglichkeit, unterschiedliche Komponenten zu instanziiieren und die Auswahl von Komponenten zentral zu steuern.

Beim Zugriff auf Parameterwerte innerhalb der Auswahlformel ist darauf zu achten, dass keine „schwingenden“ Modelle gebaut werden. Sind beispielsweise zwei Ergebnisse der Auswahlformeln von unterschiedlichen Erweiterungspunkten voneinander abhängig (z.B. Komponente A ist von Typ 1, wenn Komponente B von Typ 1 ist; Komponente B ist von Typ 2, wenn ...), so ändert sich das Ergebnis der Auswahlformeln bei jeder neuen Berechnung. Somit wird kein deterministisches Verhalten des Modells erreicht, das Modell „schwingt“.

Ein Beispiel für die Anwendung des Musters ist die Messung einer Temperatur, wofür verschiedene Sensoren unterschiedlicher Hersteller eingesetzt werden können. Der Konstrukteur wählt je nach Anforderung einen Sensor aus. Anstatt alle Sensoren mit Disablern zu implementieren und bei Bedarf einen zu aktivieren bzw. für alle Sensoren jeweils einen spezifischen Erweiterungspunkt zu definieren, wird ein einziger Erweiterungspunkt mit Auswahlformel verwendet. Die Auswahlformel greift auf einen Parameterwert zu, in dem das Ergebnis einer Berechnung oder die Auswahl des Konstrukteurs vorliegt.

Ein weiteres Beispiel ist die Generierung von Daten und Dokumenten in unterschiedlichen Kundenstandards. Für jeden Kunden wird beispielsweise für ein und dieselbe Funktionalität ein eigenes Schaltplanmakro erstellt. Die Verwendung des jeweiligen Makros wird zentral über die Auswahl des Kunden gesteuert (siehe auch Kapitel 8.4.3 Zentralsteuerung und Kapitel 8.3.3 Herstellerbibliothek).

Soll neben der Auswahl von Komponenten unterschiedlichen Typs zusätzlich auch die Anzahl der auf einen Erweiterungspunkt zu instanzierenden Komponenten variieren, so kann das Entwurfsmuster Erweiterungspunkt mit Auswahlformel und Anzahl angewendet werden.

8.2.6 Erweiterungspunkt mit Auswahlformel und Anzahl

Beim Erweiterungspunkt mit Auswahlformel und Anzahl kann zusätzlich zum Typ der einzufügenden Komponente auch die Anzahl der zu instanzierenden Komponenten variiert werden.

Dieses Konzept bietet vielfältige Möglichkeiten, auf übersichtliche und performante Art und Weise komplexe Komponentenstrukturen dynamisch zu erstellen. Dadurch kann der Einsatz vieler vorinstanzierter Komponenten mit Disablern vermieden werden. Mit Hilfe eines einzigen Erweiterungspunkts mit Auswahlformel kann auf Parameter zugegriffen werden, die die Anzahl und Art der zu verwendenden Komponenten spezifizieren (siehe Bild 8.6).

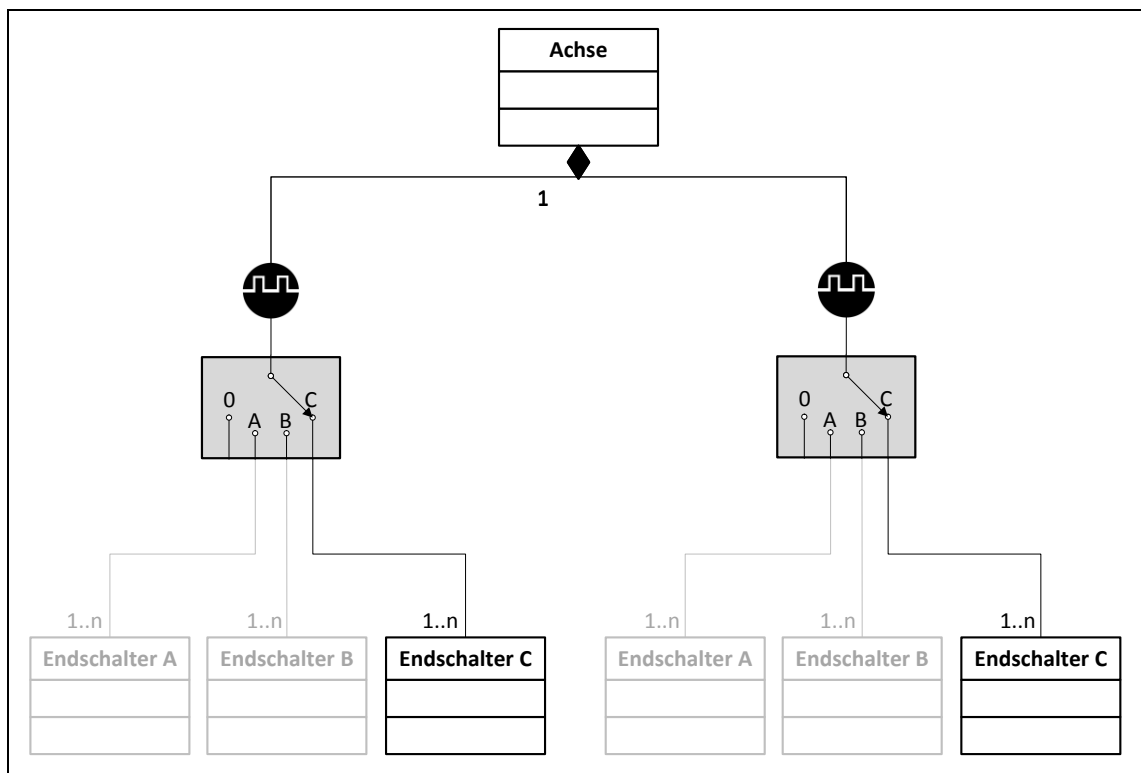


Bild 8.6: Entwurfsmuster Erweiterungspunkt mit Auswahlformel und Anzahl

Während der Komponententyp beispielsweise wieder zentral gesteuert werden kann (siehe 8.4.3 Zentralsteuerung), kann die Anzahl z.B. abhängig sein von der Größe der Anlage oder der Anzahl der verbauten Optionen.

Ein Beispiel hierfür ist der Steuerungstyp in einer Maschine oder Anlage, der abhängig vom Kundenwunsch verwendet wird. Die Anzahl der benötigten Busklemmen hängt von der Größe der Maschine oder Anlage bzw. von der Anzahl der gewählten optionalen Funktionen ab. Mit Hilfe eines einzigen Erweiterungspunkts werden nun z.B. analoge Eingangskarten der Steuerung in der richtigen Anzahl und vom gewünschten Hersteller (siehe 8.3.3 Herstellerbibliothek) verbaut. Anzahl und Hersteller werden dabei aus Parametern ausgelesen, denen eine automatische Berechnung bzw. eine Auswahl des Konstrukteurs/Kunde zu Grunde liegt.

Die Vor- und Nachteile entsprechen denen des Erweiterungspunkt-Musters mit Auswahlformel (siehe Kapitel 8.2.5). Die zusätzliche Möglichkeit, die Anzahl der instanziierten Komponenten zu variieren, ist ein mächtiges Werkzeug, um auch bei kompliziertem Regelwerk bezüglich des Typs und der Anzahl der zu verwendenden Komponenten das Modell übersichtlich und performant zu halten.

Die bisher beschriebenen Erzeugungsmuster helfen dabei, Komponenten im Baukasten so anzulegen, dass Instanzen anschließend performant, übersichtlich und mit einem hohen Grad der Wiederverwendung erzeugt werden können. Für eine Verbesserung der Übersichtlichkeit und des Grades der Wiederverwendung innerhalb der Baukästen werden jedoch weitere Entwurfsmuster benötigt. Diese so genannten Strukturmuster sollen dabei unterstützen, Baukästen zu strukturieren.

8.3 Strukturmuster

Strukturmuster unterstützen den Baukastenentwickler dabei, Komponenten und Parameter strukturiert im Baukasten zu organisieren und abzulegen. Dadurch wird die Übersichtlichkeit im Baukasten erhöht, was wiederum dazu führt, dass Komponenten und Parameter öfter wiederverwendet werden und die Änderbarkeit verbessert wird. Ein wichtiger Punkt bei der Erstellung von Baukästen ist dabei die Festlegung der Architektur, die die Ebenendefinition einer Maschine oder Anlage enthält. Dabei soll das Entwurfsmuster Trennung Architektur – Baukasten unterstützen, das im Folgenden genauer erläutert wird.

8.3.1 Trennung Architektur – Baukasten

Die Architektur, die die Definition der Ebenen einer Maschine oder Anlage beinhaltet, sowie die zulässigen Vater-Kind-Beziehungen zwischen Baukastenelementen einzelner Ebenen, wird getrennt von den tatsächlich instanziierten, also nicht abstrakten Komponenten angelegt (siehe Bild 8.7).

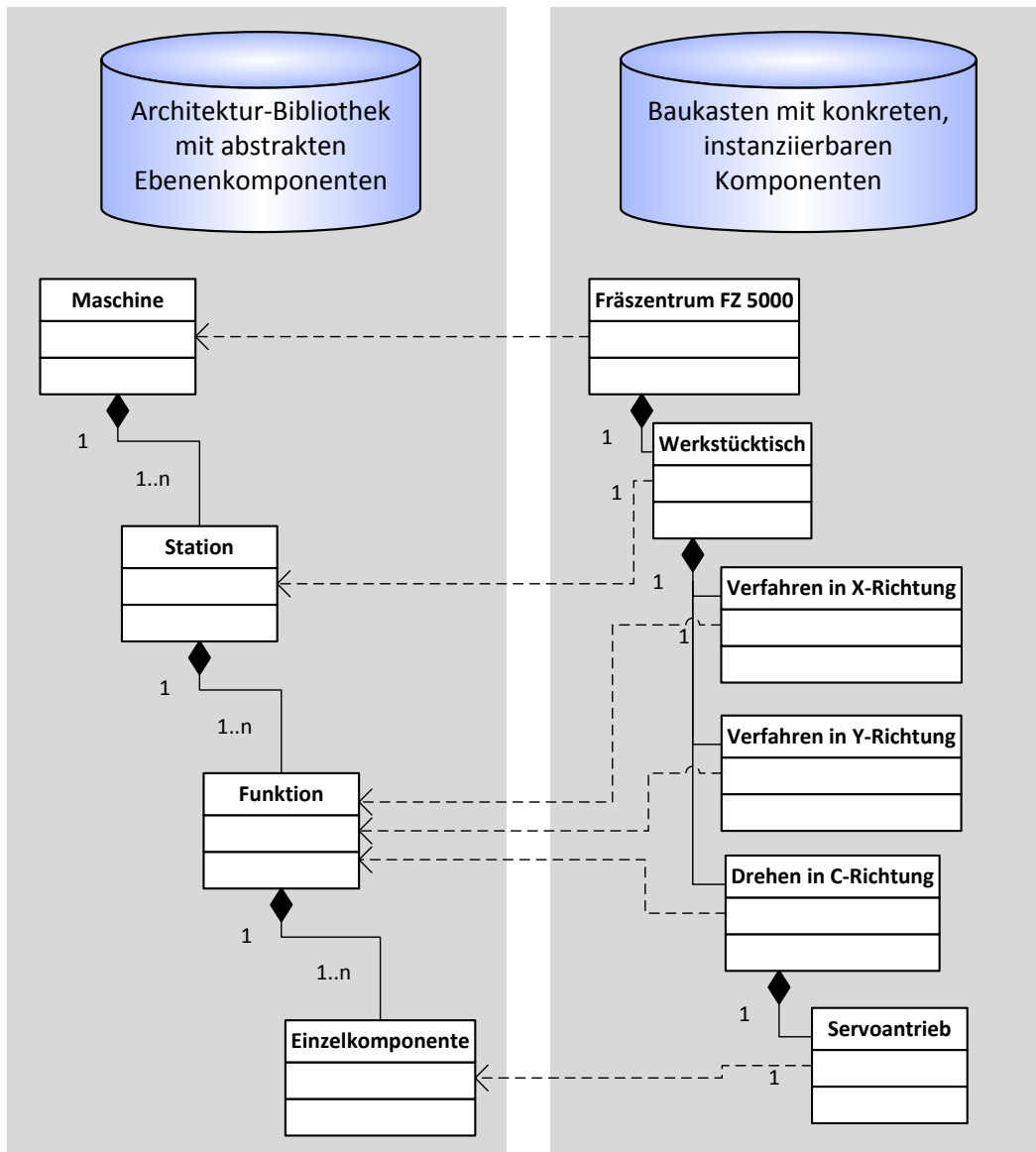


Bild 8.7: Entwurfsmuster Trennung Architektur – Baukasten am Beispiel Fräszentrum

Das Bild zeigt an einem Beispiel, wie ein Werkzeugmaschinenhersteller seinen Baukasten mit Hilfe einer Architekturbibliothek bestehend aus den Ebenen Maschine, Station, Funktion und Einzelkomponente strukturiert hat. Eine Maschine kann eine oder mehrere Stationen enthalten, eine Station eine oder mehrere Funktionen usw. Soll nun beispielsweise eine weitere Ebene Baugruppe zwischen Funktion und Ein-

zelkomponente eingefügt werden, so kann diese Änderung in der Architekturbibliothek an einer einzigen Stelle vorgenommen werden. Anschließend können im Baukasten neue Baukastenelemente von der Oberklasse Baugruppe angelegt und beispielsweise durch Komposition in anderen Baukastenelementen verbaut werden.

Diese Vorgehensweise ist vorteilhaft, weil ein Baukasten während der Entwicklung, aber auch über die gesamte Lebensdauer zahlreiche Änderungen durchläuft. Neben Optimierungen oder Aktualisierungen einzelner Baukastenelemente oder dem Hinzufügen neuer Komponenten kommt es gelegentlich auch vor, dass strukturelle Anpassungen vorgenommen werden müssen. Beispielsweise wird eine zusätzliche Ebene benötigt oder eine vorhandene soll entfernt werden. Durch die Trennung von Architektur und Baukasten werden Änderungsaufwände deutlich reduziert, da Ebenen und Vater-Kind-Beziehungen an einer einzigen Stelle definiert sind und alle Baukastenelemente von diesen abstrakten „Ebenenkomponenten“ abgeleitet sind.

Durch die Trennung von Architektur und Baukasten kann der Änderungsaufwand von Baukästen bei strukturellen Änderungen erheblich gesenkt werden. Daher ist die Anwendung dieses Musters grundsätzlich immer zu empfehlen. Dieses Entwurfsmuster ist außerdem Grundlage des 4-Ebenen-Komponentenmodells, das durch die strikte Trennung von Funktionalität, technischer Realisierung und Dokumentationsanteilen zu einer Erhöhung des Grads der Wiederverwendung beiträgt.

8.3.2 4-Ebenen-Komponentenmodell

Ein Ziel bei der Anwendung der Methoden des Funktionalen Engineerings ist, ein Baukastenelement so zu definieren, dass es an möglichst vielen unterschiedlichen Stellen wiederverwendet werden kann, also zur Standard-Komponente wird. Dadurch sinkt die Anzahl der benötigten Komponenten und somit auch der Erstellungs- und Pflegeaufwand, die Übersichtlichkeit wird erhöht.

8.3 Strukturmuster

Das Entwurfsmuster 4-Ebenen-Komponentenmodell baut auf dem Entwurfsmuster Trennung Architektur – Baukasten auf und schlägt die Abbildung einer mechatronischen Komponente als Modell mit vier Ebenen vor (siehe Bild 8.8).

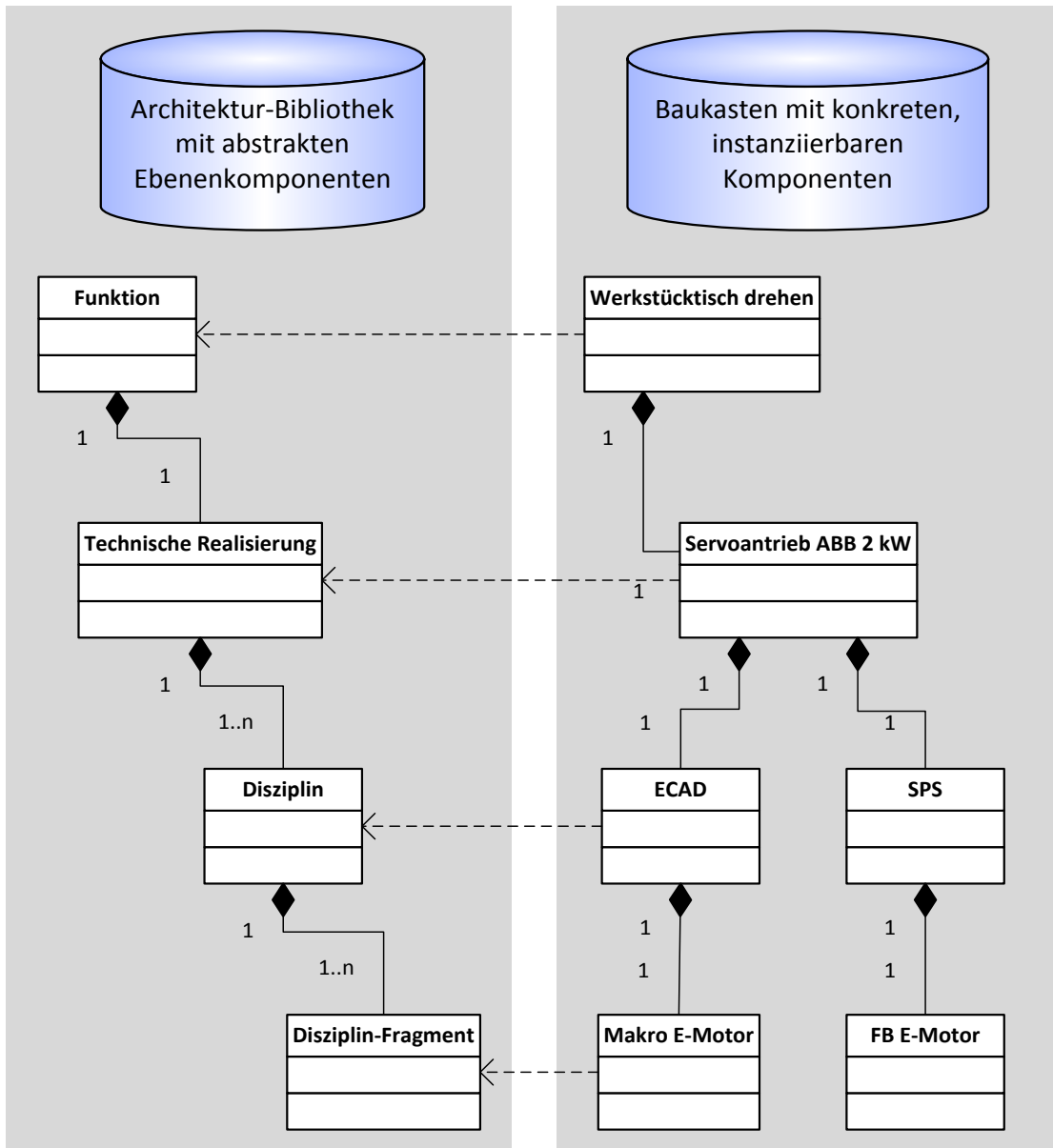


Bild 8.8: Entwurfsmuster 4-Ebenen-Komponentenmodell

Diese Ebenen sind von oben nach unten:

- Funktion

- Technische Realisierung
- Disziplin
- Disziplin-Fragment (Dokument-/Daten-Fragment)

Durch den Aufbau einer mechatronischen Komponente mit mehreren Ebenen kann die Wiederverwendbarkeit der enthaltenen Objekte erhöht werden. Beispielsweise könnte in den Funktionen Werkzeugmagazin, Sicherheitstür und Späneförderer eines Werkzeugmaschinenherstellers jeweils der gleiche Servoantrieb als konkrete technische Realisierung verbaut werden. Andererseits ist es jedoch auch möglich, den standardmäßig verbauten Motor auf Kundenwunsch durch den eines anderen Herstellers zu ersetzen. Durch die Trennung von Funktion und technischer Realisierung wird somit eine höhere Flexibilität bei gleichzeitiger Reduktion von Komponenten erreicht.

Die dritte Ebene „Disziplin“ dient der Strukturierung und somit besseren Übersichtlichkeit. Beinhaltet eine mechatronische Komponente sehr viele Fragmente (vierte Ebene) aus unterschiedlichen Disziplinen, so können diese in einzelnen Disziplin-Objekten gekapselt werden. Der Vorteil dieser Vorgehensweise ist neben einer besseren Übersichtlichkeit, dass disziplinspezifische Inhalte beispielsweise während der Konfiguration durch den Vertrieb noch unterdrückt werden und das Modell somit schlank bleibt. Erst bei der Generierung der Projektunterlagen in der entsprechenden Entwicklungsabteilung erfolgt die Einbindung dieser Inhalte.

Dieses Muster sollte bei großen oder komplexen Modellen angewendet werden, um größtmögliche Flexibilität, Übersichtlichkeit und Skalierbarkeit des Datenvolumens zu gewährleisten. Beispiele für die Anwendung des Musters finden sich auch in Kapitel 9 Firmenübergreifende Basisbaukästen.

Dem etwas höheren Implementierungsaufwand stehen mehrere Vorteile gegenüber. Zum einen wird der Grad der Wiederverwendung im Modell dadurch erhöht, dass Funktion und technische Realisierung voneinander getrennt werden. Somit kann

eine technische Realisierung für unterschiedliche Funktionen verwendet werden, umgekehrt können jedoch auch unterschiedliche technische Realisierungen in ein und derselben Funktion benutzt werden.

Durch die Zwischenebene Disziplin wird die Übersichtlichkeit erhöht, da alle Fragmente in ihrem zugehörigen Disziplinobjekt abgelegt werden. Des Weiteren kann in Kombination mit dem Erweiterungspunkt-Muster (siehe Kapitel 8.2.4) eine hohe Skalierbarkeit des Modells erreicht werden, da Disziplininhalte beispielsweise erst während der Engineering- und Generierungsphase hinzugefügt werden, der Vertriebsmitarbeiter zuvor hingegen noch mit einem schlanken und performanten Modell arbeiten kann.

Soll es auf der Ebene der technischen Realisierung möglich sein, Komponenten unterschiedlicher Hersteller einzubinden, so kann auf dieser Ebene zusätzlich das Entwurfsmuster Herstellerbibliothek eingesetzt werden, das im Folgenden genauer erläutert wird.

8.3.3 Herstellerbibliothek

Das Entwurfsmuster Herstellerbibliothek schlägt, aufbauend auf dem Entwurfsmuster 4-Ebenen-Komponentenmodell, einen zweistufigen Aufbau des Modells auf der Ebene technische Realisierung vor. In einer firmenspezifischen Bibliothek liegen sogenannte Hüllkomponenten. Über Erweiterungspunkte (siehe Kapitel 8.2.4) werden Komponenten mit konkreter Realisierung eingefügt. Zwischen Hüllkomponente und konkreter Komponente existiert ein Interface, über das passende Elemente zueinander finden. Die konkreten Komponenten sind in einer eigenen Bibliothek abgelegt (Herstellerbibliothek). Durch den Import einer Herstellerbibliothek wird global für das ganze Projekt festgelegt, dass nur Komponenten dieser Bibliothek benutzt werden (siehe Bild 8.9).

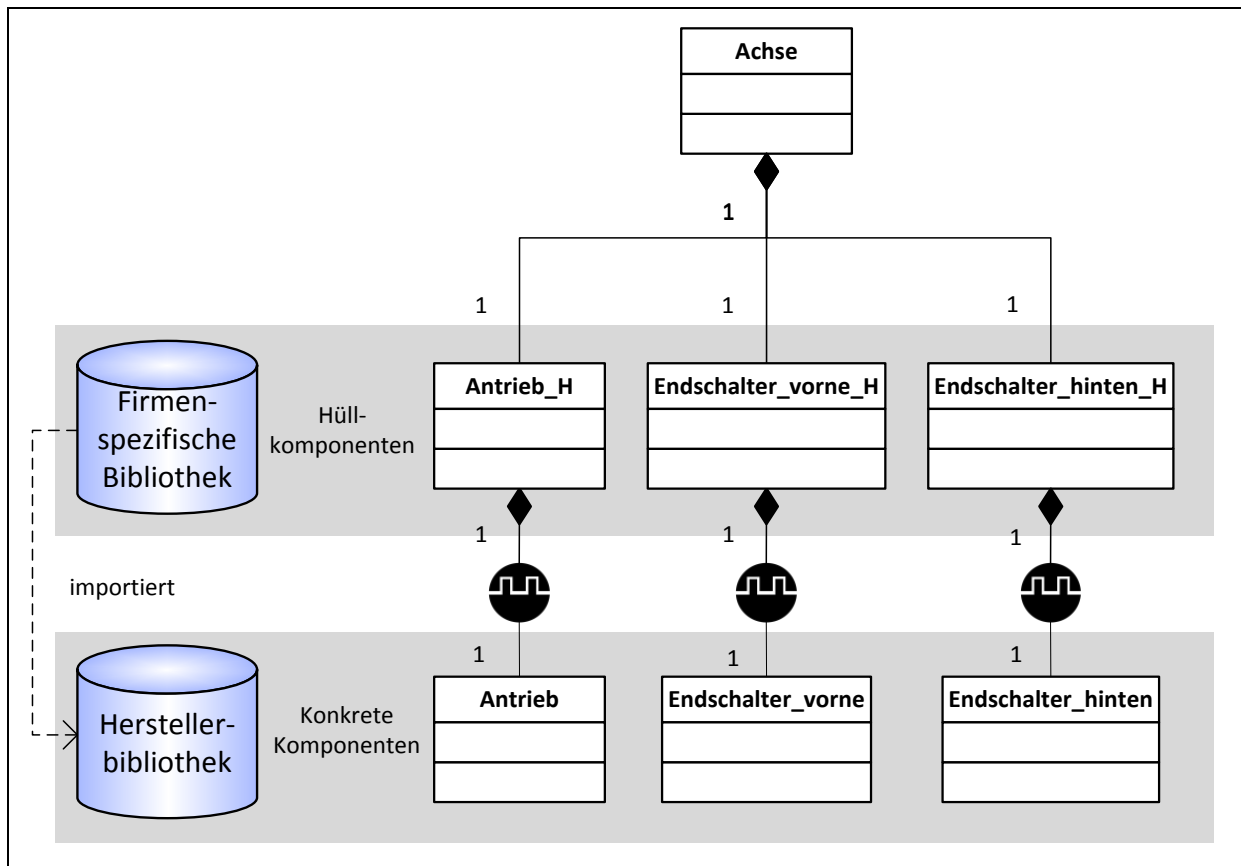


Bild 8.9: Entwurfsmuster Herstellerbibliothek

Das Muster bietet die Möglichkeit, durch Import einer anderen Bibliothek mehrere Komponenten gleichzeitig auszutauschen. Mehrere Teilefamilien mit äquivalenten Komponenten verschiedener Hersteller können in unterschiedlichen Bibliotheken abgelegt werden. Gesteuert durch den Importmechanismus werden nur Komponenten aus einer der Bibliotheken verwendet. Alternativ kann die Auswahl einer Herstellerbibliothek auch über das Entwurfsmuster Zentralsteuerung (siehe Kapitel 8.4.3) erfolgen. In beiden Fällen muss beispielsweise bei der Auswahl eines anderen Herstellers von Sensoren nicht jeder Sensor einzeln angepasst werden, sondern die Aktualisierung der Komponenten geschieht automatisch. Mit Hilfe des Entwurfsmusters Herstellerbibliothek kann auch das Konzept der wiederverwendbaren Baukästen realisiert werden (siehe Kapitel 9).

Die benötigten Herstellerbibliotheken werden beispielsweise vom jeweiligen Komponentenhersteller entwickelt und den Anwendern zur Verfügung gestellt. Wünscht nun ein Kunde eines Maschinen- oder Anlagenherstellers, dass nur Sensoren eines bestimmten Herstellers verwendet werden, so wird die entsprechende Bibliothek eingebunden und die enthaltenen Komponenten werden über Erweiterungspunkte instanziiert.

Ein weiterer Anwendungsfall liegt bei einem Engineeringdienstleister vor, der beispielsweise Produktionsanlagen für verschiedene Automobilhersteller plant. Jeder seiner Kunden gibt verschiedene Rahmenbedingungen vor, beispielsweise die zu verwendenden Steuerungen und Komponentenhersteller oder den Aufbau eines Schaltschranks. Teile des Baukastens, die bei allen Kunden gleich sind, werden in einer allgemeinen Bibliothek abgelegt, kundenspezifische Teile in jeweils individuellen Bibliotheken. Je nach dem, für welchen Kunden ein Projekt erstellt werden soll, werden zu Beginn die entsprechenden Bibliotheken importiert.

Der Vorteil des Musters liegt in der Austauschbarkeit von konkreten Komponenten und Regeln für ein ganzes Projekt. Änderungswünsche des Kunden bezüglich der Verwendung einer bestimmten Herstellerbibliothek können bis kurz vor Produktionsstart noch akzeptiert werden. Der Austausch der Komponenten und die Generierung von neuen Fertigungsunterlagen geschehen automatisch.

Diesem Vorteil steht der Nachteil gegenüber, dass der Implementierungsaufwand sehr hoch ist. Dies ist vor allem dann der Fall, wenn Herstellerbibliotheken nicht äquivalent sind, d.h. Leistungswerte von Antriebsfamilien verschiedener Hersteller sind unterschiedlich. Dann muss das Regelwerk erweitert werden, was außerdem einen hohen Pflegeaufwand mit sich bringt.

Neben der übersichtlichen Strukturierung von Komponenten ist es ebenso wichtig, die in den Komponenten verwendeten Parameter so abzulegen, dass diese leicht gefunden und somit standardisiert und wiederverwendet werden können. Dabei unterstützt das Entwurfsmuster Parameter-Bibliothek.

8.3.4 Parameter-Bibliothek

Bei der Entwicklung eines Baukastens werden viele Parameter unterschiedlichen Typs benötigt. Um den Überblick über alle vorhandenen Parameter zu behalten und das Anlegen doppelter Parameter zu vermeiden, ist es von Vorteil, alle Parameter zentral in einer Bibliothek anzulegen. Dadurch verringern sich der Pflegeaufwand auf Grund einer verbesserten Übersichtlichkeit sowie die Wahrscheinlichkeit von Fehlern durch das Verwenden eines falschen Parameters.

Deshalb schlägt das Entwurfsmuster Parameter-Bibliothek die Ablage aller benötigten Parameter in einer globalen Bibliothek vor (siehe Bild 8.10).

8.3 Strukturmuster

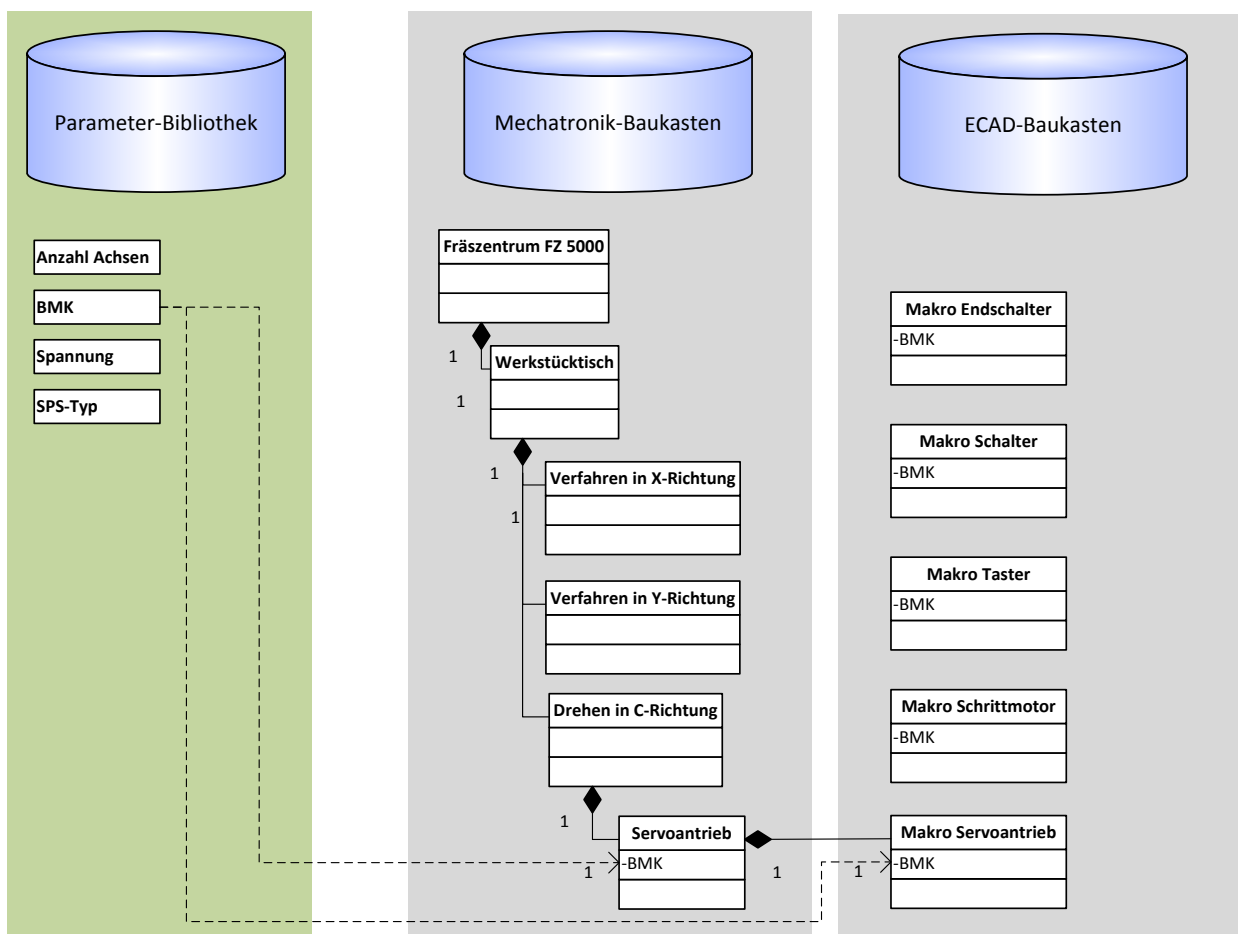


Bild 8.10: Entwurfsmuster Parameter-Bibliothek

Der dargestellte Baukasten eines Werkzeugmaschinenherstellers besteht wiederum aus mehreren Baukästen, u.a. aus einem Mechatronik-Baukasten und einem ECAD-Baukasten. Im mechatronischen Baukastenelement Servoantrieb ist ein ECAD-Makro enthalten. Beide Objekte enthalten denselben Parameter BMK. Dieser ist nicht innerhalb eines der Baukästen Mechatronik oder ECAD oder gar in beiden enthalten, sondern in einer Parameter-Bibliothek, die in beiden Baukästen sichtbar ist. Enthält der Parameter beispielsweise eine Formel zur Berechnung eines Standardwerts, so wird dieser folglich nur an einer einzigen Stelle gepflegt.

Das Muster sollte immer dann angewendet werden, wenn ein Baukasten aus mehr als einer Bibliothek besteht. Durch das Anlegen einer eigenen Parameter-Bibliothek

wird die Existenz redundanter Parameter unterbunden, wodurch die Übersichtlichkeit verbessert und der Pflegeaufwand erheblich gesenkt werden kann.

Neben der strukturierten Ablage von Parametern ist es ebenfalls wichtig, dass die in den Komponenten enthaltenen Parameter ihre Werte gegenseitig so übergeben, dass das Verhalten des Modells performant und einfach nachvollziehbar ist. Um letzteres zu erreichen ist es notwendig, dass Parameter nach einer durchgängigen Methodik organisiert sind und miteinander kommunizieren, um einen übersichtlichen und somit leicht zu ändernden Baukasten zu erhalten. Im Folgenden werden mehrere Verhaltensmuster erläutert, die den Baukastenentwickler bei der Verwendung von Parametern und der Ein- und Weitergabe von Parameterwerten methodisch unterstützen sollen.

8.4 Verhaltensmuster

Verhaltensmuster beschreiben eine Art und Weise, wie Baukastenelemente und Instanzen mittels einer geeigneten Organisation von Parametern zusammenarbeiten und Zuständigkeiten aufteilen. Eine sehr einfache und pragmatische Möglichkeit, Parameterwerte über mehrere Komponenten hinweg weiterzugeben, schlägt das Entwurfsmuster Schnittstellenparameter vor.

8.4.1 Schnittstellenparameter

Beim Entwurfsmuster Schnittstellenparameter wird auf Parameter aus übergeordneten Komponenten nicht direkt per statischer Navigationsformel zugegriffen, sondern über Schnittstellenparameter des direkten Vaterobjekts. Parameterwerte werden somit Stufe für Stufe nach unten vererbt (siehe Bild 8.11).

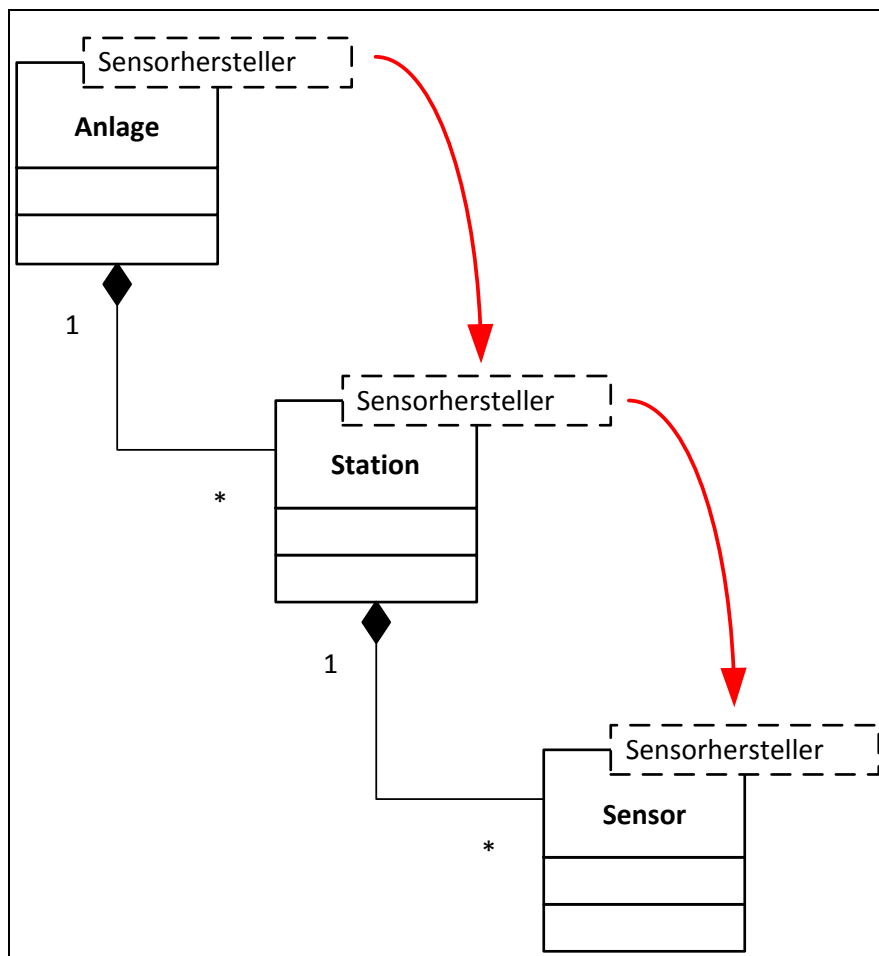


Bild 8.11: Entwurfsmuster Schnittstellenparameter

In einer großen Anlage sollen nur Sensoren eines bestimmten Herstellers verbaut werden. Der Zugriff auf den Parameter `Sensorhersteller` geschieht nun über Schnittstellenparameter über mehrere Stufen von Ebene zu Ebene hinweg (siehe auch 8.4.3 Zentralsteuerung). Wird nun eine weitere Ebene eingefügt, wird lediglich wieder ein Schnittstellenparameter eingefügt. Es müssen keinerlei Formeln angepasst werden, der Zugriff funktioniert weiterhin.

Dieses Muster erhöht die Übersichtlichkeit und damit die Änderbarkeit eines Modells. Werden Parameterwerte über statische Navigationsformeln (z.B. zwei Ebenen nach oben, Unterobjekt `y`, Parameter `x`) abgefragt, steigt der Änderungsaufwand stark an, wenn beispielsweise Zwischenobjekte eingefügt werden. In diesem Fall

müssten sämtliche Navigationsformeln manuell angepasst werden. Durch die Verwendung des Schnittstellenparameter-Musters bleibt die Änderbarkeit erhalten.

Das Entwurfsmuster Schnittstellenparameter sollte bei Modellen mit wenigen Parametern, die jeweils nur über ein oder zwei Ebenen zu vererben sind, angewandt werden. Ziel dabei ist es, den Aufwand bei der Änderung von Modellen niedrig zu halten. Dem Nachteil eines etwas höheren Implementierungsaufwand bei der Erstellung von Modellen durch Eintragen von zusätzlichen Parametern auf mehreren Ebenen steht der Vorteil gegenüber, dass bei Modelländerungen wesentlich weniger Formeln angepasst werden müssen.

Bei Modellen mit sehr vielen Parametern, die über viele Ebenen hinweg vererbt werden sollen, steigt der Ressourcenbedarf durch die Vielzahl der Parameterobjekte und Berechnungsschritte stark an. In diesem Fall ist das Entwurfsmuster Dynamische Parametervererbung zu bevorzugen. Oftmals bietet sich die Anwendung beider Muster in ein und demselben Modell an. Dabei wird von Fall zu Fall entschieden, welches Muster für die Vererbung eines bestimmten Parameterwertes geeigneter ist.

8.4.2 Dynamische Parametervererbung

Beim Entwurfsmuster Dynamische Parametervererbung wird auf Parameter aus übergeordneten Komponenten nicht direkt per statische Navigationsformel oder per Schnittstellenparameter zugegriffen, sondern über Formeln, die das Navigationsziel dynamisch berechnen (siehe Bild 8.12).

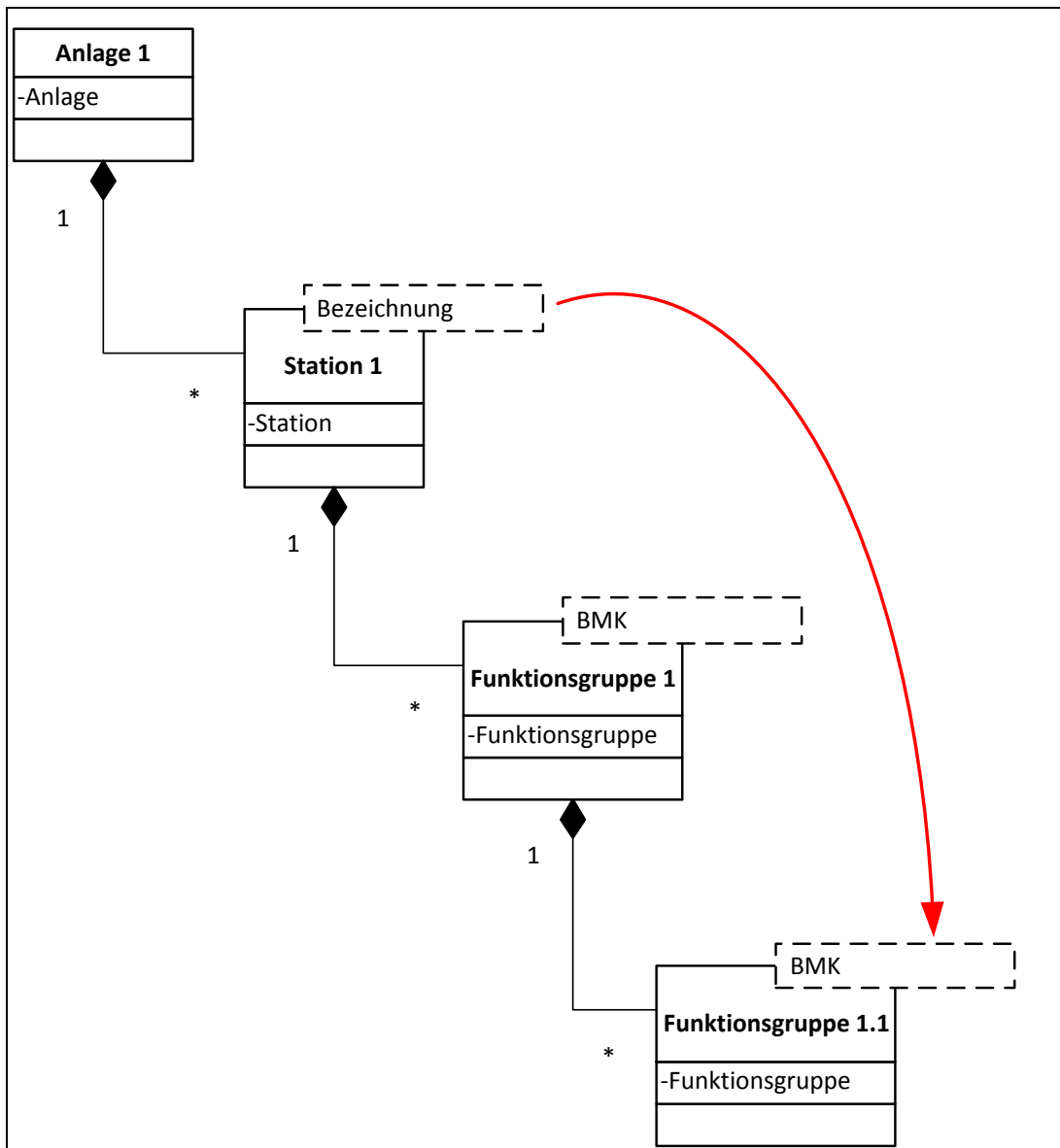


Bild 8.12: Entwurfsmuster Dynamische Parametervererbung

In einer Anlage können beispielsweise Objekte des Typs „Funktionsgruppe“ beliebig tief verschachtelt werden. Das Betriebsmittelkennzeichen (BMK) einer Funktionsgruppe leitet sich jedoch unter anderem aus dem Namen der übergeordneten Station ab. Durch die dynamische Navigation zum übergeordneten Objekt des Typs „Station“ und dem Auslesen des dort vorliegenden Parameters „Bezeichnung“ kann das BMK auch noch bestimmt werden, wenn eine weitere Zwischenebene des Typs Funktionsgruppe eingeführt wurde.

Durch die dynamische Berechnung des Navigationsziels (z.B. übergeordnetes Objekt des Typs x , Parameter y) steigt die Änderbarkeit des Modells. Gleichzeitig kann auf zusätzliche Parameter verzichtet werden, wodurch das Modell performant bleibt.

Werden Parameterwerte anderer Objekte hingegen über statische Navigationsformeln (z.B. zwei Ebenen nach oben, Unterobjekt y , Parameter x) abgefragt, steigt der Änderungsaufwand stark an, wenn beispielsweise Zwischenobjekte eingefügt werden. In diesem Fall müssten sämtliche Navigationsformeln manuell angepasst werden. Bei der Verwendung des Schnittstellenparameter-Musters steigt der Ressourcenbedarf bei Modellen mit vielen Parametern und deren Vererbung über viele Ebenen hinweg stark an.

Deshalb sollte dieses Muster bei Modellen mit vielen Parametern, die jeweils über viele Ebenen vererbt werden, angewandt werden. Dem Nachteil eines etwas höheren Implementierungsaufwands bei der Erstellung von Modellen durch Eintragen von Navigationsformeln steht der Vorteil gegenüber, dass bei Modelländerungen wesentlich weniger Formeln angepasst werden müssen und gleichzeitig der Ressourcenbedarf gering ist.

Bei Modellen mit wenigen Parametern, die über ein oder zwei Ebenen hinweg vererbt werden sollen, ist hingegen das Schnittstellenparameter-Muster (siehe Kapitel 8.4.1) zu bevorzugen, da es einfacher anzuwenden ist.

Bezieht sich die Berechnung sehr vieler Parameterwerte auf einen bestimmten, zentral gelegenen Parameter, so kann aufbauend auf den beiden Mustern Schnittstellenparameter und Dynamische Parametervererbung das Entwurfsmuster Zentralsteuerung angewendet werden.

8.4.3 Zentralsteuerung

Beim Entwurfsmuster Zentralsteuerung ist in der obersten Komponente (Wurzelement) oder einer anderen, in der Hierarchie weit oben gelegenen Komponente, ein

8.4 Verhaltensmuster

Parameter enthalten, von dem Parameterwerte in vielen untergeordneten Komponenten abhängig sind (siehe Bild 8.13, Parameter „Typ Endschalter“).

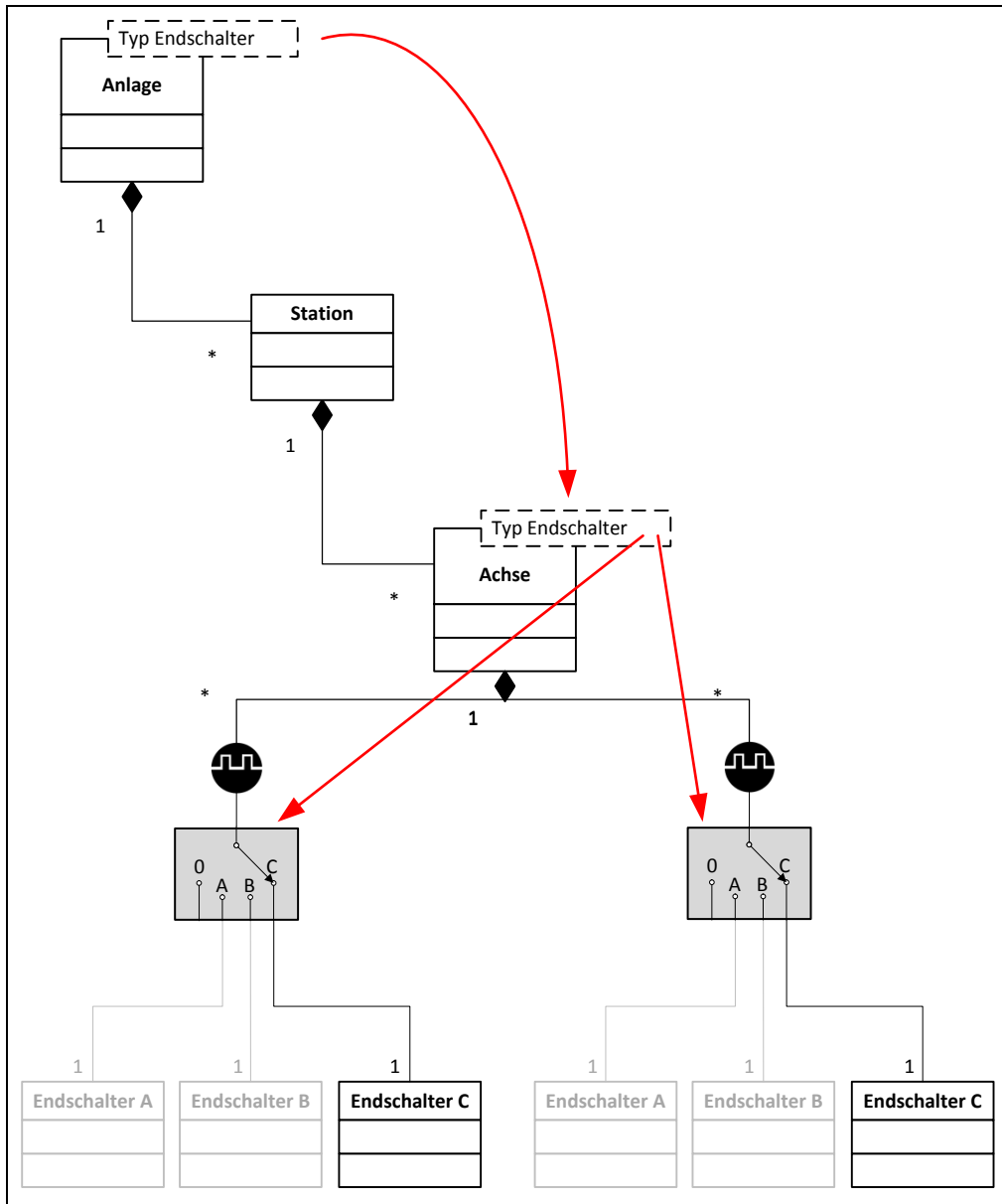


Bild 8.13: Entwurfsmuster Zentralsteuerung

So kann beispielsweise in einem einzigen Parameter die Auswahl der modellweit zu verwendenden Komponenten (Herstellerbibliothek, siehe auch 8.3.3) getroffen werden. Über Erweiterungspunkte mit Auswahlformel (siehe Kapitel 8.2.5) werden

dann abhängig von diesem zentralen Parameter in der obersten Komponente die Elemente einer bestimmten Herstellerbibliothek eingefügt.

Das Muster Zentralsteuerung bietet die Möglichkeit, durch Verändern eines einzigen Parameters modellweit Parametersätze (wie z.B. länderspezifische Frequenz des Wechselstroms) zu verändern oder mehrere Komponenten gleichzeitig auszutauschen. Gesteuert durch den Wert des Parameters werden in letzterem Fall nur Komponenten aus einer Herstellerbibliothek verwendet. Somit muss beispielsweise bei der Auswahl eines anderen Herstellers von Sensoren nicht jeder Sensor einzeln angepasst werden, sondern die Aktualisierung der Komponenten geschieht automatisch.

Die Anwendung des Musters ist dann sinnvoll, wenn modellweit viele Abhängigkeiten zwischen eingesetzten Komponenten vorliegen. Dies ist z.B. bei Steuerungskomponenten der Fall, da bei Verwendung einer CPU in den meisten Fällen zwingend auch Buskoppler, Busklemmen usw. vom selben Hersteller verbaut werden müssen.

Der Vorteil des Musters liegt in der zentralen Steuerung von Modellen. Diese erlaubt die modellweite Veränderung von Parameterwerten oder den Austausch ganzer Herstellerbibliotheken. Somit können beispielsweise auch Änderungswünsche des Kunden bezüglich der Verwendung einer bestimmten Herstellerbibliothek kurz vor Produktionsstart noch akzeptiert werden. Der Austausch der Komponenten und die Generierung von neuen Fertigungsunterlagen geschehen automatisch.

Dieses Muster stellt das Gegenteil des Entwurfsmusters Schnittstellenparameter (siehe 8.4.1) dar, bei dem jeder Parameterwert Ebene für Ebene weitervererbt wird. Die Änderbarkeit der Modelle ist jedoch auch beim Muster Zentralsteuerung gegeben, da immer auf den Parameter im Wurzelement zugegriffen wird.

Die oben beschriebenen Verhaltensmuster haben zum Ziel, das Zusammenspiel unterschiedlicher Parameter in verschiedenen Komponenten performant, übersichtlich

und damit auch leicht änderbar zu realisieren. Trotz der Anwendung dieser Muster kann es vorkommen, dass das Regelwerk in Parametern sehr groß und unübersichtlich wird, wenn beispielsweise die möglichen Kombinationen aus Varianten und Optionen durch einen vorhandenen Produktkatalog eingeschränkt werden. In diesem Fall kann es hilfreich sein, das Regelwerk nicht direkt in den Parametern, sondern in einer Tabelle abzubilden. Dieses Vorgehen unterstützt das Entwurfsmuster Entscheidungstabelle.

8.4.4 Entscheidungstabelle

Beim Entwurfsmuster Entscheidungstabelle werden die zulässigen Kombinationen aus Varianten und Optionen eines Komponentenbaukastens als einfache Wenn-Dann-Beziehungen tabellarisch in einer Entscheidungstabelle erfasst. Der Benutzer konfiguriert das Vaterobjekt der Baugruppe, die enthaltenen Komponenten werden dann dynamisch an Hand der Tabelleneinträge hinzugefügt oder entfernt (siehe Bild 8.14). Dies geschieht über Disabler (siehe Kapitel 8.2.3) oder Erweiterungspunkte (siehe Kapitel 8.2.4).

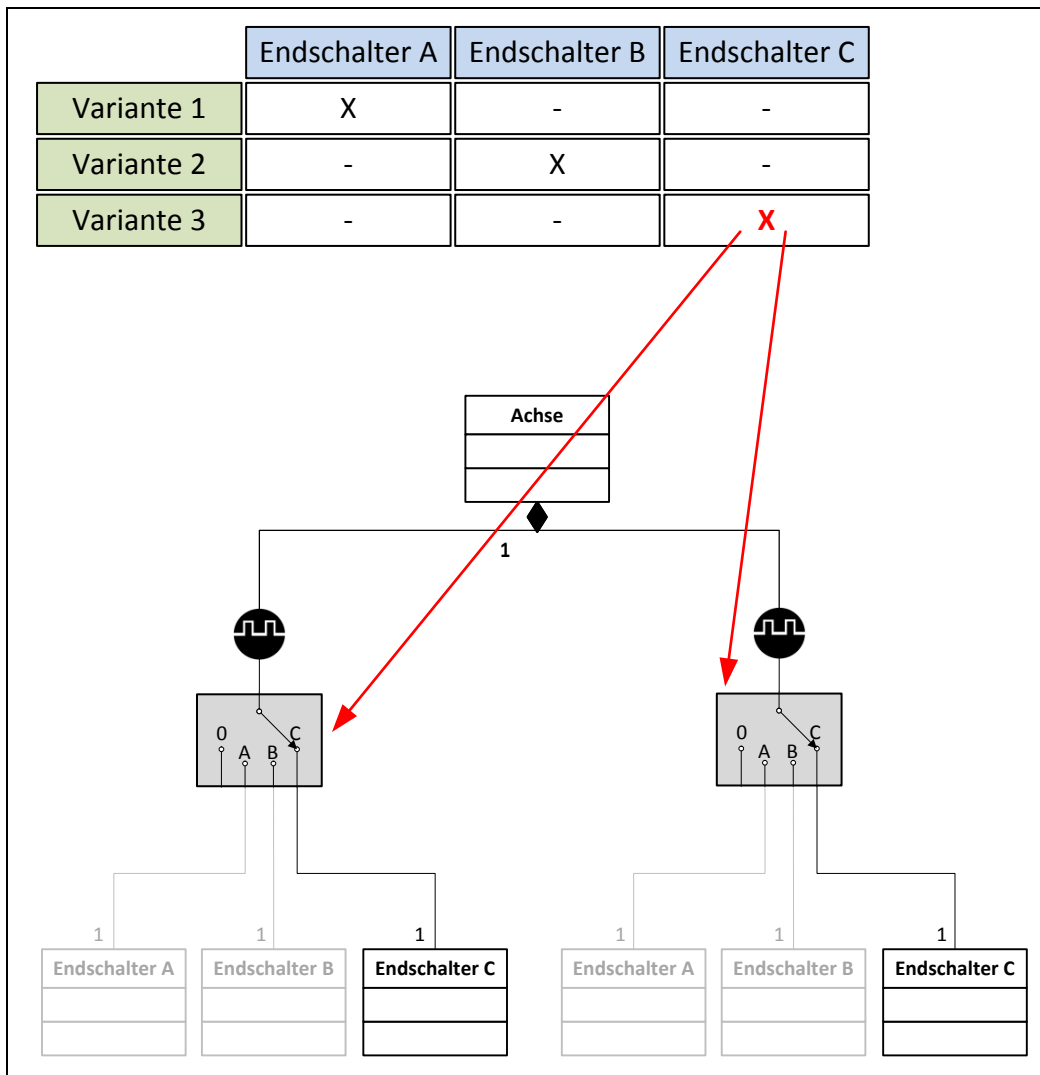


Bild 8.14: Entwurfsmuster Entscheidungstabelle

Ein Hersteller von Linearachsen bietet verschiedene Varianten einer Achse an, die jeweils unterschiedliche Endschalter eingebaut haben. Das Regelwissen, welche Achse welche Endschalter besitzt, wird anstelle langer Fallunterscheidungen übersichtlich in einer Tabelle abgelegt. Nach der Auswahl der gewünschten Variante durch den Kunden werden die passenden Endschalter an Hand der Tabelle ermittelt und in die Achse eingefügt. Die Pflege der Tabelle ist übersichtlich und intuitiv.

Oftmals existieren zahlreiche Varianten und Optionen verschiedener Produkte in einem Baukasten. Das Regelwissen eines solchen Produktspektrums besteht aus zahlreichen, komplexen Wenn-Dann-Beziehungen, um beispielsweise nicht erlaubte Konfigurationen auszuschließen. Dieses Regelwissen hat seinen Ursprung meist in technischen Abhängigkeiten oder Vertriebsstrategien. Wird es in Formeln in den jeweiligen Komponenten abgelegt, so liegt es weit verteilt im Modell und ist somit sehr unübersichtlich und aufwendig zu pflegen. Entscheidungstabellen werden daher von Nutzern für diesen Zweck oftmals bevorzugt.

Das in Kapitel 8.2.2 beschriebene Entwurfsmuster Kopiervorlage kann ebenfalls in Kombination mit dem Muster Entscheidungstabelle realisiert werden. Dadurch werden die erlaubten Kombinationen nicht im Baukasten abgelegt, sondern in eine Entscheidungstabelle „ausgelagert“. Dadurch kann die Übersichtlichkeit und Änderbarkeit des Baukastens entscheidend verbessert werden.

Das Muster Entscheidungstabelle eignet sich somit für die Modellierung von Produkten, deren mögliche Kombinationen von Komponenten durch einfache Wenn-Dann-Abfragen formuliert werden können. Bei komplexeren Berechnungen, die beispielsweise eine Summierung aller Komponenten durchführen, um eine Entscheidung für die Auswahl einer Steuerung o.ä. zu treffen, stoßen die verwendeten Tabellen an ihre Grenzen.

Der Vorteil des Musters liegt in der übersichtlichen Ablage einfachen Regelwissens und der daraus resultierenden leichten Pflege der Daten. Nachteilig wirkt sich aus, dass bei Auftreten komplexerer Berechnungen die Methode gewechselt werden muss. Deshalb sollte vor der Anwendung des Musters eine Analyse stattfinden, um herauszufinden, welche Bereiche des Modells durch Entscheidungstabellen abgebildet werden können und wo aufwendige Formeln und Algorithmen benötigt werden.

Die vier bisher beschriebenen Verhaltensmuster unterstützen den Baukastenentwickler bei der Abbildung von Regelwissen in Parametern der Komponenten. Zusätzlich wird jedoch häufig auch Regelwissen indirekt in der Definition von Konfigurationsoberflächen innerhalb eines FES abgebildet. Auch dieses Regelwissen übersichtlich und damit leicht änderbar abzubilden, ist das Ziel des Entwurfsmusters Memento.

8.4.5 Memento

Beim Entwurfsmuster Memento wird ein Parameterwert einer Komponente über eine Benutzeroberfläche (Konfigurator) vom Benutzer abgefragt. Die Verarbeitung und Speicherung des Werts geschieht über eine Art Zustandsautomaten, bestehend aus mehreren Parametern (siehe Bild 8.15).

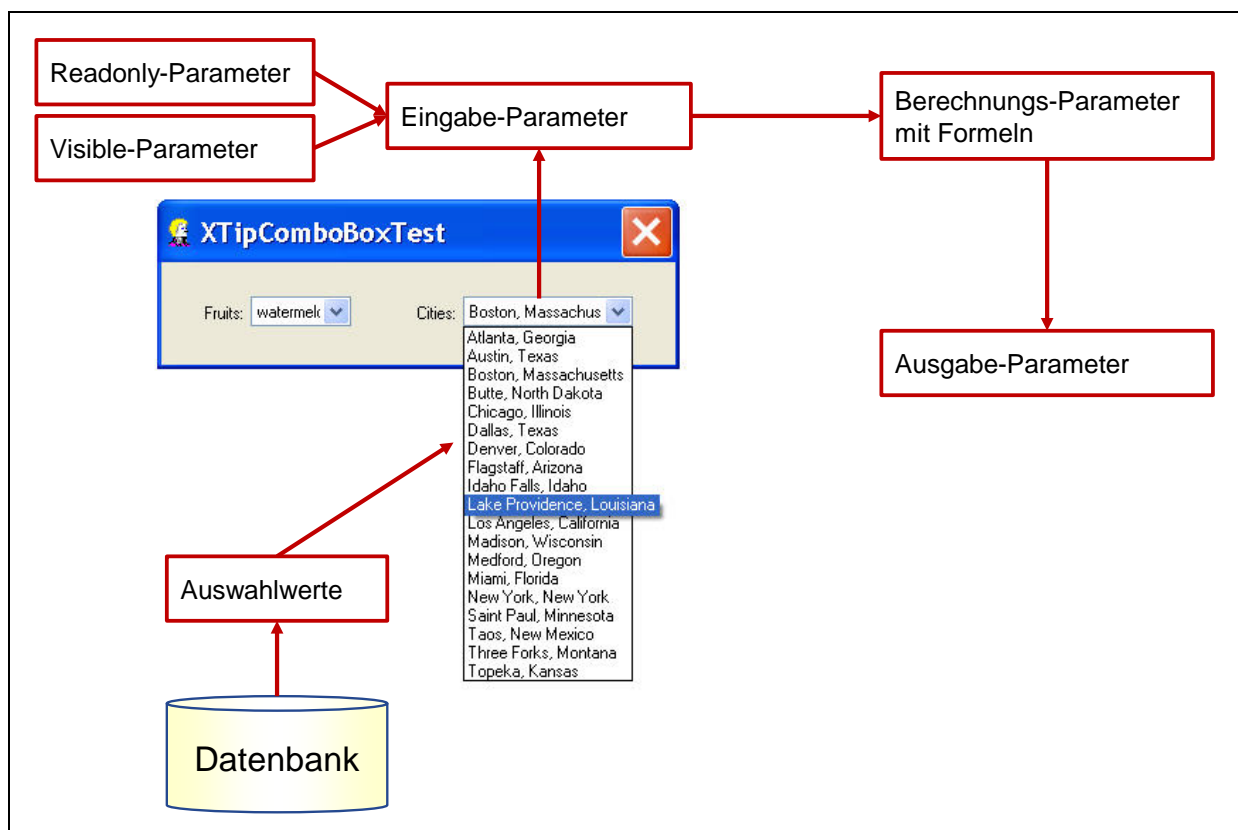


Bild 8.15: Entwurfsmuster Memento, Bsp. Zustandsmaschine für Eingabewerte in Benutzeroberflächen

Ist der Wert eines Parameters einerseits abhängig von der Eingabe des Benutzers, andererseits jedoch auch von den Werten anderer Parameter oder Datenbankabfragen, so ist eine mehrstufige Verarbeitung (Eingabe-Parameter, Berechnungs-Parameter mit Formeln, Ausgabe-Parameter) notwendig. Diese wird auch benötigt, um ein Vor- und Zurückspringen des Benutzers während einer Konfiguration bei Änderung von voneinander abhängigen Werten zu ermöglichen. Macht der Benutzer beispielsweise eine Änderung einer Auswahl rückgängig, so möchte er alle vorher ausgewählten Werte nicht noch einmal eingeben. Auch die Änderbarkeit und Sichtbarkeit des Eingabefeldes soll auf Grund der jeweiligen Parameter des Modells (Readonly-Parameter, Visible-Parameter) gesteuert werden.

In einem Produktkonfigurator für Automobile wählt der Benutzer beispielsweise zunächst eine bestimmte Ausstattungslinie aus. Nach der Auswahl der Motorisierung stellt er jedoch fest, dass es ein bestimmtes Extra nur in Zusammenhang mit einer anderen, im Moment nicht ausgewählten Ausstattungslinie gibt. Deshalb springt er in der Konfiguration zurück auf die erste Seite und wählt die andere Ausstattungslinie aus. Alle anderen, bereits getroffenen Auswahlen wie z.B. Motor oder Farbe sollen jedoch bestehen bleiben. Weiterhin sind Eingabefelder, die auf Grund bereits getroffener Entscheidungen nicht mehr möglich sind, nicht sichtbar (Attribut `visible = false`). Andere Extras sind bereits standardmäßig enthalten, also ist das Eingabefeld ausgegraut (`readonly`).

Die Anwendung dieses Musters ist dann zu empfehlen, wenn Parameterwerte von Benutzeroberflächen übernommen werden sollen und Komfort-Funktionen wie Speicherung von Benutzereingaben bei Änderung von abhängigen Werten o.ä. realisiert werden sollen. Dem Nachteil des höheren Implementierungsaufwands steht der Vorteil einer höheren Flexibilität und Übersichtlichkeit entgegen. Formeln für Berechnungen werden in Parametern implementiert, nicht in der Benutzeroberfläche.

Die in diesem Kapitel vorgestellten Erzeugungs-, Struktur- und Verhaltensmuster werden im Folgenden innerhalb von Konzepten für firmenübergreifende Basisbaukästen und Baukastenstrukturen für Kooperationen zwischen Firmen verwendet.

9 Firmenübergreifende Basisbaukästen

Ein Ansatz zur Reduzierung der Aufwände bei der Erstellung von Baukästen ist die Verwendung von firmenübergreifenden Basisbaukästen. Diese werden zentral entwickelt und gepflegt und können von Baukastenentwicklern aus unterschiedlichen Unternehmen und Branchen genutzt werden. Derartige Basisbaukästen existieren heute bereits in vielen Bereichen (siehe Kapitel 5.3). In FES hat dieser Ansatz bislang jedoch noch nicht in nennenswertem Umfang Anwendung gefunden. Ein wesentlicher Grund hierfür war bislang, dass in vielen Firmen interne Standards für die Erstellung von Daten und Dokumenten existieren, die einem Einsatz von firmenübergreifenden Basisbaukästen entgegenstehen. Diese Problematik kann jedoch durch die Trennung in firmenübergreifende und firmenspezifische (wiederverwendbare) Baukastenebenen gelöst werden (siehe Bild 9.1).

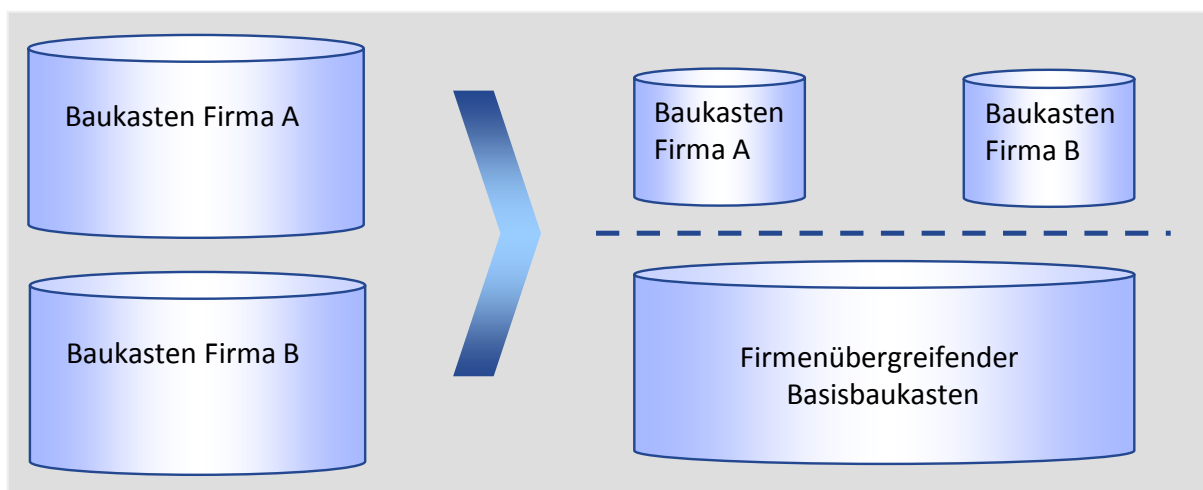


Bild 9.1: Trennung in firmenübergreifende und firmenspezifische Baukastenebenen

9.1 Einbindung firmenübergreifender Basisbaukästen

Der firmenspezifische Teil umfasst die Abbildung von

- Maschinen-/Anlagenarchitektur,
- Know-How-trächtigen technologischen Komponenten und
- disziplinspezifischen Fragmenten für die Komponenten (z.B. Schaltplanmakros, SPS-Funktionsbausteine oder Dokumentationsbausteine).

Der firmenübergreifende Teil enthält beispielsweise

- Zukaufteile (z. B. Steuerungen, Aktoren, Sensoren oder Module) mit zugehörigen Parametersätzen sowie
- disziplinspezifische Fragmente für die Zukaufteile (z.B. Standard-Schaltplanmakros).

Der firmenübergreifende Basisbaukasten wird über definierte Schnittstellen in die firmenspezifischen Baukästen eingebunden (siehe auch Entwurfsmuster Herstellerbibliothek, Kapitel 8.3.3). Firmenspezifische Architekturen und Funktionen werden wie bisher individuell erstellt. Wie die Einbindung der Basisbaukästen bei unterschiedlichen Rahmenbedingungen realisiert werden kann, wird im Folgenden genauer erläutert.

9.1 Einbindung firmenübergreifender Basisbaukästen

Um trotz der Einbindung firmenübergreifender Basisbaukästen Dokumente und Daten weiterhin nach Firmenstandard generieren zu können, sollte der Baukasten in drei Ebenen gegliedert werden. Dies zeigt Bild 9.2 am Beispiel einer SPS-Busklemme.

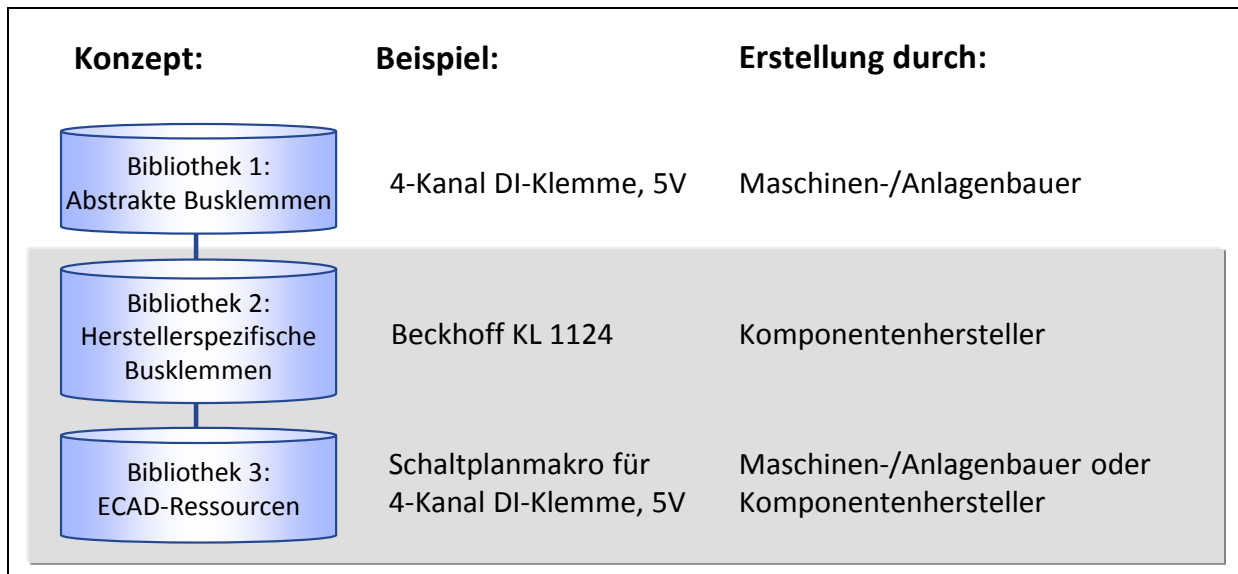


Bild 9.2: Einbindung von firmenübergreifenden Basisbaukästen am Beispiel einer SPS-Busklemme

Im Beispiel enthält Bibliothek 1 abstrakte Busklemmen und wird von jedem Maschinen- oder Anlagenhersteller individuell erstellt. Dadurch ist es beispielsweise möglich, Regeln für die Vergabe einzelner Ein-/Ausgänge nach Firmenstandard einzutragen. Bibliothek 2 beinhaltet konkrete Busklemmen eines Herstellers. Für jede Klemme ist ein Parametersatz implementiert, in den alle für den Einbau und den Betrieb notwendigen Einstellungen und Werte eingetragen werden. Diese Bibliothek ist nicht firmenspezifisch, der Einsatz beim Maschinen-/Anlagenhersteller wird vom Komponentenhersteller fest vorgegeben. Bibliothek 3 hingegen kann je nach Anforderung des Maschinen-/Anlagenbauers firmenspezifisch oder firmenübergreifend sein. Ist im Unternehmensstandard beispielsweise festgelegt, wie ein ECAD-Makro auszusehen hat, so ist die Verwendung einer firmenübergreifenden Bibliothek nicht möglich. Andernfalls kann eine zu Bibliothek 2 passende Bibliothek mit Ressourcen eingebunden werden.

Für die Einbindung der firmenübergreifenden Basisbaukästen in die firmenspezifischen Baukästen sind verschiedene Varianten denkbar, deren Anwendung abhängig

9.1 Einbindung firmenübergreifender Basisbaukästen

von den jeweils einzuhaltenden Unternehmensstandards gewählt werden kann. Drei dieser Varianten werden im Folgenden genauer erläutert.

9.1.1 Einbindung kompletter Basisbaukästen

Kommen nur Komponenten eines einzigen Herstellers zum Einsatz und ist die Verwendung von firmenspezifischen ECAD-Makros nicht vorgegeben, so kann die Einbindung des firmenübergreifenden Basisbaukastens über einfache Erweiterungspunkte ohne Auswahlbedingungen erfolgen. Für jeden Klemmentyp existiert ein fest definiertes Interface, das in den firmenspezifischen Komponenten in Bibliothek 1 implementiert werden muss. Ist dies geschehen, kann das Mapping von abstrakter Klemme zu Beckhoff-Klemme und zur Ressource automatisch stattfinden (siehe Bild 9.3).

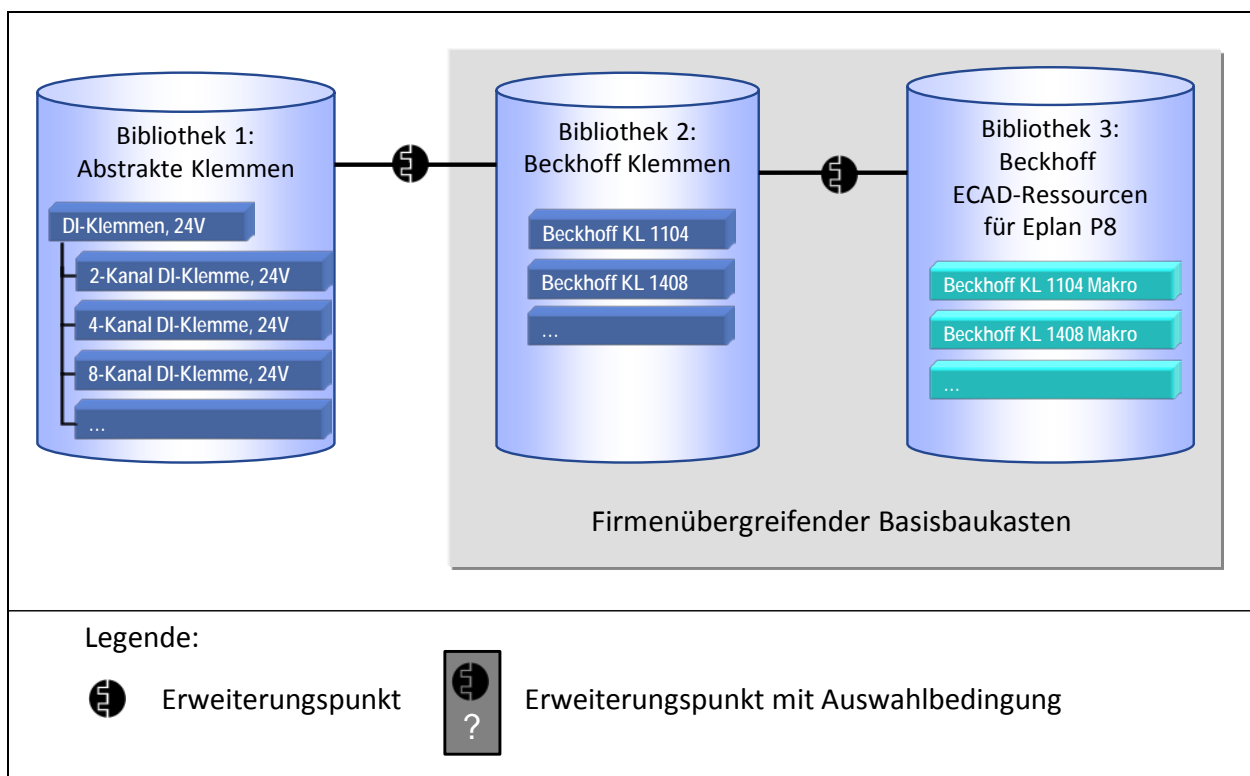


Bild 9.3: Einbindung eines kompletten Basisbaukastens

Der Vorteil bei der Anbindung kompletter Basisbaukästen mit z.B. vorgegebenen Parametern und Makros ist, dass die Aufwände bei der Einbindung in den firmenspezifischen Baukasten minimal sind und beispielsweise keine eigenen Makros erstellt und gepflegt werden müssen. Dieses Konzept ist jedoch auf die Verwendung von Komponenten eines einzelnen Herstellers begrenzt. Sollen Basisbaukästen verschiedener Hersteller verwendet werden, kann das Konzept der Einbindung einzelner Komponenten angewandt werden.

9.1.2 Einbindung einzelner Komponenten

Beim Einsatz von Komponenten unterschiedlicher Hersteller sind alle firmenübergreifenden Basisbaukästen zu importieren. Über Erweiterungspunkte mit Auswahlformel (siehe Kapitel 8.2.5) erfolgt später das Einfügen der zu verwendenden Komponente (siehe Bild 9.4).

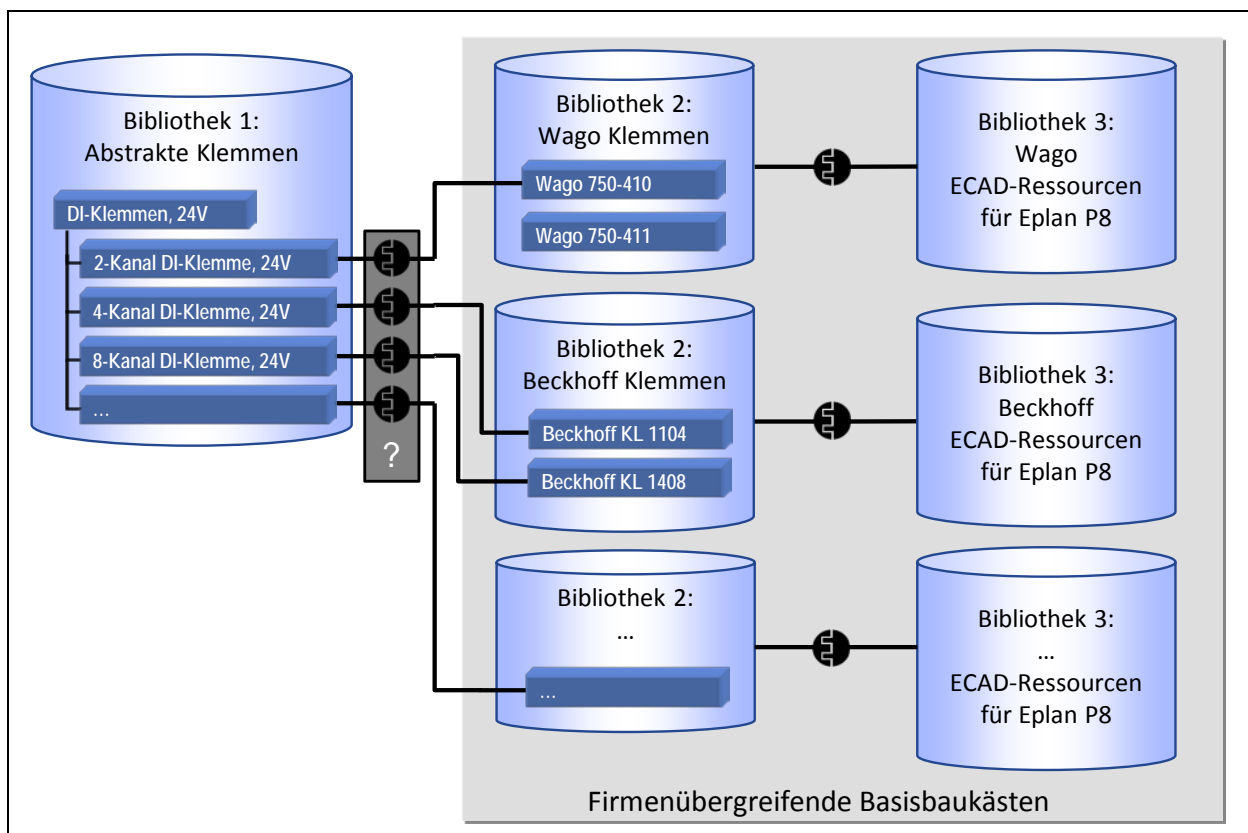


Bild 9.4: Einbindung einzelner Komponenten

Auf diese Weise kann zum Beispiel auftragsspezifisch der Steuerungshersteller gewählt werden, die Klemmenauswahl erfolgt danach automatisch. Die firmenübergreifenden ECAD-Ressourcen sind analog zur Einbindung kompletter Basisbaukästen über Erweiterungspunkte ohne Auswahlformel eingebunden, da wiederum davon ausgegangen wird, dass keine abweichenden Unternehmensstandards für Makros vorliegen.

Sollen jedoch firmenspezifischen Anpassungen erfolgen, weil beispielsweise ein abweichender Firmenstandard existiert, sollte das Konzept der Einbindung eigener Ressourcen benutzt werden.

9.1.3 Einbindung eigener Ressourcen

Analog zum Konzept der Einbindung einzelner Komponenten verwendet der Maschinen-/Anlagenhersteller auch hier Komponenten unterschiedlicher Hersteller. Die ECAD-Makros richten sich jedoch nach einem Unternehmensstandard, der von den Inhalten des firmenübergreifenden Basisbaukastens abweicht. Deshalb erfolgt die Einbindung firmenspezifischer Makros über einen Erweiterungspunkt. An dieser Stelle ist beispielsweise keine Auswahlformel notwendig, da jeweils nur ein Makro mit entsprechendem Interface gemäß einer Klemme aus Bibliothek 2 existiert (siehe Bild 9.5).

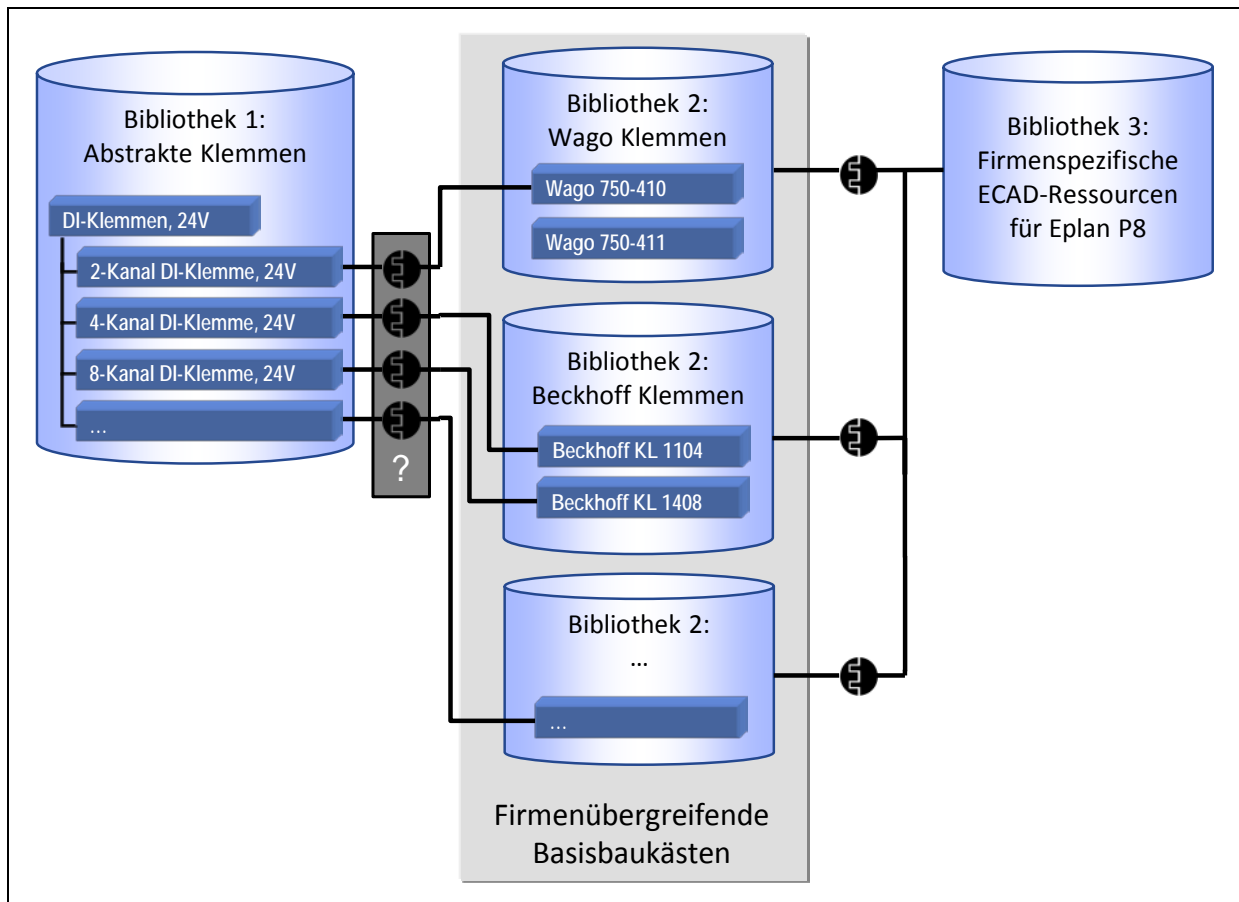


Bild 9.5: Einbindung eigener Ressourcen

Abhängig davon, ob nun Komponenten unterschiedlicher Hersteller verwendet werden sollen und ob ein Firmenstandard existiert, der die Verwendung firmenspezifischer Standard-Ressourcen vorschreibt, kann eines der drei oben beschriebenen Konzepte angewandt werden, um Basisbaukästen einzubinden. Die Realisierung des Konzepts „Einbindung einzelner Komponenten“ wird im Folgenden exemplarisch an Hand der Ergebnisse des Forschungsprojekts AQUIMO aufgezeigt.

9.2 Anwenderberichte

Ein Anwendungsbeispiel für einen firmenübergreifend wiederverwendbaren Basisbaukasten liefert das Forschungsprojekt AQUIMO (Adaptierbares Modellierungs-

werkzeug und Qualifizierungsprogramm für den Aufbau firmenspezifischer mechatronischer Engineering-Prozesse). Dieses wurde von Oktober 2006 bis März 2010 durchgeführt, gefördert vom BMBF (Bundesministerium für Bildung und Forschung) und betreut vom PTKA (Projektträger Forschungszentrum Karlsruhe) [vgl. Angerbauer 2010; Buck III 2010; Buck II 2010; Buck I 2010].

Ziel des AQUIMO-Projekts war es, die Entwurfsphase innerhalb der interdisziplinären Entwicklung eines mechatronischen Produkts durch eine Methode sowie ein Werkzeug zu unterstützen, um so eine frühzeitige, bessere Abstimmung zwischen den einzelnen Disziplinen Mechanik, Elektr(on)ik, Fluidik und Software zu gewährleisten. Durch die Parallelisierung der Entwicklungsaufgabe sind Entwickler aller Disziplinen von Beginn an beteiligt und können so frühzeitig Ideen für Optimierungen einbringen. Gerade die frühen Phasen des Entwicklungsprozesses beeinflussen maßgeblich die späteren Produktkosten, Produktionszeiten sowie die Qualität (siehe Bild 9.6).

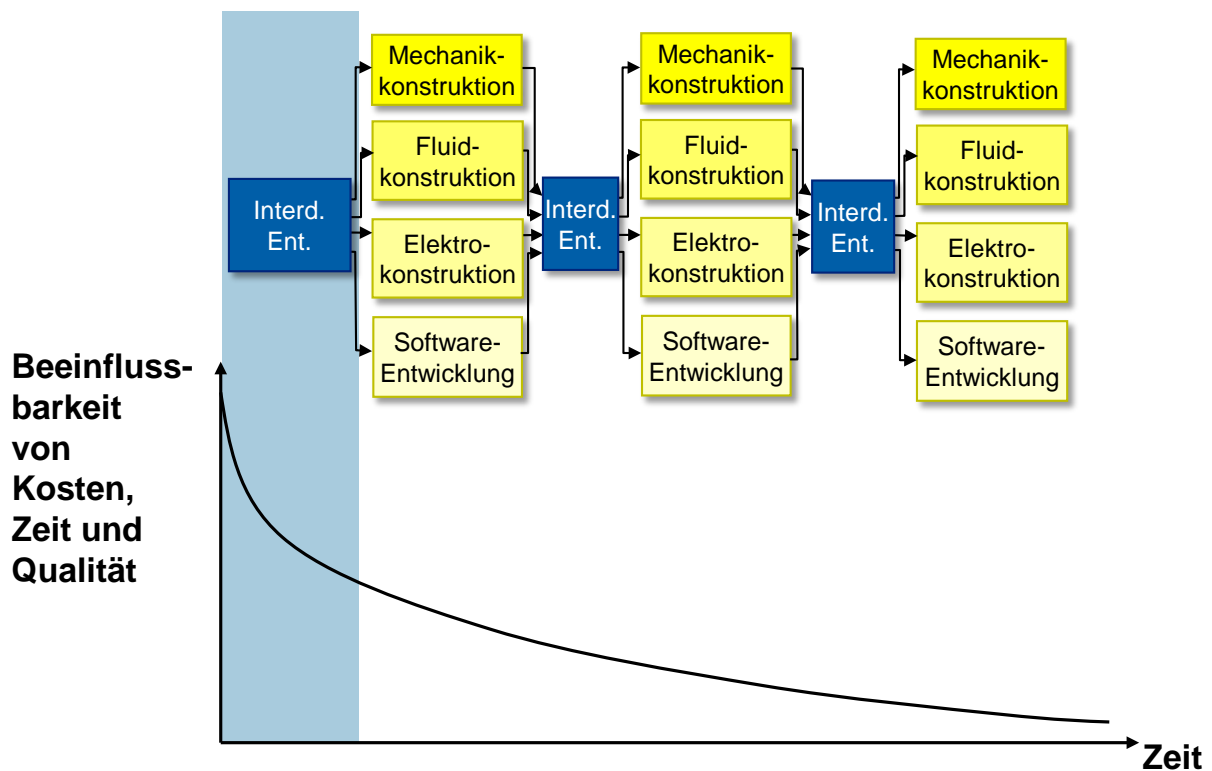


Bild 9.6: Einfluss der Entwicklungsphasen auf Kosten, Zeit und Qualität [vgl. Angerbauer 2010]

Außerdem sollte es mit AQUIMO möglich sein, mehrere Lösungsalternativen und Abläufe einer neuen Maschine oder Anlage durch den Einsatz von Simulationstechnik zu vergleichen und zu bewerten.

Um diese Ziele zu erreichen, wurden während der Projektlaufzeit eine Methode und ein Werkzeug entwickelt, womit auf spielerische Art und Weise mechatronische Entwürfe erstellt werden können. Ein Entwurf besteht dabei aus drei Teilen:

- einem 3D-Modell der neu zu entwickelnden Funktion; dieses wird nach dem Lego-Prinzip aus einfachen Baukastenkomponenten zusammgebaut;
- einem Entwurfsdiagramm, wo den Aktoren Steuerkomponenten (z.B. Ventile oder Antriebsverstärker) zugeordnet werden können;
- einem Ablaufdiagramm, in dem der sequenzielle Ablauf der neuen Funktion festgelegt werden kann.

9.2 Anwenderberichte

Die Beschreibung des Entwurfs findet im Eplan Engineering Center (EEC) statt. Anschließend wird dort auf Knopfdruck automatisch eine Simulation für das System Virtuos [vgl. ISG 2012] generiert, an Hand derer ein Vergleich, eine Bewertung oder die Verifikation des Entwurfs durchgeführt werden kann (siehe Bild 9.7).

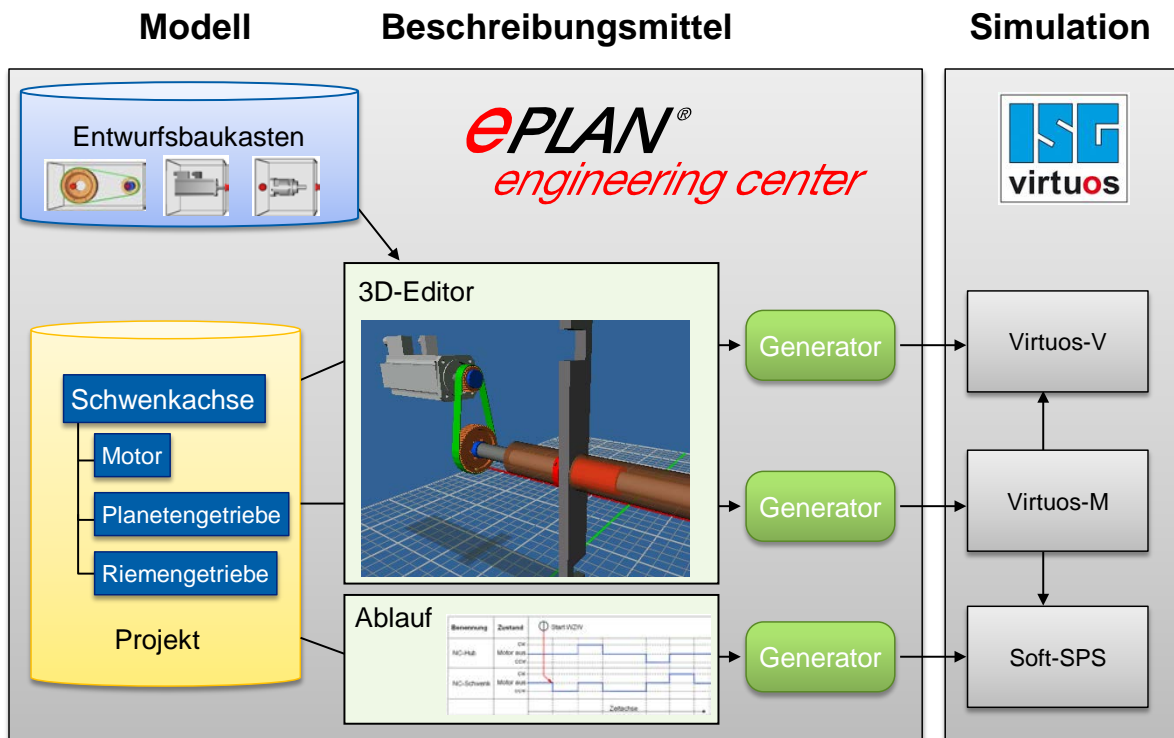


Bild 9.7: Schematische Übersicht über das AQUIMO-Werkzeug [vgl. Angerbauer 2010]

Der für den Entwurf notwendige mechatronische Basisbaukasten wurde nach dem Konzept „Einbindung einzelner Komponenten“ (siehe Kapitel 9.1.2) entwickelt. Trotz des sehr heterogenen Anwenderkonsortiums bestehend aus den Unternehmen Gebr. Heller Maschinenfabrik GmbH (Serienmaschinenbau), HOMAG Holzbearbeitungssysteme GmbH (variantenreiche Serienmaschinen) und NAGEL Maschinen- u. Werkzeugfabrik GmbH (Sondermaschinenbau) entwickelte sich während der Projektlaufzeit von AQUIMO ein firmenübergreifender Baukasten mit Hüllkomponenten, die alle für die jeweilige Funktion erforderlichen Informationen enthalten.

Dabei handelt es sich um elektrische/pneumatische/hydraulische Anschlüsse und mechanische Verbindungspunkte sowie Zustandsgraphen (siehe Bild 9.8).

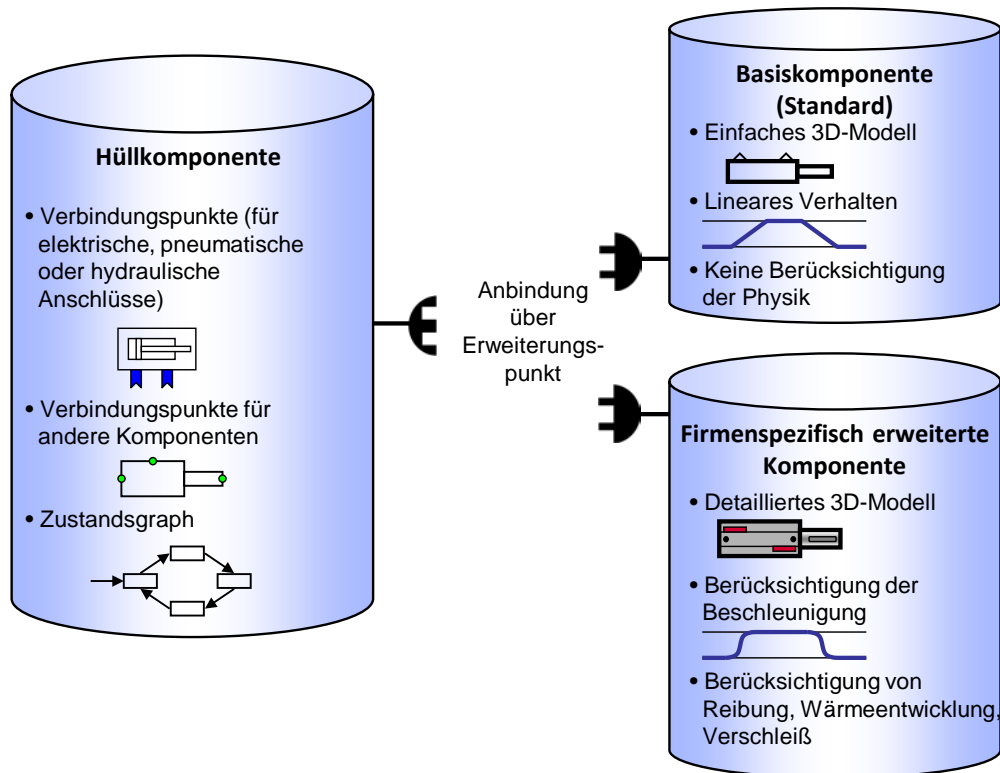


Bild 9.8: Zweistufiger AQUIMO-Baukasten [vgl. Angerbauer 2010]

Für die Generierung von Simulationsmodellen zur Visualisierung von Entwürfen für die Beteiligten aller Disziplinen (Mechanik, Elektr(on)ik, Fluidik, Software) sowie zur virtuellen Inbetriebnahme reichen rudimentäre MCAD-Daten und Verhaltensmodelle aus. Diese sind in einem Baukasten mit Basiskomponenten abgelegt, der standardmäßig über Erweiterungspunkte angebunden wird. Werden genauere Simulationsergebnisse benötigt, erfolgt die Einbindung eigener, detaillierterer Modelle. Diese können für einzelne Komponenten über einen Erweiterungspunkt aus dem firmenspezifischen Baukasten eingefügt werden.

9.3 Fazit

Die Ergebnisse des Forschungsprojekts AQUIMO belegen, dass firmenspezifische und firmenübergreifende Basisbaukästen mit Hilfe von Erweiterungspunkten kombiniert und verwendet werden können. Durch diese Arbeitsweise verringert sich der zeitliche und monetäre Aufwand bei der Erstellung von Baukastensystemen. Ein weiterer Vorteil besteht darin, dass Basisbaukästen bei Bereitstellung durch die Komponentenhersteller zentral gepflegt und Produktneuerungen frühestmöglich eingearbeitet werden können. Durch die Verwendung der vorgestellten Konzepte bleiben die Baukästen von Maschinen- und Anlagenherstellern weiterhin individuell, wobei nur die firmenspezifischen und Know-How-trächtigen Bereiche selbst erstellt und Komponenten wie beispielsweise Zukaufteile aus externen Basisbaukästen eingebunden werden.

Offen bleibt die Frage nach der Erstellung und Pflege der Bibliotheken. Denkbar sind drei Alternativen: direkt durch die Komponentenhersteller (siehe Festo u.a.), durch die Hersteller der Engineering-Tools (siehe Eplan data portal u.a.) oder aber durch Communities (siehe OSADL u.a.). Ebenfalls festzulegen sind das Format, in dem die Daten für die Verwendung in FES abgelegt werden sowie ein Normierungsgremium zur Schaffung einheitlicher Standards [vgl. Buck IV 2010; PLCopen 2012; Drath 2010; Tools 2012].

Die in diesem Kapitel beschriebenen Konzepte erlauben eine firmenübergreifende Nutzung von Basisbaukästen, die beispielsweise von Komponentenherstellern zur Verfügung gestellt werden. Möchten jedoch zwei oder mehr Unternehmen innerhalb des Engineeringprozesses unter Verwendung von FES miteinander kooperieren, so müssen andere Anforderungen wie z.B. der Know-How-Schutz erfüllt werden. Hierfür entstanden im Rahmen dieser Arbeit Konzepte, die im Folgenden genauer erläutert werden.

10 Baukastenstrukturen für Kooperationen zwischen Firmen

Für Kooperationen zwischen Firmen mit Hilfe von FES werden im Folgenden für die in Kapitel 3.3 genannten drei Kooperationsebenen (Baukastenebene, Projektebene, zwischen Baukasten- und Projektebene) Lösungsansätze vorgestellt, die die in Kapitel 4.3 gesammelten Anforderungen erfüllen.

10.1 Kooperation auf Baukastenebene

Bei Kooperationen auf Baukastenebene arbeiten die Partner bei der Erstellung der Baukästen zusammen. Die Projekte werden anschließend von einem oder mehreren der Partner erstellt und daraus alle benötigten Daten und Dokumente generiert (siehe Bild 10.1).

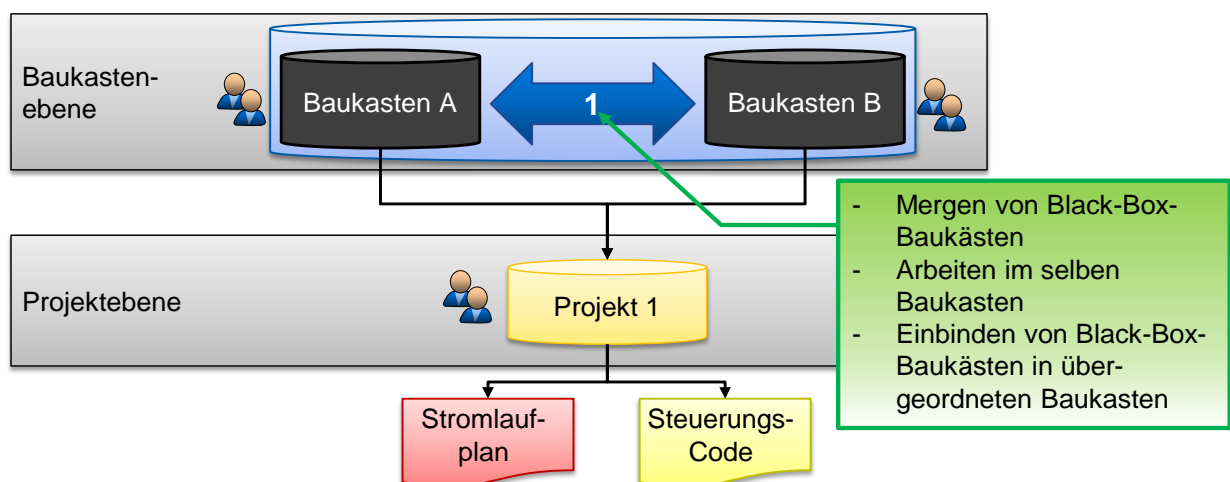


Bild 10.1: Lösungskonzept Kooperation auf Baukastenebene

Für die gemeinsame Erstellung der Baukästen gibt es grundsätzlich drei Möglichkeiten der Zusammenarbeit:

- Jeder der Partner arbeitet unabhängig von den anderen in seinem FES, ausgehend von einem gemeinsamen Stand des Baukastens. Die so entstandenen weiterentwickelten Baukästen werden in regelmäßigen Abständen durch **Mergen** (Mischen) wieder zusammen geführt. Probleme, die entstehen, wenn zwei Personen dieselbe Komponente unterschiedlich abändern, werden als Konflikt dargestellt und manuell beseitigt. Know-How-trächtige Komponenten oder Formeln werden als Black-Box implementiert, so dass sie nur durch den verantwortlichen Entwickler eingesehen werden können. Diese Art der Zusammenarbeit ist offline möglich.
- Als „kontinuierliches Mergen“ kann bezeichnet werden, wenn alle Partner in einem gemeinsamen FES **im selben Baukasten arbeiten**, Änderungen sind dann sofort bei allen Entwicklern sichtbar. Dadurch werden Konflikte, die beim Mergen entstehen können, unwahrscheinlich und werden ggf. sofort sichtbar. Hierzu ist eine Verbindung zur Datenbank notwendig (online). Ansonsten erfolgt die Zusammenarbeit analog zum Mergen.
- Einer der Partner erstellt einen **übergeordneten Baukasten, in den die Baukästen der übrigen Partner als Black-Box-Baukästen eingebunden werden**. Schnittstellen zwischen Baukästen und Komponenten werden vorher genau definiert.

Eine Kooperation auf Baukastenebene mit Hilfe einer der drei beschriebenen Möglichkeiten hat den Vorteil, dass Gesamtdokumente generiert werden können, da aus einem einzigen Projekt heraus generiert wird. Änderungen im Projekt sind ohne Mehraufwände möglich, da diese nur an einer einzigen Stelle eingegeben werden. Weiterhin sind keinerlei Übergabedokumente notwendig. Der Know-How-Schutz wird durch den Einsatz von Black-Box-Baukästen bzw. Komponenten gewährleistet. Dem gegenüber stehen hohe Abstimmungsaufwände bei der Erstellung der Baukästen.

Voraussetzung für diese Art der Kooperation ist, dass bei allen Kooperationspartnern das gleiche FES vorhanden ist. Je nach Realisierung sind automatische Updates und Online- oder Offline-Zusammenarbeit möglich.

Resümee:

Das Lösungskonzept Kooperation auf Baukastenebene eignet sich besonders für enge und über einen längeren Zeitraum laufende Kooperationen. Der Aufwand für die gemeinsame Erstellung der Baukästen ist höher, dafür sind die iterative Bearbeitung von Projekten sowie die konsistente Generierung aller Gesamtdokumente möglich.

10.2 Kooperation auf Projektebene

Bei dieser Art der Kooperation arbeitet jeder Partner in seinem eigenen FES. Lediglich auf Projektebene werden z.B. benötigte Konfigurationsdaten oder Parametrierungen ausgetauscht (siehe Bild 10.2).

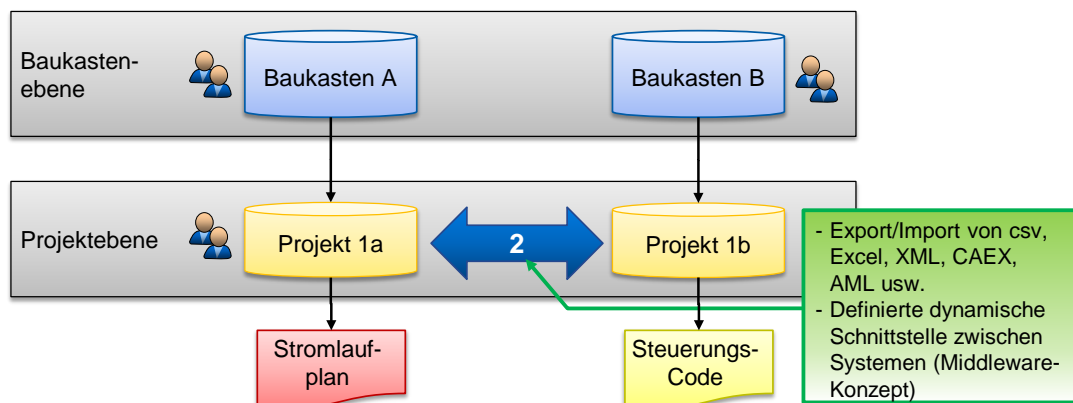


Bild 10.2: Lösungskonzept Kooperation auf Projektebene

Neben der Aufteilung eines Projektes nach Disziplinen (siehe Bild 10.2) ist es ebenso möglich, ein Projekt z.B. in Anlagenteile aufzuspalten und diese getrennt

voneinander zu bearbeiten. Für den Austausch von Daten zwischen Projekten gibt es zwei grundsätzlich unterschiedliche Ansätze:

- Entweder werden Daten in Form von **Übergabeformaten** wie z.B. csv, Excel, XML, CAEX oder AutomationML exportiert und mit Hilfe einer Mapping-Tabelle und ggf. entsprechender Strukturtransformationen automatisch im anderen Projekt importiert,
- oder es existiert eine **definierte dynamische Schnittstelle** zwischen den Systemen (Middleware), die den kontinuierlichen Austausch der Daten übernimmt.

Bei einem solchen automatisierten Austausch von Daten auf Projektebene entstehen geringere Abstimmungsaufwände, da Baukästen und Projekte der Partner völlig autark sind. Auch können bei den Partnern unterschiedliche FES im Einsatz sein. Denkbar ist auch, dass z.B. ein Zulieferer ein und denselben Baukasten bei unterschiedlichen Auftraggebern einsetzt und lediglich für jeden Partner eine individuelle Import-Schnittstelle (Mapping-Tabelle) anlegt.

Bei Änderungen im Projekt entsteht ein erhöhter Datenübergabe- und Kommunikationsaufwand, da die Daten bei allen Partnern noch einmal aktualisiert werden müssen, um konsistente Unterlagen generieren zu können. Ebenso tritt ein erhöhter Aufwand bei einer strukturellen Änderung der Übergabedokumente bzw. der Schnittstelle zu Tage, da die Baukästen aller Partner angepasst werden müssen. Der Know-How-Schutz ist gewährleistet, da lediglich Konfigurationsdaten und Parametersätze, jedoch keine Baukästen weitergegeben werden. Je nach Arbeitsweise sind auch bei diesem Lösungsansatz automatische Updates der Konfigurationsdaten und Online- oder Offline-Zusammenarbeit (Middleware/Übergabeformate) möglich.

Resümee:

Das Lösungskonzept Kooperation auf Projektebene eignet sich besonders für Kooperationen, bei denen geringe Abhängigkeiten zwischen Teilprojekten existieren und nur wenige Iterationen vorkommen.

10.3 Kooperation zwischen Baukasten- und Projektebene

Bei dieser Kooperationsart liegt eine Trennung zwischen Entwicklung und Projektierung vor, d.h. einer der Kooperationspartner erstellt einen Baukasten, der vom anderen Partner bei der Projektierung und zur anschließenden Generierung von Unterlagen verwendet wird (siehe Bild 10.3).

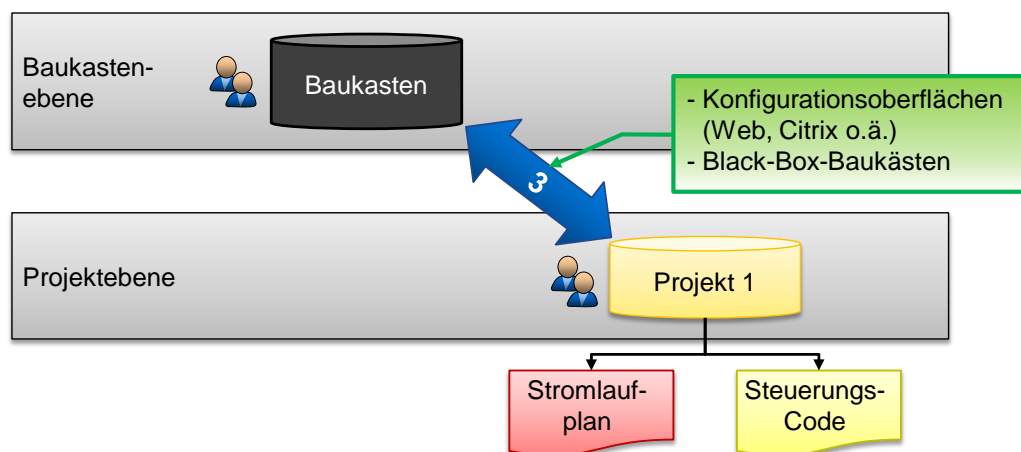


Bild 10.3: Lösungskonzept Kooperation zwischen Baukasten- und Projektebene

Der Baukasten kann auf zwei verschiedene Arten für die Projektierung zur Verfügung gestellt werden:

- Die Projektierung findet direkt im FES des Kooperationspartners statt, der den Baukasten entwickelt. Der Zugriff auf das FES geschieht in Form von **Konfigurationsoberflächen**, die beispielsweise im Web-Browser laufen und per

Citrix o.ä. übertragen werden. Hierfür ist ein Internetzugriff notwendig (online).

- Der **komplette Baukasten wird als Black-Box inkl. des FES** an einen Kooperationspartner zur Projektierung und Generierung aller Unterlagen **übergeben**. Dadurch erfolgt die Projektierung offline.

Wird der Zugriff auf Baukästen in Form von Konfigurationsoberflächen zur Verfügung gestellt, so können Updates automatisch verbreitet werden. In diesem Fall ist die Zusammenarbeit online möglich, ansonsten auch offline durch die Weitergabe von Black-Box-Baukästen. In beiden Fällen ist der Know-How-Schutz gewährleistet. Weiterhin gibt es bei diesem Lösungsansatz keinerlei Übergabedokumente. Änderungen im Projekt sind ohne Mehraufwände möglich, da diese nur an einer einzigen Stelle eingegeben werden. Voraussetzung für den Einsatz dieses Konzepts ist jedoch, dass das gesamte Know-How beim Lieferanten der Baukästen liegt oder an diesen weitergegeben werden muss.

Resümee:

Das Lösungskonzept Kooperation zwischen Entwicklung und Projektierung ist besonders gut geeignet für Kooperationen, bei denen ein Partner das gesamte Know-How besitzt und der andere Partner nach diesen Vorgaben anschließend die Zieldaten erzeugt bzw. die Produktion aufnimmt.

10.4 Übersichtstabelle

Eine Übersicht über die drei Lösungskonzepte zeigt Tabelle 10.1.

Lösungskonzept		Gleiches FES bei OEM und Zulieferer vorhanden	Know-How-Träger	Räumlich getrenntes, zeitversetztes Arbeiten	Paralleles Arbeiten im selben Projekt (automatische Updates)	Automatische Weiterverarbeitung der Übergabedaten	Know-How-Schutz / Sicherheit der Daten	Online-/Offline-Zusammenarbeit
Kooperation auf Baukasten-ebene	Mergen von Black-Box-Baukästen	ja	OEM & Zulieferer	●	● (-)	●	●	Offline
	Arbeiten im selben Baukasten	ja	OEM & Zulieferer	●	●	●	●	Online
	Einbinden von Black-Box-Baukästen in übergeordneten Baukasten	ja	OEM & Zulieferer	●	● (-)	●	●	Online & Offline
Kooperation auf Projektebene	Export/Import von csv, Excel, XML, CAEX, AML usw.	nein	OEM & Zulieferer	●	● (-)	●	●	Offline
	Definierte dynamische Schnittstelle zwischen Systemen (Middleware-Konzept)	nein	OEM & Zulieferer	●	●	●	●	Online
Kooperation zwischen Baukasten- und Projektebene	Konfigurationsoberflächen (Web, Citrix o.ä.)	nein	OEM	●	●	●	●	Online
	Black-Box-Baukästen	ja	OEM	●	● (-)	●	●	Offline

Tabelle 10.1: Vergleich der Lösungskonzepte anhand der Rahmenbedingungen

Der Tabelle ist zu entnehmen, welches Konzept im jeweiligen Anwendungsfall aufgrund der vorliegenden Rahmenbedingungen gewählt werden kann.

10.5 Anwenderberichte

Als Realisierungsbeispiel wird hier der Engineering-Prozess im Anlagenbau der Audi AG genauer erläutert. Dort wird ein Baukasten für Karosserierohbauanlagen erstellt und gepflegt, der für die Projektierung neuer Anlagen gemäß Lösungskonzept „Kooperation zwischen Baukasten- und Projektebene“ an Dienstleister weitergegeben wird. Mit Hilfe des FES Eplan Engineering Center konfigurieren und para-

metrieren die Dienstleister die neue Anlage und generieren anschließend die benötigten Daten und Dokumente wie z.B. Schaltplan oder Steuerungscode [vgl. Buck II 2012].

Bild 10.4 zeigt beispielhaft eine solche Karosserierohbauanlage und die Aufteilung für die Vergabe an unterschiedliche Dienstleister (DL1 – DL4).

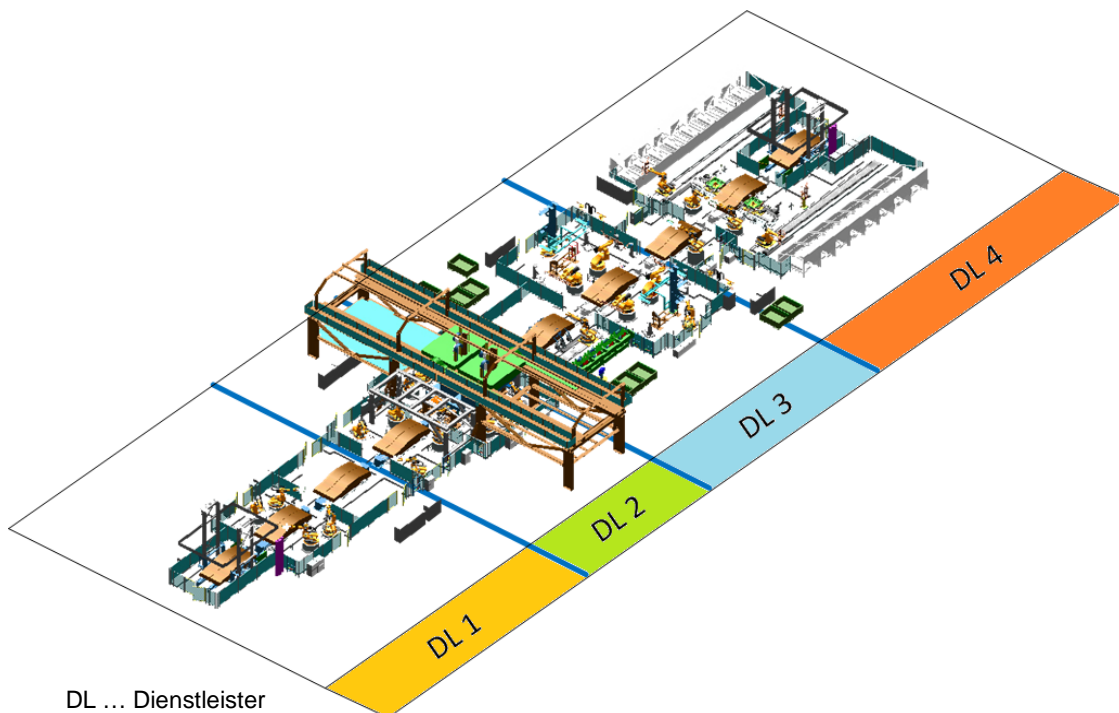


Bild 10.4: Beispiel Engineering einer Karosseriebauanlagen (Quelle: Audi AG)

Die Aufteilung der Anlage in vier Bereiche wurde in diesem Fall entlang des Materialflusses der Rohkarossen vorgenommen. Der Workflow für das Engineering der Anlage ist in Bild 10.5 schematisch skizziert.

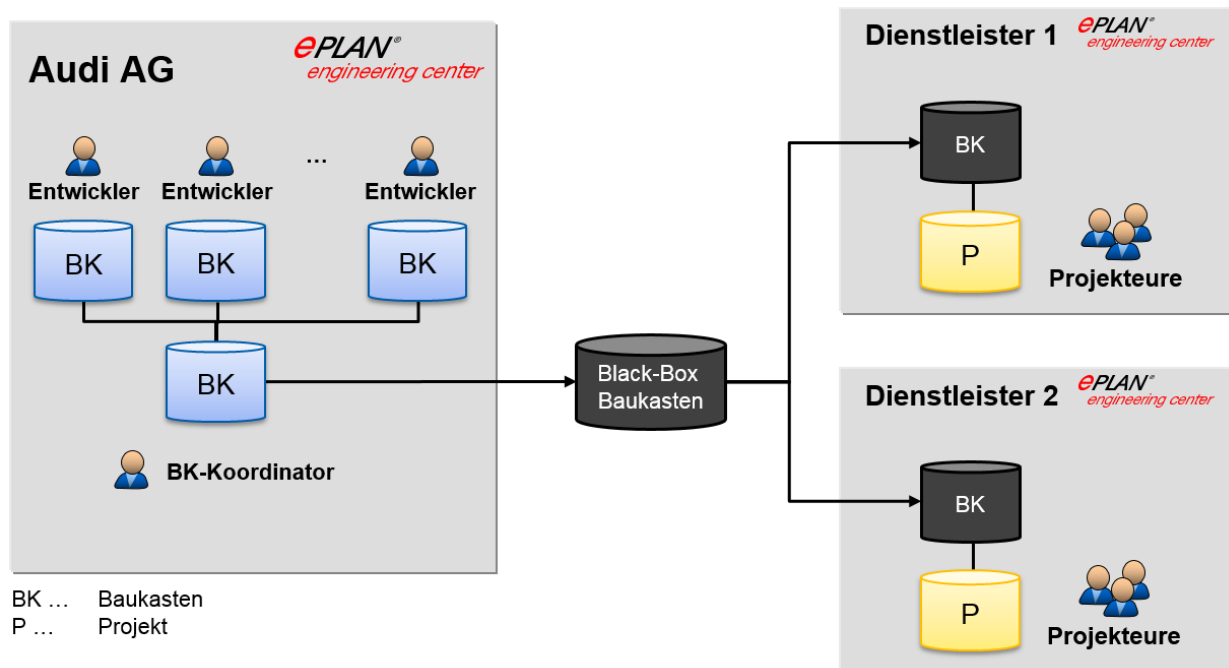
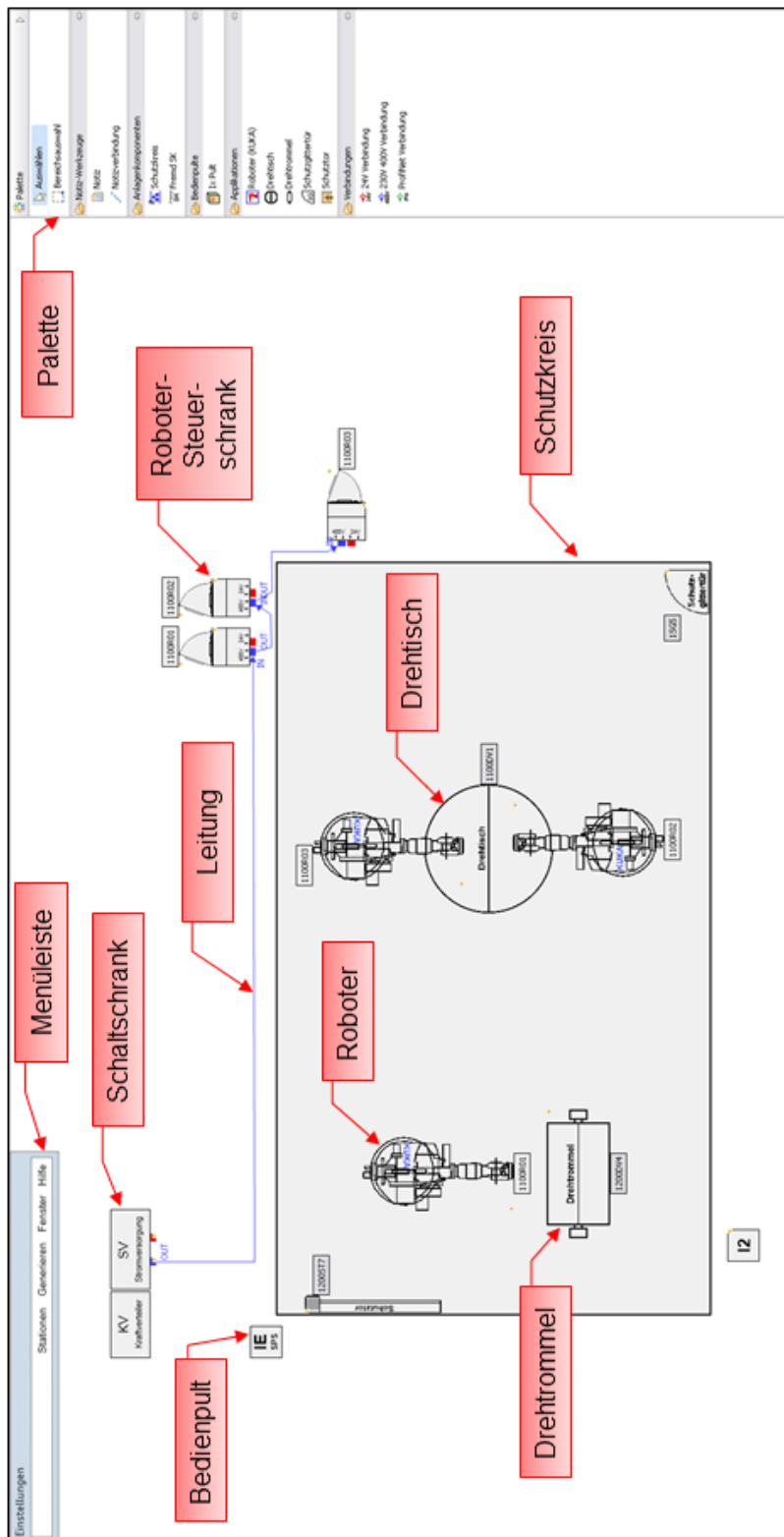


Bild 10.5: Realisierung des Lösungskonzepts „Kooperation zwischen Baukasten- und Projektebene“ bei der Audi AG

Entwickler im Werkzeugbau bei Audi erstellen gemeinsam im Eplan Engineering Center alle benötigten Baukästen. Der Baukasten-Koordinator überwacht die Erstellung, das Mergen der Baukästen zu einem einzigen Baukasten sowie die Freigabe dieses Baukastens. Bei einem neuen Projekt wird der aktuelle Baukasten als Black-Box für die Projektierung der Arbeitspakete an die verschiedenen Dienstleister übergeben. Die Projektoren des Dienstleisters haben ebenfalls das Eplan Engineering Center installiert und können nun gemeinsam mit Hilfe des Baukastens die Projektierung der Anlage vornehmen und anschließend die Zieldokumente generieren.

Eine Konfigurationsoberfläche, die für die Projektierung verwendet wird, ist in Bild 10.6 zu sehen.



(Quelle: Audi AG)

Bild 10.6: Konfigurationsoberfläche für Karosserierohbauanlagen (Quelle: Audi AG)

In einem 2D-Layout werden von den Projektoren z.B. Schutzkreise, Roboter oder Drehtische platziert, indem diese per Drag&Drop aus der Palette (siehe rechts im Bild) in das Layout gezogen werden. Physikalische oder logische Verknüpfungen können anschließend mit Hilfe verschiedenfarbiger Verbindungen vorgenommen werden. Die Konfiguration der einzelnen Objekte wie beispielsweise Roboter oder Bedienpult erfolgt auf individuellen Benutzeroberflächen, die durch Doppelklick auf ein Objekt geöffnet werden.

10.6 Fazit

Am Beispiel der Kooperation der Audi AG mit Engineering-Dienstleistern auf Basis von FES konnte das Lösungskonzept „Kooperation zwischen Baukasten- und Projektebene“ verifiziert werden. Der firmenübergreifende Engineeringprozess konnte deutlich verbessert werden, sowohl die Projektierungszeiten wurden deutlich verkürzt als auch die Qualität der Projektdaten und –Dokumente gesteigert.

11 Zusammenfassung und Ausblick

Unternehmen des Maschinen- und Anlagenbaus bewegen sich heute in einem globalisierten Markt, was zu einem starken Preisdruck führt. Um trotz der hohen Lohnkosten hierzulande hohe Absatzzahlen zu erreichen, setzen viele Unternehmen heutzutage auf Innovation, Energieeffizienz, Qualität und damit hohe Verfügbarkeit und lange Lebensdauer ihrer Produkte. Als weiteren wichtigen Wettbewerbsvorteil hat sich in den letzten Jahren die Individualisierung der Produkte herausgestellt, die es ermöglicht, dem Kunden die für ihn optimale Lösung anzubieten und dadurch ein gutes Preis-Leistungs-Verhältnis zu erreichen. Die Individualisierung von Maschinen und Anlagen führt in den Unternehmen jedoch zu einer erhöhten Komplexität, was sich vor allem im Engineering-Bereich deutlich zeigt, wo explodierende Kosten, Qualitätsprobleme und verlängerte Projektlaufzeiten zu beobachten sind.

Diesen Problemen versuchen viele Unternehmen mit baukastenbasierten Funktionalen Engineering Systemen (FES) entgegenzuwirken. Durch Modularisierung und Wiederverwendung von mechatronischen Komponenten ist es möglich, variantenreiche Maschinen und Anlagen zu konfigurieren und Daten und Projektunterlagen automatisch zu generieren. Die Erstellung der Baukästen für FES ist jedoch zeit- und kostenintensiv. Zudem finden viele Engineering-Prozesse nicht mehr zentral in einem Unternehmen statt, sondern werden firmenübergreifend in Kooperationen hoch spezialisierter Zulieferer und Dienstleister abgewickelt.

Für die reproduzierbare Erstellung leicht zu wartender Baukästen wurde deshalb in dieser Arbeit ein Katalog aus mechatronischen Entwurfsmustern erarbeitet. Diese wurden zum einen benutzt, um Konzepte für firmenübergreifend wiederverwendbare Basisbaukästen zu entwickeln, womit sich Aufwände bei der Erstellung von

Baukästen reduzieren lassen. Zum anderen wurden mit Hilfe dieser Entwurfsmuster Konzepte erstellt, um den Einsatz von FES in Kooperationen zwischen Firmen zu ermöglichen.

Zusammenfassend wurden im Rahmen dieser Arbeit im Wesentlichen folgende vier Maßnahmen erarbeitet, um den Einsatz von FES zu erleichtern und zu optimieren:

- Um Optimierungspotenziale in vorhandenen Baukästen transparent zu machen entstand ein „**Model-Analyser**“, der in Modellen innerhalb des FES E-plan Engineering Center eingesetzt werden kann.
- Das in der Softwaretechnik verbreitete Prinzip der Entwurfsmuster wurde auf den Bereich der FES übertragen und ein **Katalog aus 15 verschiedenen mechatronischen Entwurfsmustern** entwickelt.
- Mit Hilfe dieser neu entwickelten Entwurfsmuster wurden drei unterschiedliche Konzepte ausgearbeitet, die den Einsatz **firmenübergreifend wiederverwendbarer Basisbaukästen** ermöglichen.
- Für den **Einsatz von FES in firmenübergreifenden Kooperationen** entstanden Konzepte zur Unterstützung durchgängiger Engineering-Prozesse für drei unterschiedliche Kooperationsformen. Die Konzepte basieren ebenfalls auf den entwickelten mechatronischen Entwurfsmustern.

Der in dieser Arbeit entwickelte Model-Analyser wurde bei der Optimierung eines Kundenmodells aus dem Bereich Anlagenbau eingesetzt. Die mechatronischen Entwurfsmuster sowie je ein Konzept für Basisbaukästen und für Kooperationen konnten am Forschungsprojekt AQUIMO sowie beim Einsatz im Werkzeugbau der Audi AG getestet und verifiziert werden.

Um dem Baukastenentwickler den Einsatz der in dieser Arbeit vorgestellten mechatronischen Entwurfsmuster zu erleichtern, wäre es denkbar, diese direkt innerhalb eines FES zu implementieren. Aus der Modellierung eines Produktspektrums mit

Hilfe bekannter Modellierungsmethoden wie z.B. MindMaps könnte man dann einen FES-Baukasten samt Konfigurationsoberfläche automatisch ableiten, ohne Expertenkenntnisse im Bereich der Modellierung von FES-Baukästen haben zu müssen. Dabei könnten im Hintergrund die vorgestellten Entwurfsmuster angewandt werden, um performante und übersichtliche Modelle mit einem hohen Grad an Wiederverwendung zu erhalten.

Des Weiteren sind für die Realisierung und Verbreitung firmenübergreifend wiederverwendbarer Basisbaukästen noch herstellerunabhängige Standards und Datenformate notwendig. Hierfür bedarf es entsprechender Gremien, die diese Standards ausarbeiten und im Markt verbreiten.

Abbildungsverzeichnis

Bild 2.1:	Überblick FES – Baukasten – Entwurfsmuster	3
Bild 2.2:	Sequenzielles Engineering [angelehnt an Litto 2004]	4
Bild 2.3:	Erstellen von neuen Projekten durch Kopieren und Anpassen [vgl. Litto 2004]	5
Bild 2.4:	Funktionale Engineering Systeme (FES).....	7
Bild 2.5:	Abgrenzung von Baukasten-Begriffen [angelehnt an Lewek 2005, S. 21]	8
Bild 3.1:	Problemstellung bei der Entwicklung, Nutzung und Pflege von Baukästen für FES	11
Bild 3.2:	Unterschiedliche Modellierungen von Servomotoren	13
Bild 3.3:	Beispiel eines gut (oben) bzw. schlecht (unten) zu wartenden Baukastens	15
Bild 3.4:	Ebenen der Wiederverwendung von Komponenten	17
Bild 3.5:	Manuelle Datenübergabe zwischen Automobilhersteller und Zulieferer.....	18
Bild 3.6:	Aufwände für Kommunikation und Datenzusammenführung bei Kooperationen.....	20
Bild 3.7:	Kooperationsarten	22
Bild 5.1:	Objektorientierte Entwurfsmetriken nach Chidamber und Kemerer [angelehnt an Sneed 2010].....	28
Bild 5.2:	Kooperationen auf Baukastenebene (1), auf Projektebene (2) sowie zwischen Baukasten- und Projektebene (3).....	33
Bild 5.3:	Übersicht über das AutomationML Datenformat [vgl. Drath 2010].....	36
Bild 7.1:	Beispiel einer Projektübersicht innerhalb eines FES.....	49
Bild 7.2:	Screenshot Ergebnis der Analyse eines Modells	59
Bild 8.1:	Entwurfsmuster Komposition	64
Bild 8.2:	Entwurfsmuster Kopiervorlage	66
Bild 8.3:	Entwurfsmuster Disabler	68
Bild 8.4:	Entwurfsmuster Erweiterungspunkt	70
Bild 8.5:	Entwurfsmuster Erweiterungspunkt mit Auswahlformel	71
Bild 8.6:	Entwurfsmuster Erweiterungspunkt mit Auswahlformel und Anzahl	74

Bild 8.7:	Entwurfsmuster Trennung Architektur – Baukasten am Beispiel Fräszentrum	76
Bild 8.8:	Entwurfsmuster 4-Ebenen-Komponentenmodell	78
Bild 8.9:	Entwurfsmuster Herstellerbibliothek.....	81
Bild 8.10:	Entwurfsmuster Parameter-Bibliothek	84
Bild 8.11:	Entwurfsmuster Schnittstellenparameter	86
Bild 8.12:	Entwurfsmuster Dynamische Parametervererbung	88
Bild 8.13:	Entwurfsmuster Zentralsteuerung.....	90
Bild 8.14:	Entwurfsmuster Entscheidungstabelle.....	93
Bild 8.15:	Entwurfsmuster Memento, Bsp. Zustandsmaschine für Eingabewerte in Benutzeroberflächen.....	95
Bild 9.1:	Trennung in firmenübergreifende und firmenspezifische Baukastenebenen.....	99
Bild 9.2:	Einbindung von firmenübergreifenden Basisbaukästen am Beispiel einer SPS-Busklemme	101
Bild 9.3:	Einbindung eines kompletten Basisbaukastens	102
Bild 9.4:	Einbindung einzelner Komponenten	103
Bild 9.5:	Einbindung eigener Ressourcen.....	105
Bild 9.6:	Einfluss der Entwicklungsphasen auf Kosten, Zeit und Qualität [vgl. Angerbauer 2010].....	107
Bild 9.7:	Schematische Übersicht über das AQUIMO-Werkzeug [vgl. Angerbauer 2010].....	108
Bild 9.8:	Zweistufiger AQUIMO-Baukasten [vgl. Angerbauer 2010].....	109
Bild 10.1:	Lösungskonzept Kooperation auf Baukastenebene.....	111
Bild 10.2:	Lösungskonzept Kooperation auf Projektebene	113
Bild 10.3:	Lösungskonzept Kooperation zwischen Baukasten- und Projektebene.....	115
Bild 10.4:	Beispiel Engineering einer Karosseriebauanlagen (Quelle: Audi AG)	118
Bild 10.5:	Realisierung des Lösungskonzepts „Kooperation zwischen Baukasten- und Projektebene“ bei der Audi AG.....	119
Bild 10.6:	Konfigurationsoberfläche für Karosserierohbauanlagen (Quelle: Audi AG)	120

Tabellenverzeichnis

Tabelle 5.1:	Übersicht über die Anforderungen und den Stand der Technik	39
Tabelle 7.1:	Kategorisierung der Metriken	48
Tabelle 7.2:	Bedeutung der Farben für die Grenzwerte der Metriken.....	54
Tabelle 7.3:	Grenzwerte der Metriken, Teil 1: Absolutwert.....	55
Tabelle 7.4:	Grenzwerte der Metriken, Teil 2: Maximum.....	56
Tabelle 7.5:	Grenzwerte der Metriken, Teil 3: Median	57
Tabelle 8.1:	Übersicht über die Entwurfsmuster	62
Tabelle 10.1:	Vergleich der Lösungskonzepte anhand der Rahmenbedingungen.....	117

Literaturverzeichnis

- Alexander 1995 Alexander, Christopher; Ishikawa, Sara; Silverstein, Murray:
Eine Muster-Sprache: Städte - Gebäude - Konstruktion.
Wien: Löcker, 1995
- Angerbauer 2010 Angerbauer, Ronald; Buck, Raphael u.a.:
AQUIMO - Ein Leitfaden für Maschinen und Anlagenbauer.
Frankfurt am Main: VDMA Verlag, 2010
- Balzert 1998 Balzert, Helmut:
*Lehrbuch der Software-Technik: Software-Management,
Software-Qualitätssicherung, Unternehmensmodellierung.* 1.
Aufl.
Heidelberg; Berlin: Spektrum Akademischer Verlag, 1998
- Balzert 2001 Balzert, Helmut:
Lehrbuch der Software-Technik: Software-Entwicklung. 2.
Aufl.
Heidelberg; Berlin: Spektrum Akademischer Verlag, 2001
- Balzert 2011 Balzert, Helmut:
*Lehrbuch der Softwaretechnik - Entwurf, Implementierung,
Installation und Betrieb.* 3. Aufl.
Heidelberg: Spektrum Akademischer Verlag, 2011
- Biggerstaff 1989 Biggerstaff, Ted J.:
Software Reusability - Concepts and Models.
New York: ACM Press, 1989

- Borowski 1961 Borowski, Karl-Heinz:
Das Baukastensystem in der Technik.
Berlin; Göttingen; Heidelberg: Springer, 1961
- Bracht 2013 Bracht, Uwe; Sontag, Till:
Smarte Fabrikplanung.
In: wt Werkstattstechnik online 103 (2013), Nr. 4, S. 356-362
- Buck I 2010 Buck, Raphael:
Aquimo: Der Entwurfsbaukasten.
In: SPS Magazin 23 (2010), November, S. 74-77
- Buck II 2010 Buck, Raphael:
*AQUIMO: Mechatronische Zusammenarbeit verbessert
Maschinen und Anlagen.*
In: intelligenter produzieren (2010), Nr. 2, S. 44-45
- Buck III 2010 Buck, Raphael:
Kompetenz aus dem Kasten.
In: MONTAGE technik (2010), Nr. 4, S. 26-28
- Buck IV 2010 Buck, Raphael; Litto, Marco; Verl, Alexander:
*Firmenübergreifend einsetzbare und adaptierbare
Baukastenarchitekturen für das mechatronische Engineering.*
In: Bender, Klaus; Schumacher, Walter; Verl, Alexander
(Hrsg.): Tagungsband SPS/IPC/DRIVES 2010, 23.-25.
November 2010, Nürnberg.
Berlin; Offenbach: VDE Verlag, 2010, S. 455-463

- Buck I 2012 Buck, Raphael; Litto, Marco:
Durchgängige Auftragsbearbeitungsprozesse für Maschinen- und Anlagen vom Vertrieb bis zur Inbetriebnahme.
In: Tagungsband Fertigungstechnisches Kolloquium (FTK),
25.-26. September 2012, Stuttgart.
Stuttgart: Gesellschaft für Fertigungstechnik, 2012, S. 205-226
- Buck II 2012 Buck, Raphael; Lewek, Jörg; Verl, Alexander:
Funktionales Engineering: Unterstützung firmenübergreifender Engineeringprozesse.
In: Tagungsband Internationales Forum Mechatronik 2012,
21.-22. November 2012, Mayrhofen, Österreich.
Innsbruck: Standortagentur Tirol, 2012, S. 95-114
- Drath 2010 Drath, Rainer:
Datenaustausch in der Anlagenplanung mit AutomationML - Integration von CAEX, PLCopen XML und COLLADA.
Berlin; Heidelberg: Springer, 2010
- Ebert 1996 Ebert, Christof; Dumke, Reiner:
Software-Metriken in der Praxis: Einführung und Anwendung von Software-Metriken in der industriellen Praxis.
Berlin; Heidelberg: Springer, 1996
- Eclipse 2012 Eclipse: Eclipse Foundation, 2012.
<http://www.eclipse.org/> (19. Juni 2012)
- Eplan 2012 Eplan: EPLAN Software & Service, 2012.
<http://www.eplan.de/produkte/mechatronik/eplan-engineering-center/> (8. August 2012)

- Eric 2008 Eric, Freeman; Elisabeth, Freeman u.a.:
Entwurfsmuster von Kopf bis Fuß.
Köln: O'Reilly, 2008
- Eversheim 2003 Eversheim, Walter:
Innovationsmanagement für technische Produkte.
Berlin; Heidelberg: Springer, 2003
- Fedai 2006 Fedai, Murat:
*Regelbasierter, semantischer Datenaustausch zwischen
Anlagenplanungssystemen auf Basis der XML-Technologie.*
Düsseldorf: VDI Verlag, 2006
Zugl. Aachen, Techn. Hochsch., Diss., 2006
- Franke 2002 Franke, Hans-Joachim; Hesselbach, Jürgen u.a.:
*Variantenmanagement in der Einzel- und
Kleinserienfertigung.*
München; Wien: Hanser, 2002
- Gamma 2004 Gamma, Erich; Helm, Richard u.a.:
*Entwurfsmuster - Elemente wiederverwendbarer
objektorientierter Software.*
München: Addison-Wesley, 2004
- Gamma 1992 Gamma, Erich:
*Objektorientierte Software-Entwicklung am Beispiel von
ET++.*
Berlin; Heidelberg: Springer, 1992
Zugl. Zürich, Univ., Diss., 1992

- Gausemeier 2009 Gausemeier, Jürgen; Plass, Christoph; Wenzelmann, Christoph:
Zukunftsorientierte Unternehmensgestaltung - Strategien, Geschäftsprozesse und IT-Systeme für die Produktion von morgen.
München; Wien: Hanser, 2009
- Georgii 2009 Georgii, Hans-Otto:
Stochastik: Einführung in die Wahrscheinlichkeitstheorie und Statistik.
Berlin: De Gruyter, 2009
- Grechenig 2010 Grechenig, Thomas; Bernhart, Mario u.a.:
Softwaretechnik - Mit Fallbeispielen aus realen Entwicklungsprojekten.
München: Pearson Studium, 2010
- Hackel 2011 Hackel, Monika:
Auf dem Weg zum interdisziplinären mechatronischen Konstruktionsprozess.
Frankfurt am Main: Lang, 2011
Zugl. Hagen, Fernuniv., Diss., 2010
- Hillside 2012 Hillside: The Hillside Group, 2012.
<http://hillside.net/patterns> (27. Juni 2012)
- ISG 2012 ISG: ISG Industrielle Steuerungstechnik GmbH, 2012.
<http://www.isg-stuttgart.de/virtuos.html> (15. Januar 2012)
- IBM 2012 IBM: Jazz, 2012.
<https://jazz.net/> (16. Juni 2012)

- Lewek 2005 Lewek, Jörg:
*Adaptierbares Informationssystem zur Erstellung
baukastenbasierter Fertigungseinrichtungen.*
Heimsheim: Jost-Jetter, 2005
Zugl. Stuttgart, Univ., Diss., 2005
- Litto 2004 Litto, M.; Koradja, I. u.a.:
*Baukastenbasiertes Engineering mit Föederal - Ein Leitfaden
für Maschinen- und Anlagenbauer.*
Frankfurt am Main: VDMA Verlag, 2004
Zugl. Stuttgart, Univ., Diss., 2004
- Neyrinck 2013 Neyrinck, Adrian; Kaupp, Markus; Verl, Alexander:
*Fähigkeitenbasierte Variantenerzeugung für den simulativen
Anlagenvergleich.*
In: wt Werkstattstechnik online 103 (2013), Nr. 5, S. 444-448
- Nyhuis 2013 Nyhuis, Peter:
*Herausforderung Komplexitätserhöhung erfolgreich
bewältigen.*
In: wt Werkstattstechnik online 103 (2013), Nr. 4, S. 263
- PLCopen 2012 PLCopen: PLCopen, 2012.
<http://www.plcopen.org/> (6. Februar 2012)
- Reuse 2008 Reuse, Bernd; Vollmar, Roland:
Informatikforschung in Deutschland.
Berlin; Heidelberg; New York: Springer, 2008
- Schuh 2005 Schuh, Günther:
Produktkomplexität managen: Strategien - Methoden - Tools.
München: Hanser, 2005

- Schmelzer 2008 Schmelzer, Hermann Josef; Sesselmann, Wolfgang:
Geschäftsprozessmanagement in der Praxis.
München: Hanser, 2008
- Siemens 2012 Siemens: Siemens Teamcenter, 2012.
http://www.plm.automation.siemens.com/de_de/products/teamcenter/community-collaboration/index.shtml (17. August 2012)
- Sneed 2010 Sneed, Harry M.; Seidl, Richard; Baumgartner, Manfred:
Software in Zahlen - Die Vermessung von Applikationen.
München: Hanser, 2010
- Tools 2012 Tools: Step Tools, 2012.
<http://steptools.com/library/standard/> (18. Februar 2012)
- Tomasz 2010 Tomasz, Gonsior:
Standardisierung vs. Differenzierung - Beschaffungsorientierte Betrachtung der Modularisierung entlang der Wertschöpfungskette.
Berlin: epubli, 2010
Zugl. Köln, Univ., Diss., 2008
- Verl 2009 Verl, Alexander; Müller, Verena; Haubelt, Andreas:
Baukastenbasiertes simulationsgestütztes Engineering.
In: A&D-Kompendium 2009/2010.
München: publish-industry Verlag, 2009, S. 72-74

- Voss 2012 Voss, geb. Müller, Verena:
Wiederverwendbare Simulationsmodelle für die domänen- und disziplinübergreifende Produktentwicklung.
Heimsheim: Jost-Jetter, 2012
Zugl. Stuttgart, Univ., Diss., 2012
- Westkämper 2009 Westkämper, Engelbert; Zahn, Erich:
Wandlungsfähige Produktionsunternehmen - Das Stuttgarter Unternehmensmodell.
Berlin; Heidelberg: Springer, 2009
- Weyrich 2012 Weyrich, Michael; Klein, Philipp:
Modulbasiertes Engineering von Produktionsanlagen.
In: wt Werkstattstechnik online 102 (2012), Nr. 9, S. 592-597

Lebenslauf

<i>Persönliches</i>	Name	Raphael Buck
	Geburtstag	2. Februar 1983
	Geburtsort	Riedlingen
<i>Schulbildung</i>	1989-1993	Grundschule Altheim
	1993-2002	Kreisgymnasium Riedlingen
<i>Zivildienst</i>	2002-2003	Kreisklinik Riedlingen
<i>Studium</i>	2003-2008	Universität Stuttgart Automatisierungstechnik in der Produktion, Hauptfächer Steuerungstechnik und Fabrikbetrieb
<i>Berufstätigkeit</i>	2008-2009	SMS Siemag AG, Düsseldorf (ehemals SMS Demag AG)
	2009-2013	Institut für Steuerungstechnik der Werkzeug- maschinen und Fertigungseinrichtungen, Universität Stuttgart
	2009-2013	Eplan Software & Service GmbH & Co. KG, Stuttgart (ehemals Mind8 GmbH & Co. KG)
	Seit 2013	SEW Eurodrive GmbH & Co. KG, Bruchsal

Die Arbeit stellt einen Katalog aus mechatronischen Entwurfsmustern für die reproduzierbare Erstellung leicht zu wartender Baukästen für Funktionale Engineering Systeme (FES) vor. Um Optimierungspotenziale in vorhandenen Baukästen transparent und die Anwendung der entwickelten Entwurfsmuster messbar zu machen, werden Metriken zur Analyse von FES-Baukästen ermittelt. Die mechatronischen Entwurfsmuster werden zum einen benutzt, um Konzepte für firmenübergreifend wiederverwendbare Basisbaukästen zu entwickeln, womit sich Aufwände bei der Erstellung von Baukästen reduzieren lassen. Zum anderen werden mit Hilfe dieser Entwurfsmuster Konzepte erstellt, um den Einsatz von FES in Kooperationen zwischen Firmen zu ermöglichen.

ISBN 978-3-8396-0828-9



FRAUNHOFER VERLAG