

Hefreihe des Instituts für Bauingenieurwesen
Book Series of the Department of Civil Engineering
Technische Universität Berlin

Herausgeber:

Editors:

Prof. Dr.-Ing. Matthias Barjenbruch
Prof. Dr.-Ing. Karsten Geißler
Prof. Dr.-Ing. Reinhard Hinkelmann
Prof. Dr.-Ing. Wolfgang Huhnt
Prof. Dr.-Ing. Bernd Kochendörfer
Prof. Dr.-Ing. Yuri Petryna
Prof. Dr.-Ing. Stavros Savidis
Prof. Dr. sc. techn. Mike Schlaich
Prof. Dr.-Ing. Volker Schmid
Prof. Dr.-Ing. Frank U. Vogdt

Shaker Verlag

Aachen 2010

Muster in Prozessen der Bauablaufplanung

Ein Branch-and-Bound-Verfahren zur
Mustererkennung in Planungs- und
Ausführungsprozessen

vorgelegt von

Dipl.-Ing. Felix Archibald Enge
aus Berlin

an der Fakultät VI - Planen Bauen Umwelt
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
– Dr.-Ing. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr.-Ing. Bernd Kochendörfer

Berichter: Prof. Dr.-Ing. Wolfgang Huhnt

Berichter: Prof. Dr.-Ing. Stefan M. Holzer

Tag der wissenschaftlichen Aussprache: 16. November 2009

Berlin 2010

D83

Heftreihe des Instituts für Bauingenieurwesen
Book Series of the Department of Civil Engineering
Technische Universität Berlin

Band 6

Felix Enge

Muster in Prozessen der Bauablaufplanung

Ein Branch-and-Bound-Verfahren zur Mustererkennung
in Planungs- und Ausführungsprozessen

D 83 (Diss. TU Berlin)

Shaker Verlag
Aachen 2010

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Zugl.: Berlin, Techn. Univ., Diss., 2009

Muster in Prozessen der Bauablaufplanung

Dissertationsschrift von Felix Enge
Fakultät VI – Planen, Bauen, Umwelt
der Technischen Universität Berlin

Vorsitzender: Prof. Dr.-Ing. Bernd Kochendörfer
Berichter: Prof. Dr.-Ing. Wolfgang Huhnt
Berichter: Prof. Dr.-Ing. Stefan M. Holzer

Tag der wissenschaftlichen Aussprache: 16.11.2009

Copyright Shaker Verlag 2010

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8322-9053-5

ISSN 1868-8357

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen
Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9
Internet: www.shaker.de • E-Mail: info@shaker.de

Für Johanna

„Wer seine Zukunft formen will,
muss in der Gegenwart leben ...“

Antoine de Saint-Exupéry

Vorwort

Die vorliegende Arbeit ist während meiner Tätigkeit als Wissenschaftlicher Mitarbeiter am Fachgebiet *Internetbasierte Prozessmodellierung für ad-hoc Organisationen im Bauwesen* des Instituts für Bauingenieurwesen der Technischen Universität Berlin unter der Leitung von Herrn Prof. Dr.-Ing. Wolfgang Huhnt entstanden. Die Motivation zu der behandelten Thematik *Muster in Prozessen der Bauablaufplanung* entstammt meiner wissenschaftlichen Arbeit im Rahmen der Forschungsschwerpunkte des Fachgebiets auf dem Gebiet der Planungs- und Ausführungsprozesse.

Herrn Prof. Dr.-Ing. Wolfgang Huhnt danke ich sehr herzlich für die engagierte Betreuung meiner Arbeit. Sein unentwegter Eifer, seine Begeisterung und seine ständige Bereitschaft zu fachlichen Diskussionen haben mir einen Rahmen geboten, der maßgeblich zum Erfolg dieser Arbeit beigetragen hat. Bedanken möchte ich mich auch bei Herrn Prof. Dr.-Ing. Stefan Holzer für die Bereitschaft als Zweitgutachter ein unabhängiges Urteil abzugeben sowie bei Herrn Prof. Dr.-Ing. Bernd Kochendörfer für den Vorsitz des Promotionsausschusses.

Als Wissenschaftlicher Mitarbeiter hatte ich im Rahmen meiner Tätigkeit die Gelegenheit, Diplomanden zu betreuen. Hervorheben möchte ich die Betreuung von Herrn Dipl.-Ing. Sven Richter. Die von ihm behandelten Inhalte haben wesentlich zu der Konkretisierung meiner eigenen Arbeit beigetragen. Vor diesem Hintergrund markiert dieser Zeitraum den Beginn meiner Promotion und darüber hinaus aber auch den Beginn einer nachhaltig guten Zusammenarbeit bis ins aktuelle Projekt hinein.

Hervorheben möchte ich auch das außergewöhnliche Engagement von Herrn cand.-Ing. Lukas Olbrich. Trotz einer nicht unerheblichen eigenen Arbeitslast hat Herr Olbrich meine Arbeit kontinuierlich und beharrlich verfolgt und sie mit großer Akribie korrekturgelesen. Seine Anregungen und Empfehlungen haben fachlich wie formal Spuren hinterlassen, die ich sehr schätze.

Nicht unerwähnt bleiben sollen die Bemühungen von Heidrun Kämpf und Jana Drescher, die mich dabei unterstützt haben, über die fachlichen

Anforderungen hinaus, auch den linguistischen gerecht zu werden.

Mein ganz besonderer Dank gilt letztlich meiner Frau Johanna für ihre fortwährende liebevolle Unterstützung, Ihr Verständnis, ihre endlose Geduld sowie ihre unermüdliche Nachsicht haben mir den notwendigen familiären Rückhalt zur Durchführung dieser Arbeit gegeben.

Berlin, im März 2010

Felix Enge

Abstract

Within the project preparation phase, experienced professionals manually map design information onto process information with the aim to develop realistic and practical schedules. Unfortunately, the mapping itself is neither part of any underlying data model nor is it supported by current scheduling tools. As a consequence the process of setting up the data model for a schedule is still not supported formally.

Huhnt und Enge [2006]¹ described a modelling technique that addresses the missing linkage between design and process information. The approach makes use of so called component types. These are template sub-processes that describe the fabrication procedure of typical building components. Decomposing the building into components and assigning a component type to each component allows for formal support while scheduling. Depending on the decomposition of the building into components and the complexity of the involved sub-processes the specification effort differs. The question about optimal sub-processes arises: Which layout of building components and sub-processes results in minimal specification effort?

This dissertation presents a branch and bound algorithm to determine optimal decompositions of planning and construction processes into design information and process information. For a given schedule, which has been modelled on a template-based modelling technique, all possible decompositions are determined. During the decomposition process the encountered configurations are compared. Those with minimal specification effort are registered. The knowledge about optimal decompositions is used to improve the basis of process-templates for future projects. Theoretical and practical examples are examined and discussed.

¹ [Huhnt und Enge, 2006, Can algorithms support the specification of construction schedules?]

Zusammenfassung

Im Rahmen der Vorbereitung von Bauprojekten befassen sich erfahrene Ingenieure mit der Planung von komplexen Bauabläufen, mit dem Ziel, Vorgaben für einen kosten- und zeitoptimalen Sollablauf zu bestimmen. Hierbei sind Entwurfsinformationen, die beispielsweise aus CAD-Plänen stammen, auf Prozessinformationen abzubilden. Software-Werkzeuge erlauben die graphische Bearbeitung, die Spezifikation von Abhängigkeiten und die Überprüfung von Eigenschaften des zugrunde liegenden Graphen. Die Logik der Eingabe des modellierten Ablaufs wird jedoch nicht adressiert.

Huhnt und Enge [2006]² beschreiben eine Methodik, auf deren Grundlage eine Verknüpfung der Entwurfsinformationen mit Prozessinformationen vorgenommen werden kann. Die Methodik nutzt Prozessvorlagen, die als Bauteiltypen bezeichnet werden. Sie stellen den Herstellungsablauf typischer Bauteile dar. Zerlegt man ein Bauwerk in seine Bauteile und ordnet diesen die Bauteiltypen zu, so lässt sich das Gestalten der Bauabläufe formalisieren. Abhängig von der Art der Zerlegung - Zerlegung des Bauwerks in Bauteile - und der Komplexität der verwendeten Bauteiltypen - Prozessvorlagen - ergibt sich ein unterschiedlich hoher Spezifikationsaufwand für den Nutzer. Vor dem Hintergrund des Spezifikationsaufwands stellt sich die Frage nach der optimalen Zerlegung eines Bauwerks in Bauteile und Bauteiltypen.

Im Rahmen dieser Arbeit wird auf ein Branch-and-Bound-Verfahren eingegangen, das zur Bestimmung optimaler Zerlegungen von Planungs- und Ausführungsprozessen in Entwurfskomponenten und korrespondierende Teilprozesse dient. Durch Rückrechnung werden für gegebene Bauabläufe, die auf der Grundlage der oben genannten Modellierungstechnik entwickelt wurden, diejenigen Zerlegungen gesucht, für die der Spezifikationsaufwand minimal ist. Aus der Kenntnis der optimalen Zerlegungen lassen sich Prozessvorlagen kontinuierlich verbessern. Erfahrungen aus abgeschlossenen Projekten tragen dazu bei, Prozessvorlagen für anstehende Projekte bereitzustellen. An theoretischen und praktischen Beispielen wird die Methodik erläutert und diskutiert.

² [Huhnt und Enge, 2006, Can algorithms support the specification of construction schedules?]

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	2
1.1.1	Zustandsbasierte Modellierung von Bauabläufen . . .	3
1.1.2	Spezifikationsaufwand	4
1.1.3	Fragestellung	5
1.1.4	Zielsetzung	6
1.2	Herangehensweise	6
1.2.1	Muster in Bauabläufen	7
1.2.2	Optimierungsproblem	7
1.2.3	Branch-and-Bound-Verfahren	8
1.3	Gliederung	8
1.3.1	Bauablaufplanung	8
1.3.2	Mustererkennung in Bauabläufen	9
2	Bauablaufplanung	11
2.1	Begriffe und Abgrenzungen	11
2.1.1	Projekt	11
2.1.2	Prozess	12
2.1.3	Prozessorientierter Ansatz	14
2.1.4	Prozessmodell	14
2.2	Randbedingungen der Bauablaufplanung	14
2.2.1	Abgrenzung zur Produktionsplanung	15
2.2.2	Sichtweisen unterschiedlicher Beteiligter	16
2.2.3	Planungsphasen und Detaillierungsgrad	17
2.2.4	Arten von Abhängigkeiten	20
2.3	Methoden und Verfahren der Bauablaufplanung	20
2.3.1	Terminieren von Vorgängen	21
2.3.2	Netzplantechnik	21
2.3.3	Ansätze aus Forschung und Entwicklung	23
2.4	Zusammenfassung	28

3	Prozessorientiertes Modell für die Bauablaufplanung	29
3.1	Mathematischer Hintergrund	30
3.2	Prozesse: Modellierungsmethodik	31
3.3	Bauablaufplanung: Terminologie und Datenmodell	34
3.3.1	Projektunabhängige und -spezifische Informationen	34
3.3.2	Strukturierung von Informationen	36
3.3.3	eERM Leistungsgegenstand und -vorlage	38
3.3.4	eERM Leistungsprozess und -vorlage	39
3.3.5	eERM Vorgang und -vorlage	42
3.3.6	eERM Vorgangsposition und -vorlage	42
3.3.7	eERM Zustand, Ereignis und -vorlage	43
3.3.8	eEPK Funktion und Ereignis	44
3.3.9	Ereignisgraph	47
3.4	Bauablaufplanung: Modellierung	49
3.4.1	Modellieren projektunabhängiger Informationen	50
3.4.2	Modellieren projektspezifischer Informationen	54
3.4.3	Spezifikation von Voraussetzungen	58
3.4.4	Spezifikationsaufwand	60
3.5	Diskussion	63
4	Muster in Bauabläufen	65
4.1	Mustererkennung	66
4.2	Abstraktion der Problemstellung	69
4.2.1	Der Prozess als Ereignisgraph	69
4.2.2	Zerlegung eines Ereignisgraphen	71
4.2.3	Prozessmodell ausgedrückt als Zerlegung	73
4.2.4	Identifikation von Ereignisgraphen	73
4.2.5	Muster von Ereignisgraphen	75
4.2.6	Klassen von Komponenten	76
4.2.7	Bewertung eines Ereignisgraphen	77
4.2.8	Bewertung einer Zerlegung	77
4.2.9	Normierte Bewertung einer Zerlegung	78
4.2.10	Bewertung eines Prozessmodells	79
4.3	Optimale Zerlegung	80
4.3.1	Optimierungsproblem	80
4.3.2	Branch-and-Bound	82
4.3.3	Minimale Komponenten	101
4.3.4	Entwickeln der Komponenten	106
4.3.5	Überschneidungsfreie Komponentenkombinationen	114
4.3.6	Bewertung von Komponentenkombinationen	118
4.3.7	Algorithmus	120

4.3.8	Verlauf des Algorithmus	124
5	Auswertung exemplarischer Beispiele	129
5.1	Beispiel 1: Herstellung von Wänden	130
5.2	Mustererkennung für das Beispiel	136
5.2.1	Initialisieren	136
5.2.2	Verlauf des Verfahrens	138
5.2.3	Ergebnis der Suche	142
5.3	Interpretation der Muster	145
5.3.1	Gestalt der Komponenten einer Zerlegung	146
5.3.2	Von der Komponente zur Leistungsprozessvorlage	148
5.4	Beispiel 2: Herstellung von Wänden	149
5.5	Aufwandsbetrachtungen	153
6	Zusammenfassung	157
6.1	Modellierungsmethodik	157
6.2	Mustererkennung	158
7	Ausblick	161
7.1	Methodik zur Auswertung von optimalen Zerlegungen	161
7.2	Zusammenwirken mit modellbasierten Ansätzen	162
7.3	Mustererkennung in Bauteilzusammenhängen	163
	Literaturverzeichnis	165
	Abbildungsverzeichnis	171
	Symbolverzeichnis	173
	Abkürzungsverzeichnis	177
A	Begriffe aus der Graphentheorie	179
A.1	Mengen	179
A.2	Unäre Relationen	182
A.3	Binäre Relationen	183
A.4	Graphen	190
A.5	Wege und Zyklen	193

Kapitel 1

Einleitung

Im Bauwesen sind an der Entwicklung, der Planung, der Konstruktion, der Herstellung, dem Betrieb, dem Umbau und dem Rückbau rund um ein Bauwerk eine Vielzahl an Personen beteiligt. Diese gehören verschiedenen Organisationen an, die aus unterschiedlichen Disziplinen stammen und entsprechend unterschiedliche Sichtweisen haben. Für jedes Projekt ist die Zusammensetzung der Beteiligten eine andere. Unterschiedliche Verfahren kommen zum Einsatz, verfügbare Produktionsmittel ändern sich, und die Art der Interaktion der Beteiligten variiert. Arbeitsabläufe lassen sich nicht ohne weiteres von einem Projekt auf ein anderes übertragen. Dies sind nur einige Aspekte, die ausdrücken, dass Bauprojekte einzigartig sind und individuell vorbereitet werden müssen. Dies betrifft insbesondere die Erarbeitung von Bauabläufen im Rahmen der Ablaufplanung.

Abhängig von der Rolle eines Planers und seiner Sichtweise auf das Projekt, dient die Ablaufplanung als Grundlage u.a. für die Dauern-, Termin-, Kapazitäts-, Bereitstellungs- und Kostenplanung. Ziel der Ablaufplanung ist es, unter Berücksichtigung aller Randbedingungen, konsistente und korrekte Vorgaben für einen kosten- und zeitoptimalen Sollablauf eines Bauprojekts zu erstellen.

Eine klassische Herangehensweise zur Dauern-, Termin-, Kapazitäts-, Bereitstellungs- und Kostenplanung ist die Anwendung der Netzplantechnik, bei der der Bauablauf auf einen Graphen abgebildet wird. Der Vorteil der Netzplantechnik liegt in dem formalen Ansatz zur Bestimmung der Zeitfenster. Lokale Änderungen im zugrunde liegenden Graphen wirken sich durch erneute Vorwärts- und Rückwärtsrechnung korrekt auf den gesamten Ablauf aus. Die Anwendung dieser Methodik setzt allerdings voraus, dass aus der Ablaufplanung alle Anordnungsbeziehungen zwischen den Vorgängen richtig und vollständig ermittelt sind.

Software-Werkzeuge stehen zur Verfügung, die den bearbeitenden Ingenieur bei seiner Tätigkeit unterstützen. Sie erlauben die graphische Bearbeitung, die Spezifikation von Abhängigkeiten und die Überprüfung von Eigenschaften des zugrunde liegenden Graphen. Die Logik der Eingabe des modellierten Ablaufs wird jedoch nicht adressiert. Eine Aussage darüber, ob alle logischen Abhängigkeiten berücksichtigt sind, können nur aus der Anschauung heraus beantwortet werden. Die Überprüfung der Abhängigkeiten auf logische Korrektheit und Vollständigkeit ist mit einem enormen Aufwand verbunden. Gründe, die zu der Spezifikation von Anordnungsbeziehungen geführt haben, werden nicht explizit modelliert. Nur die Ersteller einer Ablaufplanung halten dieses Wissen als Gedankenmodell vor. Für ein Anpassen, Ändern oder unabhängiges Überprüfen durch Dritte ist es erforderlich, dieses Gedankenmodell aus dem Kontext heraus zu rekonstruieren. Dies ist aufwendig und fehleranfällig.

Das übergeordnete Ziel dieser Arbeit ist es, das Gestalten komplexer Bauabläufe in immer kürzeren Planungszeiten durch formale Methoden zu unterstützen, um konsistente und korrekte Ablaufmodelle für die Netzplantechnik zur Verfügung zu stellen. Im Fokus steht die Reduzierung des Spezifikationsaufwands und die Bewältigung der Komplexität von Bauabläufen.

1.1 Motivation

Die Ermittlung der Abhängigkeiten zwischen Vorgängen in komplexen Bauabläufen stellt eine Herausforderung dar. In der Praxis erfolgt diese Tätigkeit aus der Anschauung heraus. Auch mit großem Aufwand lässt sich nicht sicherstellen, dass alle Abhängigkeiten für einen reibungslosen Ablauf erfasst sind. In vielfältigen Forschungsvorhaben wurde die Problematik aufgegriffen, den Aufwand bei dem Erstellen von Ablaufplänen zu reduzieren und gleichzeitig die logische Korrektheit und Vollständigkeit der Ergebnisse zu erhöhen.

Huhnt und Lawrence [2004]¹ beschreiben eine Methodik zur Gestaltung von Planungsabläufen, die das Ziel verfolgt, mit einem Minimum an Eingangsinformationen auszukommen und formale Mechanismen zur Berechnung des Zusammenhangs anzuwenden. Im Fokus steht die Reduzierung des Spezifikationsaufwands und die Bewältigung der Komplexität von Planungsabläufen. In dem Beitrag von Huhnt [2005]² und Huhnt

¹ [Huhnt und Lawrence, 2004, Methodik zur Modellierung konsistenter Soll-Vorgaben für komplexe Planungsprozesse]

² [Huhnt, 2005, Generating Sequences of Construction Tasks]

und Enge [2006]³ wird das Konzept auf Ausführungsabläufe übertragen.

Als Teil dieser Arbeit werden die oben genannten Ansätze aufgegriffen und in einer allgemeingültigen Modellierungsmethodik zusammengeführt. Der Methodik liegt eine zustandsbasierte Betrachtung⁴ zugrunde, die es erlaubt, aus einer Produktbeschreibung auf Abläufe zu schließen.

1.1.1 Zustandsbasierte Modellierung von Bauabläufen

Im Rahmen eines Projekts werden Leistungen erbracht, z.B. *Statik prüfen* als Teilleistung eines Planungsablaufs oder *Stahlbetonwand einschalen* als Teilleistung eines Ausführungsablaufs. Jeder Leistung - *Statik prüfen* bzw. *Stahlbetonwand einschalen* - liegt ein Leistungsgegenstand⁵ - *Statik* bzw. *Stahlbetonwand* - zugrunde. Eine Leistung beschreibt eine Menge an Vorgängen, die einen Leistungsgegenstand von einem Zustand in einen anderen transformieren. Das Resultat einer Transformation lässt sich durch Zustandsbeschreibungen ausdrücken. Fasst man die Zustände, die sich auf denselben Leistungsgegenstand beziehen, zu einer Menge zusammen und ordnet diese Menge mit Hilfe einer Ordnungsrelation, so erhält man einen Graphen, der einen Ablauf beschreibt. Der Ablauf bezieht sich ausschließlich auf einen konkreten Leistungsgegenstand. Der Graph wird als Leistungsprozess bezeichnet. Dieses Vorgehen erlaubt die Trennung von Produktinformationen und Ablaufinformationen.

Leistungsprozesse lassen sich unabhängig von einem konkreten Projekt beschreiben. Beispielsweise lässt sich der Ablauf, bezogen auf die Zustandsänderungen, für die Entwicklung einer *Statik* unabhängig von einem konkreten Projekt isoliert betrachten und durch einen typischen Leistungsprozess beschreiben. Ebenso lässt sich der Leistungsprozess einer typischen *Stahlbetonwand* im Vorfeld beschreiben, ohne den konkreten Kontext im Projekt zu kennen. Unabhängig von einem konkreten Projekt entwickelte Leistungsprozesse werden als Leistungsprozessvorlagen bezeichnet.

Betrachtet man ein Projekt, so lässt sich dieses in Leistungsgegenstände zerlegen. Jedem Leistungsgegenstand wird eine Leistungsprozessvorlage zugeordnet. Aus dem Tupel, bestehend aus projektspezifischem Leistungsgegenstand und projektunabhängiger Leistungsprozessvorlage,

³ [Huhnt und Enge, 2006, Can algorithms support the specification of construction schedules?]

⁴ [Enge, 2005, Zustandsmodellierung als Grundlage für Ausführungsterminpläne]

⁵ Im Allgemeinen stellt ein Leistungsgegenstand ein Produkt dar. Dies ist aber nicht immer der Fall. Um allgemeingültig zu bleiben wird daher der Begriff Leistungsgegenstand verwendet.

lässt sich ein projektspezifischer Leistungsprozess ableiten. Leitet man für jeden Leistungsgegenstand eines Projekts auf der Grundlage der ihm zugeordneten Leistungsprozessvorlage einen Leistungsprozess ab, so entsteht ein Ablaufmodell, das aus unabhängigen Teilabläufen besteht.

Zusätzliche Verknüpfungen sind erforderlich, um die Teilabläufe miteinander in Beziehung zu setzen. Die Spezifikation der zusätzlichen Verknüpfungen erfolgt nur für unterschiedliche Teilabläufe. Alle zusätzlichen Verknüpfungen dienen der Vernetzung der ursprünglich unabhängigen Teilabläufe. Verknüpfungen innerhalb eines Teilablaufs sind aus der Leistungsprozessvorlage bekannt.

Die Spezifikation der zusätzlichen Verknüpfungen erfolgt über die Angabe von Voraussetzungen für Vorgänge. Im Gegensatz zu der klassischen Vorgehensweise werden die Vorgänge nicht direkt verknüpft. Verknüpfungen zwischen Vorgängen ergeben sich aus einer graphentheoretischen Wegbetrachtung. Zur Erläuterung: Ein Vorgang transformiert einen Leistungsgegenstand in einen Zustand. Wird einem anderen Vorgang dieser Zustand als Voraussetzung eingepreßt, ergibt sich eine indirekte Verknüpfung der Vorgänge der Weglänge *zwei*, die berechnet werden kann.

Das Vorgehen, grundsätzliche Vorgänge über Zustandsbeschreibungen von Leistungsgegenständen zu kapseln, erlaubt es, das Gedankenmodell des Planers explizit zu machen und darüber hinaus, die Vorgänge herausgelöst aus dem Gesamtablauf betrachten zu können. Dies trägt zur Nachvollziehbarkeit bei und reduziert die Komplexität.

1.1.2 Spezifikationsaufwand

Der Spezifikationsaufwand für ein Ablaufmodell ergibt sich aus dem Anlegen der Leistungsprozessvorlagen, der Zerlegung des Projekts in Leistungsgegenstände und der Spezifikation der zusätzlichen Abhängigkeiten. Eine Einsparung gegenüber der klassischen Vorgehensweise ergibt sich aus der Verwendung der Leistungsprozessvorlagen. Sie werden einmal angelegt und tragen bei n gleichen Leistungsgegenständen $(n - 1)$ -mal zu einer Einsparung bei. Die Einsparung ist um so größer, je mehr Leistungsprozessvorlagen aus bereits abgeschlossenen Projekten in einer projektübergreifenden Leistungsprozessvorlagenbasis zur Wiederverwendung zur Verfügung stehen.

Modelliert man nach dem hier beschriebenen Verfahren einen Bauablauf, so lässt sich dasselbe Ergebnis auf unterschiedliche Weise erzeugen. Verschiedene Einteilungen in Leistungsgegenstände haben unterschiedliche Mengen an Leistungsprozessvorlagen zur Folge und entspre-

chend eine unterschiedliche Menge an zusätzlichen Verknüpfungen. Die möglichen Varianten bewegen sich innerhalb von zwei Grenzfällen. Entweder wird das ganze Projekt als ein Leistungsgegenstand betrachtet, oder jeder Vorgang stellt eine Leistung dar, für die ein eigener Leistungsgegenstand modelliert werden muss. Im ersten Fall müsste eine Leistungsprozessvorlage entwickelt werden, die den gesamten Ablauf repräsentiert. Es gäbe genau einen Leistungsgegenstand, keine zusätzlich zu spezifizierenden Voraussetzungen, aber eine Leistungsprozessvorlage, deren Spezifikationsaufwand dem der klassischen Herangehensweise entspräche. Im anderen Fall bestünde jede Leistungsprozessvorlage lediglich aus genau einem Vorgang und einer Zustandsbeschreibung. Die Anzahl der Leistungsgegenstände entspräche der Anzahl der Vorgänge und alle Verknüpfungen müssten über zusätzlich zu spezifizierende Abhängigkeiten modelliert werden. Der Aufwand entspräche ebenfalls dem der klassischen Herangehensweise.

Jede Einteilung des Projekts in Leistungsgegenstände, bei der eine Leistungsprozessvorlage mehrmals zum Einsatz kommt, führt gegenüber der klassischen Herangehensweise zu einer Einsparung des Spezifikationsaufwands. Unterschiedliche Einteilungen können zu unterschiedlichen Aufwänden führen. Die Anzahl der Varianten, die möglich sind, ist abhängig von der Anzahl der unterschiedlichen Muster von Teilabläufen und der Häufigkeit der Teilabläufe mit gleichem Muster. Teilabläufe mit gleichem Muster finden sich überall dort, wo gleichartige Leistungsgegenstände hergestellt werden, beispielsweise bei der Herstellung eines Stahlbetonrahmens mit zwei Stützen und einem Balken. Derselbe Leistungsgegenstand ließe sich aber auch anders einteilen, indem die Stützen und der Balken als unterschiedliche Leistungsgegenstände modelliert würden. Unterschiedliche Einteilungen erfordern unterschiedliche Leistungsprozessvorlagen und resultieren in unterschiedlichen Spezifikationsaufwänden. Ist die Einteilung günstig, sind nur wenig Leistungsprozessvorlagen erforderlich, die sich oft wiederholen. Als ungünstig, bezogen auf den Spezifikationsaufwand, erweisen sich solche Einteilungen, die viele verschiedene Leistungsprozessvorlagen erfordern, die sich aber seltener wiederholen.

1.1.3 Fragestellung

Im Allgemeinen wird ein Projekt aus der Anschauung heraus in Leistungsgegenstände zerlegt. Die Einteilung ist fachlich motiviert. Ihr liegt eine gewerkweise, eine kalkulatorische oder eine nach Projektstrukturplan

orientierte Sicht zugrunde. Entsprechend der Einteilung in Leistungsgegenstände sind spezifische Leistungsprozessvorlagen erforderlich. Diese sind entweder gemäß der Anforderungen zu entwickeln oder sie stehen bereits aus abgeschlossenen Projekten zur Verfügung. Da Leistungsprozessvorlagen projektunabhängig sind, erweitert sich ihre Menge mit jedem behandelten Projekt. Sie stellen eine Wissensbasis dar, in der sich die Erfahrungen eines planenden Ingenieurs akkumulieren.

Der Spezifikationsaufwand von Ausführungsprozessen infolge der Anwendung des hier beschriebenen Verfahrens hängt stark von dem Zusammenspiel der Leistungsgegenstände und der Leistungsprozessvorlagen ab, die in die Modellierung einfließen. Bei der Komplexität von Bauabläufen lässt sich im Vorfeld nicht aus der Anschauung heraus ermitteln, für welches Zusammenspiel sich ein Minimum an Eingabeaufwand ergeben wird. Eine Aussage darüber, welches Zusammenspiel zu einem Minimum an Eingabeaufwand führt, lässt sich nur für einen vollständig entwickelten Bauablauf durch Rückrechnung ermitteln.

1.1.4 Zielsetzung

Um die Effizienz des Modellierungsansatzes weiter zu erhöhen, soll ein Ansatz entwickelt werden, der einen entwickelten Bauablauf als Eingabe erhält, für den das optimale Zusammenspiel aus Leistungsgegenständen und Leistungsprozessvorlagen berechnet wird. Die Gegenüberstellung der im Rahmen der Modellierung verwendeten Leistungsprozessvorlage mit denen, die sich aus der Rückrechnung als optimal erweisen, erlaubt die Anpassung der in der Wissensbasis enthaltenen Leistungsprozessvorlagen, um Bauabläufe von zukünftigen Projekten effizienter entwickeln zu können.

1.2 Herangehensweise

Ausgangspunkt ist ein modellierter Bauablauf. Losgelöst von seinem fachlichen Hintergrund handelt es sich um einen zyklensfreien, gerichteten Graphen, bestehend aus Knoten, die über Kanten in Beziehung stehen. Entfernt man aus diesem Graphen bestimmte Kanten, so dass unabhängige Komponenten entstehen, dann wird das Gebilde aus unabhängigen Komponenten als Zerlegung bezeichnet. Die Menge der entfernten Kanten wird als Menge der zusätzlichen Kanten beschrieben. Jede Zerlegung und ihre zugehörige Menge an zusätzlichen Kanten lässt sich bewerten.

Es wird eine Bewertung eingeführt, die ein Maß für den Spezifikationsaufwand darstellt. Gesucht sind diejenigen Zerlegungen des Graphen, für die die Bewertung minimal ist.

1.2.1 Muster in Bauabläufen

Jeder Knoten und jede Kante hat einen eindeutigen Identifikator, der sie unterscheidbar macht. Bildet man die Identifikatoren der Knoten und Kanten einer Komponente nach einer bestimmten Systematik auf eine Zeichenkette ab, so erhält man eindeutige Identifikatoren für eine Komponente. Ordnet man den Elementen eines Graphen neben dem Identifikator auch eine Beschriftung zu, die nicht eindeutig sein muss, so besteht die Möglichkeit, nach derselben Systematik wie für die Identifikatoren, die Beschriftungen der Knoten und Kanten einer Komponente auf eine Zeichenkette abzubilden. Diese gibt Aufschluss über die Gestalt einer Komponente. Die Zeichenkette, die sich auf der Grundlage der Beschriftungen für eine Komponente ergibt, wird als Muster bezeichnet. Verwendet man als Beschriftung die projektunabhängigen Bezeichner von Vorgängen (z.B. betonieren), Leistungsgegenständen (z.B. StB-Wand) und Zuständen (z.B. betoniert), so lässt sich aus dem Muster⁶ einer Komponente ein projektunabhängiger Teilablauf identifizieren. Das Muster einer Komponente kann genutzt werden, um die Menge der Komponenten einer Zerlegung in Äquivalenzklassen einzuteilen. Jede Äquivalenzklasse steht für eine unterschiedliche Leistungsprozessvorlage. Alle Komponenten derselben Äquivalenzklasse lassen sich von derselben Leistungsprozessvorlage ableiten. Die Einteilung der Komponenten in Äquivalenzklassen dient als Grundlage für die Bewertung einer Zerlegung.

1.2.2 Optimierungsproblem

Für die Bestimmung der Zerlegungen mit minimaler Bewertung wird ein Optimierungsproblem formuliert. Gegeben ist zum einen der Lösungsraum Ω , der die Menge der möglichen Zerlegungen eines gegebenen Bauablaufs umfasst und zum anderen die Bewertung einer Zerlegung als Zielfunktion. Gesucht ist die Teilmenge $\Omega_{opt} \subseteq \Omega$, für die die Bewertung der Elemente $\omega \in \Omega_{opt}$ minimal ist.

Bei dem Minimierungsproblem handelt es sich um ein Suchproblem. Nicht der minimale Wert der Bewertung ist von Interesse, sondern die Ele-

⁶ Anmerkung: Das Muster ist eine Zeichenkette, nicht der Teilablauf selbst.

mente $\omega \in \Omega_{opt}$, für die die Bewertung minimal ist. Um den Aufwand der Suche zu reduzieren, wird ein Branch-and-Bound-Verfahren eingesetzt.

1.2.3 Branch-and-Bound-Verfahren

Das Prinzip eines Branch-and-Bound-Verfahrens besteht darin, den Lösungsraum schrittweise in Teilmengen zu zerlegen und diejenigen Teilmengen aus der Betrachtung auszuschließen, die nachweislich keine bessere Lösung enthalten, als die bisher beste gefundene Lösung. Durch rekursives Zerlegen entsteht sukzessiv eine Baumstruktur. Mit jeder Zerlegung wächst der Baum und die Teilmengen werden kleiner. Erreicht man eine Teilmenge, die nur noch ein Element enthält, wird es mit der bisher besten Lösung verglichen und gegebenenfalls ersetzt oder ergänzt. Die Suche ist abgeschlossen, wenn keine weitere Teilmenge mehr zu untersuchen ist. Im ungünstigsten Fall müssen alle Lösungen untersucht werden.

1.3 Gliederung

Die Arbeit gliedert sich inhaltlich in zwei Teile:

- Bauablaufplanung
- Mustererkennung in Bauabläufen

1.3.1 Bauablaufplanung

Die Komplexität von Bauprojekten ist auf eine Vielzahl von Ursachen zurückzuführen. Viele Beteiligte aus unterschiedlichen Fachdisziplinen arbeiten zusammen. Bauherren wünschen individuelle Lösungen. Bauwerke unterscheiden sich in Funktion, in Konstruktion und in den verwendeten Materialien. Unterschiedliche Bauverfahren werden angewendet. Zeitliche und finanzielle Rahmenbedingungen variieren von Projekt zu Projekt. Die Variationsvielfalt an Detaillösungen ist unerschöpflich. Als Konsequenz dieser Eigenschaften der Bauprojekte müssen die Bauprozesse individuell für jedes Projekt neu betrachtet und durchdacht werden. Dies betrifft insbesondere die Erarbeitung der Bauabläufe.

Kapitel 2 geht auf die Eigenschaften von Bauabläufen und die Bauablaufplanung ein. Relevante Begriffe werden eingeführt und abgegrenzt. Es werden Ansätze aus der Forschung skizziert, deren Ziel es ist, den Spezifikationsaufwand beim Erstellen zu reduzieren und die Qualität der Ergeb-

nisse gegenüber den aus der Anschauung heraus entwickelten Bauabläufen zu erhöhen.

In Kapitel 3 wird auf der Grundlage der Ansätze von Huhnt und Lawrence [2004], Huhnt [2005] und Huhnt und Enge [2006] eine Modellierungsmethodik zum Erstellen von Bauabläufen beschrieben, die die logischen Abhängigkeiten, die technologisch bedingt sind, in den Mittelpunkt rückt, und die es erlaubt, auf der Grundlage von formalen Algorithmen die Abfolge von Vorgängen zu berechnen. Der Vorteil liegt in der Qualität der berechneten Ergebnisse, die gesichert vorliegen und deren Eigenschaften dem zugrunde liegenden mathematischen Verfahren entsprechen. Teilabläufe werden projektunabhängig beschrieben und als Vorlagen bereitgestellt. Innerhalb eines Projekts werden die Vorlagen genutzt, um für konkrete Leistungen Teilprozesse abzuleiten. Über die Angabe von Voraussetzungen werden Teilprozesse miteinander verknüpft.

Das Ergebnis ist ein Graph, der sowohl die technologisch zwingenden Abhängigkeiten vollständig und konsistent beschreibt, als auch eine Aussage über die Qualität im Sinne der fachlichen Korrektheit zulässt. Damit steht ein Rohling für die Netzplantechnik zur Verfügung, der um weitere Angaben, wie beispielsweise Ressourcenbegrenzungen, Terminvorgaben etc., zu ergänzen ist. Auf Grund der formalen Herangehensweise sind die Eigenschaften des resultierenden Prozessmodells mathematisch begründet und die Vollständigkeit und Korrektheit kann konsistent zur Eingabe gewährleistet werden.

1.3.2 Mustererkennung in Bauabläufen

Der Kern der Arbeit befasst sich mit der Minimierung des Spezifikationsaufwands von Bauabläufen, die auf der Grundlage der zustandsbasierten Methodik zum Gestalten von Bauabläufen modelliert werden.

Abhängig von der Einteilung des Projekts in Leistungsgegenstände ergibt sich für die Modellierung eines Bauablaufs ein konkreter Spezifikationsaufwand. Unterschiedliche Einteilungen des Projekts in Leistungsgegenstände können im resultierenden Spezifikationsaufwand variieren. Die Ermittlung einer optimalen Einteilung des Projekts in Leistungsgegenstände stellt eine Umkehrung des vorgestellten Modellierungsansatzes dar. Aus einem gegebenen Ablauf wird eine Menge an Zerlegungen gesucht, für die der Spezifikationsaufwand minimal ist.

Die Ermittlung optimaler Zerlegungen lässt sich als Suche nach Mustern in einem Graphen auffassen. Da die Muster, nach denen im Graphen gesucht werden soll, im Vorfeld nicht bekannt sind, sondern selbst eben-

falls einen Teil der Lösung darstellen, ist die Komplexität der Problemstellung exponentiell. Nur durch die Berücksichtigung von Randbedingungen, die sich aus dem Modellierungsansatz ergeben, lässt sich die Komplexität der Suche reduzieren.

In Kapitel 4 wird ein Branch-and-Bound-Verfahren beschrieben, das den Lösungsraum vollständig berücksichtigt und ohne Heuristiken die Menge der optimalen Zerlegungen bestimmt.

Hiermit liegt eine Methodik vor, um die Qualität der Leistungsvorlagen rückblickend für einen gegebenen Prozess zu beurteilen. Die Auswertung abgeschlossener Projekte kann dazu beitragen, die Basis der Leistungsvorlagen für anstehende, ähnliche Projekte zu optimieren.

Kapitel 5 behandelt die Auswertung von Bauabläufen. An Beispielen wird das Verhalten des Branch-and-Bound-Verfahrens beschrieben und ausgewertet.

Die gewonnenen Erkenntnisse werden in Kapitel 6 zusammengefasst. Die Arbeit schließt in Kapitel 7 mit einem Ausblick.

Ergänzend werden im Anhang A Grundbegriffe und Definitionen aus der Mengen- und Relationenalgebra sowie der Graphentheorie zusammengefasst. Sie dienen der Abgrenzung, um ein einheitliches Verständnis zu schaffen. Der Betrachtungsweise dieser Arbeit liegen die mathematischen Grundlagen von Pahl und Damrath [2000] zugrunde.

Kapitel 2

Bauablaufplanung

Bauvorhaben benötigen eine detaillierte Planung, um die gesetzten Ziele unter Einhaltung von Termin- und Kosten- und Qualitätsvorgaben realisieren zu können. Dieses Kapitel geht auf die Eigenschaften von Bauvorhaben ein, deren Randbedingungen dazu führen, dass Bauabläufe einzigartig sind und individuelle Vorbereitung erfordern. In Abschnitt 2.1 werden grundlegende Begriffe eingeordnet und erläutert. Darauf aufbauend wird in Abschnitt 2.2 auf die Aufgaben der Bauablaufplanung und ihre spezifischen Anforderungen eingegangen. In Abschnitt 2.3 erfolgt eine Beschreibung der heutigen Vorgehensweise zum Erstellen von Bauabläufen sowie eine Beschreibung von Ansätzen aus der Forschung und Entwicklung. Aus der Zusammenfassung am Ende des Kapitels ergeben sich Anforderungen, die Einfluss auf die Entwicklung des Modellierungsansatzes für Bauabläufe haben und Teil des nächsten Kapitels sind.

2.1 Begriffe und Abgrenzungen

2.1.1 Projekt

Als Synonym für den Begriff *Vorhaben* wird der Begriff *Projekt* verwendet. Eine allgemeingültige Definition des Projektbegriffs liefert die DIN 69901-5 [2009]¹. Der Begriff *Projekt* wird definiert als ein

„Vorhaben, das im Wesentlichen durch die Einmaligkeit der Bedingungen in ihrer Gesamtheit gekennzeichnet ist, wie z.B. Zielvorgabe, zeitliche, finanzielle, personelle und andere Begrenzungen, projektspezifische Organisation.“

¹ [DIN 69901-5, 2009, Projektmanagement - Projektmanagementsysteme - Teil 5: Begriffe, S. 11]

Vor dem Hintergrund der Einmaligkeit der Bedingungen haben Bauvorhaben Projektcharakter. Im Gegensatz zu dem Begriff *Vorhaben*, der die Absicht etwas zu tun hervorhebt, drückt der Begriff *Projekt* seine konkrete Umsetzung aus. Ein Projekt hat einen zeitlich festgelegten Rahmen mit Anfangs- und Endtermin und ist aus der Sicht von Beteiligten mit einem konkreten Projektauftrag verbunden. Das *Project Management Institute*² definiert ein Projekt aus der Sicht des Projektmanagements wie folgt:

„Ein Projekt ist ein zeitlich begrenztes Unternehmen, das unternommen wird, um ein einmaliges Produkt, eine Dienstleistung oder ein Ergebnis zu erzeugen.“

Der Projektbegriff dient der Abgrenzung der Tätigkeiten, die mit der erfolgreichen Abwicklung eines Projektes zusammenhängen, von denen, die insgesamt in einem Unternehmen anfallen. Schwarze [2006]³ definiert das Projekt mit deutlichem Bezug zur Ausführung:

„Ein Projekt ist ein zeitlich, räumlich und sachlich begrenztes komplexes Arbeitsvorhaben, bei dem durch den Einsatz von Verbrauchsgütern (Material, Energie usw.), Nutzungsgütern (z.B. Maschinen) und Arbeitskräften eine bestimmte Zielsetzung (Aufgabe) zu erreichen ist.“

2.1.2 Prozess

Der Projektbegriff nimmt Bezug auf die Gesamtheit einer Vielzahl von Arbeitsabläufen, um unter Berücksichtigung von Zwängen bezüglich der Zeit, der Kosten und der Ressourcen ein Ziel zu erreichen. Der Prozessbegriff hingegen legt den Fokus auf die formale Beschreibung in sich geschlossener Arbeitsabläufe. Ein Projekt als Ganzes stellt einen Prozess dar. Aber auch in sich abgeschlossene, in einem Projekt enthaltene Arbeitsabläufe lassen sich als Prozesse beschreiben. Im Gegensatz zu einem Projekt kann sich ein Prozess wiederholen.

Der Begriff Prozess gehört zu der Terminologie verschiedener Anwendungsgebiete. Innerhalb der Anwendungsgebiete existieren unterschiedliche Definitionen, denen unterschiedliche Anschauungen zugrunde liegen. Entsprechend finden sich in der wissenschaftlichen Literatur eine Vielzahl an unterschiedlichen Begriffserklärungen.

Den nachfolgenden Betrachtungen liegen Ansätze aus der Betriebswirtschaftslehre und der Wirtschaftsinformatik zugrunde. Aus diesen

² [PMBOK, 2003, A Guide to the Project Management Body of Knowledge, S. 4]

³ [Schwarze, 2006, Projekte und Projektmanagement, S. 13]

Anwendungsgebieten heraus wurden prozessorientierte Methodiken mit dem Ziel entwickelt⁴, die Wettbewerbsfähigkeit von Unternehmen durch Leistungsverbesserung sowie die Verbesserung der Kundenzufriedenheit und die Zufriedenheit anderer interessierter Parteien zu erhöhen. Eine allgemeingültige Definition des Prozessbegriffs, auf den sich die unterschiedlichen Ansätze im Kern zusammenfassen lassen, findet man in der DIN EN ISO 9000 [2005]⁵. Danach ist ein Prozess definiert, als ein

„Satz von in Wechselbeziehung oder Wechselwirkung stehenden Tätigkeiten, der Eingaben in Ergebnisse umwandelt.“

Aus betriebswirtschaftlicher Sicht wird der Prozess als Geschäftsprozess bezeichnet und konkretisiert. Unterschiedliche Definitionen heben dabei verschiedene Aspekte⁶ hervor. Scheer⁷ beschreibt einen Geschäftsprozess als

„eine zusammengehörige Abfolge von Unternehmensverrichtungen zum Zweck einer Leistungserstellung. Ausgang und Ergebnis des Geschäftsprozesses ist eine Leistung, die von einem internen oder externen Kunden angefordert und abgenommen wird.“

Betrachtet man eine Unternehmensverrichtung als Vorgang und die durch einen Vorgang erzielte Leistung als Zustand, so lässt sich das Arbeitsvorhaben eines Projektes im Bauwesen wie folgt als Prozess beschreiben:

Prozess: Ein Prozess stellt eine Zerlegung eines komplexen Arbeitsvorhabens in eine Menge an Vorgängen und Zuständen dar, für die eine Abfolge definiert ist. Vorgänge stellen zeitfordernde Geschehen dar. Ihnen werden Ressourcen, Maschinen und Material zugeordnet. Zustände stellen Ereignisse dar, die einen Vorgang auslösen und die das Ergebnis eines Vorgangs beschreiben.

⁴ [Gaitanides, 1983, Prozessorganisation: Entwicklung, Ansätze und Programme prozessorientierter Organisationsgestaltung], [Davenport, 1992, Process Innovation: Reengineering Work Through Information Technology]

⁵ [DIN EN ISO 9000, 2005, Definition des Prozessbegriffs aus der Sicht des Qualitätsmanagements, Abs. 3.4.1, S 23]

⁶ [Stapf, 2000, Übersicht über unterschiedliche Definitionen des Geschäftsprozess, S. 28 ff.]

⁷ [Scheer, 1998, ARIS - Vom Geschäftsprozess zum Anwendungssystem, S. 3]

2.1.3 Prozessorientierter Ansatz

Ein Projekt lässt sich effizienter umsetzen, wenn Vorgänge und dazugehörige Ressourcen als Prozess modelliert und gesteuert werden. Damit sich Organisationen wirksam betätigen können, müssen sie zahlreiche miteinander verknüpfte und in Wechselwirkung zueinander stehende Prozesse erkennen und handhaben. Oft bildet das Ergebnis des einen Prozesses die direkte Eingabe für den nächsten. Das systematische Erkennen sowie Handhaben dieser verschiedenen Prozesse innerhalb einer Organisation, vor allem aber der Wechselwirkungen zwischen solchen Prozessen, wird als *prozessorientierter Ansatz*⁸ bezeichnet.

2.1.4 Prozessmodell

Der Begriff *Modellieren* beschreibt das Abbilden eines Realitätsausschnitts auf abstrakte Beschreibungsgrößen. Fasst man gleichartige Beschreibungsgrößen zu Mengen zusammen und beschreibt ihre Zusammenhänge durch Relationen, erhält man ein strukturiertes Datenmodell. Um dieses Datenmodell bearbeiten zu können, sind Operationen erforderlich. Diese sind für die Mengen und die Relationen zu definieren. Regeln müssen aufgestellt werden, um die Verträglichkeit der Operationen untereinander sicherzustellen.

Prozessmodell: Als Prozessmodell wird das Gebilde aus Mengen, Relationen, Operationen und Regeln bezeichnet, das in seiner Gesamtheit zur formalen Beschreibung und Bearbeitung von Arbeitsabläufen dient.

2.2 Randbedingungen der Bauablaufplanung

Ablaufplanungsprobleme treten in unterschiedlichen Anwendungsgebieten auf. In der stationären Industrie befasst sich die Ablaufplanung mit der Produktionsplanung. Allgemein besteht das Ziel einer Produktionsplanung darin, die operative, zeitliche und mengenmäßige Planung, Steuerung und Kontrolle der Produktion von Waren und Gütern durchzuführen und diese kontinuierlich zu optimieren⁹. Gesucht wird ein Ablaufplan,

⁸ [DIN EN ISO 9000, 2005, Prozessorientierter Ansatz, Abs. 2.4, S. 8]

⁹ [Schuh, 2006, Kapitel 2: Grundlagen der Produktionsplanung und -steuerung, S. 11 ff.]

der eine möglichst optimale, terminliche Zuordnung der einzelnen Vorgänge zu Ressourcen vornimmt und dabei gegebene Nebenbedingungen und Zielsetzungen erfüllt.

2.2.1 Abgrenzung zur Produktionsplanung

Im Bauwesen treten nach Bauer [2006]¹⁰ Ablaufplanungsprobleme auf zwei unterschiedlichen Ebenen auf. Zum einen befasst sich die Ablaufplanung mit der Planung des Bauablaufs eines konkreten Bauvorhabens und zum anderen mit der übergeordneten Koordination von Bauvorhaben aus der Sicht eines Beteiligten.

Die zweite Form der Ablaufplanung kann als Produktionsplanung - analog zur stationären Industrie - aufgefasst werden. Jedes Bauvorhaben stellt für einen Beteiligten einen Auftrag dar und das abzuliefernde Ergebnis ein Produkt. Die Menge aller Aufträge einer Unternehmung beschreibt den Produktionsumfang. Gesucht wird ein Ablaufplan, der alle Produktionsfaktoren eines Unternehmens optimal einsetzt.

Voraussetzung für eine optimale Einteilung der Produktionsfaktoren eines Unternehmens ist, dass eine optimale Planung der jeweiligen Bauabläufe existiert, für die es einen Auftrag hat. Nur wenn für jedes Bauvorhaben eine zuverlässige Ablaufplanung vorliegt, können Störungen der übergeordneten Planung vermieden werden. Diese wirken sich wiederum auf die Ablaufplanung eines Bauvorhabens aus.

Im Rahmen dieser Arbeit wird nur auf die Ablaufplanung eines einzelnen Bauvorhabens eingegangen, nicht auf die übergeordnete Koordination von mehreren Bauvorhaben. Ihr liegt das Verständnis nach Würfele und Bielefeld [2007]¹¹ zugrunde:

„Unter Ablaufplanung versteht man die Planung und Organisation aller für den jeweiligen Ablaufplan relevanten Vorgänge und Ereignisse zur Ermittlung der Abhängigkeiten zwischen den Vorgängen.“

Die Ablaufplanung eines einzelnen Bauvorhabens wird im Folgenden als Bauablaufplanung bezeichnet. Das Ergebnis einer Bauablaufplanung ist ein Bauablaufplan. Unterschiedliche Rollen der Beteiligten eines Bauprojekts führen zu unterschiedlichen Sichtweisen.

¹⁰ [Bauer, 2006, Zur Abgrenzung der Produktionsplanung in der stationären Industrie, S. 528]

¹¹ [Würfele und Bielefeld, 2007, Allgemeine Begriffe der Termin- und Ablaufplanung, S. 30]

2.2.2 Sichtweisen unterschiedlicher Beteiligter

An einem Bauvorhaben sind eine Vielzahl an Personen aus unterschiedlichen Fachdisziplinen beteiligt. Diese lassen sich unterschiedlichen Rollen zuordnen, z.B. Bauherr, Architekt, Fachingenieur, Behörde, Baufirma oder Lieferant. Für jedes Bauvorhaben ist die Zusammensetzung der Beteiligten eine andere. Sie gehören unterschiedlichen Organisationen an und haben unterschiedliche Sichtweisen.

Die Anforderungen an eine Ablaufplanung sind abhängig von der Rolle, die ein Beteiligter innerhalb eines Projekts einnimmt. Bauer [2006]¹² unterscheidet zwei grundsätzliche Sichtweisen: die des Auftraggebers und die des Unternehmers. Für den Auftraggeber steht die zeitliche Koordination aller Vorbereitungs- und Planungshandlungen, Bereitstellungen, Genehmigungen usw. im Mittelpunkt, um unterschiedliche Projektbeteiligte zu koordinieren und einen störungsfreien Projektablauf zu gewährleisten.

„Die Ablaufplanung des Bauherrn ist also eine Generalplanung. Sie dient der Koordination und Überwachung des Bauvorhabens von der Konzeption der Bauidee bis zur Ingebrauchnahme des Werkes.“¹³

Die Ablaufplanung aus der Sicht des Unternehmers dient der Ermittlung des kostengünstigsten Weges zur Herstellung von Bauwerken mit vorgegebenen Randbedingungen. Der Detaillierungsgrad ist so zu wählen, dass die Dauer des Gesamtvorhabens und Termine für die Ausschreibung, die Vergabe und die Fertigstellung einzelner Bauabschnitte und die Termine für die Ausbau- und Zulieferfirmen bestimmt werden können.

Die Anforderungen an eine Ablaufplanung, die sich aus den Aufgabenbereichen einer Unternehmung ergeben, sind durch die Bauausführung geprägt. Nicht die übergeordnete Koordination steht im Mittelpunkt, sondern die Steuerung der Realisierung der Ausführung. Der Ablaufplanung geht die Auswahl zweckmäßiger Verfahren unter Berücksichtigung von Ressourcen, Betriebsmitteln und Werkstoffen voraus. Sobald die Herstellungsverfahren festgelegt sind, können sie in Arbeitspakete bzw. Vorgänge heruntergebrochen werden. Diese sind unter Berücksichtigung von vorgeschriebenen Bauzeiten, Kapazitätsgrenzen, Meilensteinterminen und ihren Abhängigkeiten untereinander in ihrer Abfolge zeitlich anzuordnen. Nach Bauer¹⁴ sind folgende Anforderungen zu erfüllen:

¹² [Bauer, 2006, Aufgabe der Ablaufplanung, S. 528]

¹³ [Bauer, 2006, Zur Sichtweise des Auftraggebers auf die Ablaufplanung, S. 529]

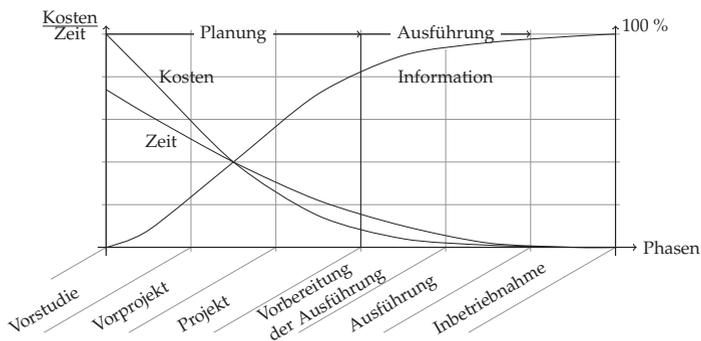
¹⁴ [Bauer, 2006, Zur Sichtweise des Unternehmers auf die Ablaufplanung, S. 529]

- „Die Bestimmung der Dauer der Teilvorgänge und der Gesamtbauzeit aufgrund zur Verfügung stehender Produktionskapazitäten,
- die Koordinierung der Teilvorgänge zu technologisch verträglichen Bauvorgangsketten und
- die Optimierung des Bauvorganges, vor allem der wirtschaftlich oder technisch maßgebenden Bauleistungen.“

Neben der Sichtweise der unterschiedlichen Beteiligten hat der Detaillierungsgrad der zur Verfügung stehenden Informationen einen wesentlichen Einfluss auf die Bauablaufplanung.

2.2.3 Planungsphasen und Detaillierungsgrad

Nach Brandenberger und Ruosch [1991]¹⁵ ist der gesamte Projektablauf in Phasen strukturiert. Für Bauprojekte sind dies die Grobphasen Planung (Gestaltung) und Ausführung (Realisierung), die sich weiter aufgliedern in die Detailphasen Vorstudie, Vorprojekt, Projekt, Vorbereitung der Ausführung, Ausführung und Inbetriebnahme.



In Anlehnung an:
 [Brandenberger und Ruosch, 1991, S. 26]
 und [Stark und Brecheler, 2006, S. 35]

Abbildung 2.1: Einsparungsmöglichkeiten im Projektverlauf

¹⁵ [Brandenberger und Ruosch, 1991, Projektmanagement im Bauwesen, S. 22]

Durch die Strukturierung in Phasen wird ein Projekt nach den Erkenntnissen des Projektmanagements in Abschnitte gegliedert. Die Übergänge kennzeichnen charakteristische Momente im Planungsablauf, an denen Entscheidungen über den weiteren Verlauf getroffen werden und überprüft werden kann, ob die Anforderungen des Auftraggebers eingehalten werden.

Mit jeder Phase kommen Entscheidungen hinzu, die die Freiheitsgrade einschränken und den Detaillierungsgrad der Planung verfeinern. Festlegungen, die getroffen werden, wirken sich auf die Projektdauer und die Projektkosten aus. Mit zunehmendem Projektfortschritt sinken die Einsparmöglichkeiten, um Kosten durch Anpassungen in der Planung beeinflussen zu können. Ein qualitativer Zusammenhang ist in Abbildung 2.1 gezeigt.

Der Ausschuss der Ingenieurverbände für die Honorarordnung e.V. (AHO)¹⁶ empfiehlt eine Einteilung der Projektphasen in Projektvorbereitung, Projektplanung, Vorbereitung der Ausführung, Ausführung und Projektdokumentation/Inbetriebnahme. Die Verordnung über die Honorare für Leistungen der Architekten und Ingenieure (HOAI)¹⁷ sieht eine Einteilung in 9 Projektphasen vor. Abbildung 2.2 zeigt eine Gegenüberstellung der Projektphasen nach AHO und HOAI.

Wie aus den Phaseneinteilungen ersichtlich, erreicht die Bauablaufplanung im Verlauf der Projektphasen verschiedene Planungstiefen. Nach Bauer [2006]¹⁸ sind drei Stufen zu unterscheiden:

1. „Der Grob Ablauf- oder Rahmenterminplan, der als Übersichtsplan für die wichtigsten Bauabschnitte Rahmentermin festlegt (Zeiteinheit 1 Monat),
2. der Koordinationsablaufplan, er umfasst ebenfalls das gesamte Projekt und dient der Steuerung und Kontrolle des Bauablaufs (Zeiteinheit 1 Arbeitstag) und
3. die Feinablaufpläne, damit werden im Zuge der Bauausführung die einzelnen Arbeitsabschnitte detailliert (Zeiteinheit 1 AT).“

Der Grob Ablaufplan dient der Ermittlung von Rahmenterminen und der langfristigen Steuerung des Bauablaufs. Er wird in der Projektvorberei-

¹⁶ [AHO, 1998, Projektsteuerung - Untersuchungen zum Leistungsbild des §31 HOAI und zur Honorierung für die Projektsteuerung]

¹⁷ [HOAI, 2006, Verordnung über die Honorare für Leistungen der Architekten und Ingenieure (HOAI). Eurofassung 2002]

¹⁸ [Bauer, 2006, Abschnitt: 11.4 Planungsschritte, S 538 ff.]

	Projektphasen AHO	Leistungsphasen nach §15 HOAI
Grobablaufplan	1. Projektvorbereitung	1. Grundlagenermittlung
Koordinationsablaufplan, Feinablaufplan Planung	2. Projektplanung	2. Vorplanung 3. Entwurfsplanung 4. Genehmigungsplanung
Feinablaufplan Ausführung	3. Vorbereitung der Ausführung	5. Ausführungsplanung 6. Vorbereiten der Vergabe 7. Mitwirkung bei der Vergabe
	4. Ausführung Objekterstellung 5. Projektdokumentation Inbetriebnahme	8. Objektüberwachung (Bauüberwachung) 9. Objektbetreuung und Dokumentation

Abbildung 2.2: Strukturierung des Projekts in Phasen

tung aufgestellt (HOAI: Leistungsphase 1) und berücksichtigt in erster Linie die Anforderungen des Bauherren. In der Projektplanungsphase (HOAI: Leistungsphase 2-4) wird unter Einhaltung der Vorgaben aus dem Grobablaufplan der Koordinationsablaufplan aufgestellt. Er dient der Projektleitung und dem Projektsteuerer als Instrument zur Koordination und Kontrolle der Beteiligten. Dem Koordinationsablaufplan liegt die Sichtweise des Bauherrn zugrunde. Außerdem sind in dieser Phase die Feinablaufpläne für die Planungsprozesse anzufertigen. Ihnen liegt die Sichtweise von Unternehmen zugrunde. In der Phase der Vorbereitung der Ausführung (HOAI: Leistungsphase 5-7) wird der Feinablaufplan für die Ausführungsprozesse angefertigt. Ihm liegt die Sichtweise von Unternehmen zugrunde. Für eine detaillierte Beschreibung der Detaillierungsgrade und ihrer Anforderungen sei auf Kochendörfer et al. [2007]¹⁹ und Bauer [2006]²⁰ verwiesen.

¹⁹ [Kochendörfer et al., 2007, Abschnitt: 5.2 Detaillierungsgrade von Ablaufplänen, S. 97 ff.]

²⁰ [Bauer, 2006, Kapitel: 11 Ablaufplanung, S. 527 ff.]

2.2.4 Arten von Abhängigkeiten

Abhängigkeiten zwischen Vorgängen können verschiedene Ursachen haben. Würfele und Bielefeld [2007]²¹ unterscheiden in diesem Zusammenhang vier verschiedenartige Gruppen von Abhängigkeiten:

1. technologisch bedingte, zwingende Abhängigkeiten (z. B. Beginn des Ausbaus erst nach der Fertigstellung der Gebäudehülle),
2. vorgegebene, externe Randbedingungen (z. B. an Fördergelder gebundene Zwischentermine, zwingend einzuhaltender Fertigstellungstermin),
3. kapazitätsbedingte Abhängigkeiten (z. B. Kranauslastung, Personalengpässe durch Ferienzeit) und
4. terminplantechnische, durch den Planer vorgegebene Abhängigkeiten (z. B. Entzerrung des Terminplans durch Einbau von Pufferzeiten, Ressourcenanpassung).

Im Rahmen dieser Arbeit stehen die technologisch bedingten, zwingenden Abhängigkeiten im Mittelpunkt der Betrachtungen.

2.3 Methoden und Verfahren der Bauablaufplanung

Abhängig von der Rolle eines Planers und dem angestrebten Detaillierungsgrad wird ein Arbeitsverzeichnis aufgestellt, das alle zu erbringenden Teilleistungen erfasst. Auf der Grundlage dieses Arbeitsverzeichnisses werden Vorgänge ermittelt und deren logische Reihenfolge festgelegt. Für jeden Vorgang wird eine Dauer bestimmt. Je nach Sichtweise und Detaillierungsgrad unterscheiden sich die zu modellierenden Randbedingungen und die Vorgehensweise zur Bestimmung von Dauern. Software-Werkzeuge stehen zur Verfügung, die den bearbeitenden Ingenieur bei seiner Tätigkeit unterstützen. Sie erlauben die graphische Bearbeitung, die Spezifikation von Abhängigkeiten und die Überprüfung von Eigenschaften des zugrunde liegenden Graphen. Die Logik der Eingabe des modellierten Ablaufs wird jedoch nicht adressiert. Es bleibt der Erfahrung der

²¹ [Würfele und Bielefeld, 2007, Klärung und Darstellung der Abhängigkeiten der Vorgänge untereinander, S. 54 ff.]

bearbeitenden Ingenieure überlassen, Randbedingungen zu erkennen und die Reihenfolge der Teilleistungen entsprechend abzustimmen. Eine Aussage darüber, ob alle logischen Abhängigkeiten berücksichtigt sind, kann nur aus der Anschauung heraus getroffen werden.

2.3.1 Terminieren von Vorgängen

IT-gestützte Terminplanungswerkzeuge stellen für die Terminierung von Vorgängen - der Zuordnung eines Zeitfensters zu einem Vorgang - zwei verschiedene Möglichkeiten zur Verfügung: die manuelle Terminierung und die automatische Terminierung. Bei der manuellen Terminierung ordnet man einem Vorgang manuell einen festen Start- oder Endtermin zu. Ein korrespondierender End- oder Starttermin ergibt sich aus der Dauer des Vorgangs. Die automatische Terminierung von Vorgängen erfordert die Angabe von Anordnungsbeziehungen, aus denen sich auf der Grundlage der in der Netzplantechnik definierten Vorwärts- und Rückwärtsrechnung Zeitfenster für die Vorgänge berechnen lassen. Beide Arten sind innerhalb derselben Modelle zulässig. Problematisch sind Netzpläne, bei denen aus fachlicher Sicht Abhängigkeiten existieren, diese aber nicht über Anordnungsbeziehungen modelliert werden, sondern über die Angabe von festen Terminen. Änderungen an Vorgängen können sich nur über den Teil des Netzplans auswirken, der über Anordnungsbeziehungen verknüpft ist. Die Tatsache, dass Abhängigkeiten nicht zwingend über Anordnungsbeziehungen modelliert werden müssen, führt dazu, dass Änderungen immer eine manuelle Überprüfung des gesamten Netzplans nach sich ziehen. Für diese Kontrolle besteht keine formale Unterstützung. Fehler lassen sich nur schwer ermitteln. Die Korrektheit hängt ausschließlich von der Erfahrung des Bearbeiters ab.

2.3.2 Netzplantechnik

Die Ablaufplanung befasst sich mit dem Problem, Vorgänge in ihrer logischen Reihenfolge so zu ordnen, dass unter Berücksichtigung von Randbedingungen vorgegebene Zielsetzungen erreicht werden. Aufbauend auf der Ablaufplanung erfolgt eine Dauernplanung. Für jeden Vorgang ist die Dauer zu ermitteln. Die Ablaufplanung und die Dauernplanung sind die Voraussetzungen für die Terminplanung, bei der für jeden Vorgang ein Zeitfenster terminiert wird. Die Grundlage der Terminplanung ist die Netzplantechnik.

Ein Netzplan ist eine Darstellungsform, die es erlaubt, komplexe Pro-

zesse in Form von Knoten-Kanten-Diagrammen zu veranschaulichen. Die Netzplantechnik stellt die Methodik dar, mit der die zugrunde liegenden Graphen analysiert und ausgewertet werden können²². Die theoretische Grundlage der Netzplantechnik ist die Graphentheorie.

Erfüllt der Graph bestimmte Eigenschaften, lässt sich durch die in der Netzplantechnik definierte Vorwärts- und Rückwärtsrechnung für jeden Vorgang ein Zeitfenster berechnen (früherster Anfang FA, frühestes Ende FE, spätester Anfang SA und spätestes Ende SE). Dieses Zeitfenster gibt an, wann ein Vorgang durchzuführen ist, ohne dass Konflikte mit anderen Vorgängen des Bauablaufs entstehen. Die Netzplantechnik umfasst nach DIN 69900 [2009]²³

„alle Verfahren der Analyse, Beschreibung, Planung, Steuerung und Überwachung von Abläufen auf der Grundlage der Graphentheorie, wobei Zeit, Kosten, Einsatzmittel und weitere Einflussgrößen berücksichtigt werden können.“

Bei einem Vorgangsknoten-Netzplan (VKN) werden die Knoten des Graphen mit Vorgängen assoziiert und die Kanten mit der Anordnungsbeziehung zwischen zwei Vorgängen. Den Knoten wird eine Dauer zugeordnet und den Kanten ein festgelegter Anordnungstyp und eine Dauer. Die Methode, basierend auf Vorgangsknoten-Netzplänen, wird in der Literatur als *Metra Potential Method (MPM)* bezeichnet.

Bei einem Vorgangspfeil-Netzplan (VPN) werden die Vorgänge als Pfeile modelliert und ihre Ergebnisse als Knoten. Ergebnisse stellen Ereignisse dar, die wiederum als Auslöser für andere Vorgänge in Erscheinung treten. Die Methode, basierend auf Vorgangspfeil-Netzplänen, wird in der Literatur als *Critical Path Method (CPM)* bezeichnet.

Bei einem Ereignisknoten-Netzplan (EKN) werden die Ereignisse (Zustände) als Knoten und die zeitlichen Abhängigkeiten als Pfeile dargestellt. Sie sind in ihrer Struktur ähnlich wie die Vorgangspfeil-Netzpläne. Eine häufig beachtete Variante stellt die *Program Evaluation and Review Technique (PERT)* dar. Hierbei werden die Dauern von Vorgängen nicht deterministisch festgelegt, sondern in Form einer Wahrscheinlichkeitsverteilung angegeben. Darüber hinaus gibt es weitere Verfahren, auf die hier nicht näher eingegangen wird.

²² [Götzke, 1969, Netzplantechnik], [Altrogge, 1979, Netzplantechnik], [Brandenberger und Ruosch, 1993, Ablaufplanung im Bauwesen], [Seeling, 1996, Projektsteuerung im Bauwesen]

²³ [DIN 69900, 2009, Projektmanagement - Netzplantechnik; Beschreibungen und Begriffe, Abs. 3.51, S. 10]

Aufgrund der methodischen Herangehensweise wird die Netzplantechnik als eine der wichtigsten Instrumente des Projektmanagements angesehen²⁴. Ihre Methodik setzt aber erst ein, nachdem die Abhängigkeiten zwischen den Vorgängen und die Dauern der Vorgänge bekannt sind. Sie unterstützt nicht die Gestaltung und die Entwicklung konsistenter Prozessmodelle, aus denen die Graphen für die graphentheoretischen Berechnungen abgeleitet werden. Insbesondere die Ermittlung der Abhängigkeiten im Rahmen der Ablaufplanung entscheidet über die Qualität der Ergebnisse. Werden Abhängigkeiten zwischen Vorgängen nicht erkannt, dann hat dies keinen Einfluss auf die Berechnungen, jedoch verursachen diese Fehler zum Teil gravierende Störungen im Verlauf der Projektdurchführung.

2.3.3 Ansätze aus Forschung und Entwicklung

Im Bereich des Operations Research werden bereits seit Anfang der 50er Jahre Ablaufplanungsprobleme untersucht. Wegen ihrer Komplexität werden in erster Linie idealisierte Probleme aus der stationären Industrie betrachtet, z.B. das *Job-Shop-Problem*²⁵ und das daraus abgeleitete *Flow-Shop-Problem*. Es geht darum, eine Menge von Aufgaben auf einer zur Verfügung stehenden Anzahl von verschiedenen Maschinen auszuführen. Aufgaben setzen sich aus Teilaufgaben zusammen. Das Ziel ist es, alle Aufgaben in einer minimalen Verarbeitungszeit zu erbringen. Hierbei sind Randbedingungen einzuhalten. Im Gegensatz zu den *Job-Shop-Problemen* sind für die *Flow-Shop-Probleme* die Abfolge der Teilaufgaben einer Aufgabe bekannt. Die Idealisierungen der Problemstellungen erlauben Komplexitätsaussagen und dienen der Suche nach geeigneten Heuristiken.

Seit Anfang der 80er Jahre werden Methoden der künstlichen Intelligenz (KI) eingesetzt, um die Lösung praktischer Ablaufplanungsprobleme vor allem durch neue Modellierungs- und Problemlösungstechniken zu unterstützen. Erste Ansätze folgten der Grundidee, allgemeingültige Prinzipien zu entwickeln, mit denen sich beliebige Problemstellungen lösen lassen, wie z.B. der *General Problem Solver (GPS)* von Newell et al. [1960]²⁶. Angewendet auf einfach strukturierte Aufgaben, z.B. Spiele oder das Bewegen von Klötzchen, erwies sich der Ansatz als leistungsfähig.

²⁴ [Homann, 1999, Instrumente der Terminsteuerung, S. 240 ff.]

²⁵ [Carlier und Pinson, 1989, An algorithm for solving the job-shop problem], [Hoitomt et al., 1993, A practical approach to job-shop scheduling problems]

²⁶ [Newell et al., 1960, Report on a general problem-solving program for a computer]

Komplexe Problemstellungen ließen sich aber nicht allgemeingültig lösen. Kurbel [1992]²⁷ umschreibt diese Systeme als *allgemeine Problemlösungsprogramme* und kommt zu dem Schluss:

- „Eine wichtige Form des Problemlösens ist das zielgerichtete Durchlaufen eines Suchbaums.
- Die Entwicklung eines allgemeinen Problemlösungsprogramms ist zu schwierig und ein fruchtloses Unterfangen, denn je mehr Problemklassen ein einzelnes Programm bewältigt, desto schwächer ist sein Verhalten bei einem ganz bestimmten Problem.“

Die Erfahrungen mit allgemeinen Problemlösungsprogrammen führte zu einer Spezialisierung. Anstatt Problemstellungen allgemeingültig zu lösen, wurden Systeme entwickelt, die zwar in ihrer Methodik allgemeingültig waren, aber immer stärker das Anwendungsgebiet in den Mittelpunkt rückten.

Besondere Aufmerksamkeit wurde in die Entwicklung wissensbasierter Systeme investiert. Das Prinzip der wissensbasierten Systeme besteht darin, Fachwissen in einer formal beschriebenen Repräsentationsform in einer Wissensbasis zu sammeln und diese mit Hilfe von Methoden zur Lösung oder Bewertung bestimmter Problemstellungen auszuwerten. Besteht die Wissensbasis aus Erfahrungswissen in Form von Fakten, in der für einen gegebenen Fall durch Ähnlichkeitsvergleiche aus vorhandenem Wissen geschöpft wird, so handelt es sich um fallbasierte Systeme bzw. die Methode des fallbasierten Schließens (*Case-Based-Reasoning (CBR)*)²⁸. Enthält die Wissensbasis Regelwissen in der Form: *Wenn . . . dann . . .*, so handelt es sich um ein regelbasiertes System (*Business Rule Management Systeme (BRMS)*). Sind die Systeme in der Lage, Schlussfolgerungen zu ziehen, die ihrerseits Teil der Wissensbasis werden, so handelt es sich um Expertensysteme²⁹. Eine Einordnung von wissensbasierten Systemen liefert Waterman [1985]³⁰.

Nachfolgend werden Ansätze aufgeführt, die wissensbasierte Systeme einsetzen, um aus Bauwerksbeschreibungen auf Bauabläufe zu schließen. Es stehen diejenigen Ansätze im Mittelpunkt der Betrachtungen, die die Bestimmung der Abhängigkeiten zwischen Vorgängen adressieren.

²⁷ [Kurbel, 1992, Kapitel 1: Künstliche Intelligenz, S. 1 ff.]

²⁸ [Richter, 2003, Kapitel 11: Fallbasiertes Schließen, S. 407 ff.]

²⁹ [Kurbel, 1992, Kapitel 3: Entwicklung von Expertensystemen, S. 65 ff.]

³⁰ [Waterman, 1985, A guide to expert systems]

Fox et al. [1983]³¹ beschreiben das Programmsystem *ISIS*, ein Programmsystem zur kosten- und ressourcenoptimierten Disposition von Produktionsabläufen aus der Sicht eines Unternehmens. Fachwissen steht in Form von Operationen (Teilabläufen) und Bedingungen (engl. constraints) zur Verfügung. Aus den Eingangsgrößen einer konkreten Problemstellung wird ein Suchbaum generiert, dessen Pfade unterschiedliche Varianten zulässiger Abläufe repräsentieren. Dieser Baum wird methodisch nach Lösungen durchsucht. Constraints werden einerseits als Regeln für die Bestimmung von Varianten und andererseits als Kriterien für die Steuerung der Suche genutzt. Die Suche berücksichtigt u. a. Prioritäten, Ressourcen, Kosten, Maschinen und Auslastung. Um die Komplexität zu beherrschen, werden bei der Suche Heuristiken eingesetzt.

Hendrickson et al. [1987]³² stellen das Expertensystem *CONSTRUCTION PLANEX* vor, das aus der Strukturierung eines Projektgegenstands auf seinen Ablauf schließt. Der Projektgegenstand wird hierarchisch nach *MasterFormat*³³ strukturiert. Den Strukturierungsknoten sind entsprechend ihres Detaillierungsgrads Vorgänge zugeordnet. Regeln legen fest, welche Abhängigkeiten beim Übergang der Strukturierung von grob nach fein zwischen den Vorgängen der nächsten Ebene einzuhalten sind. Ebenso existieren Regeln, die den Zusammenhang zwischen gröber und feiner strukturierten Komponenten beschreiben. Für ein konkretes Projekt lässt sich auf der Grundlage der Strukturierung für verschiedene Detaillierungsgrade die Menge der Vorgänge und die Menge ihrer Abhängigkeiten bestimmen.

Chevallier und Russell [2001]³⁴ präsentieren ein Expertensystem, das die Erfahrungen aus abgeschlossenen Projekten in Vorlagen für anstehende Projekte überführt und bereitstellt. Projekterfahrungen werden in physikalische Komponenten - was wird hergestellt -, Prozesskomponenten - wie wird etwas hergestellt - und in skalierbare Verknüpfungsregeln aufgespaltet. Aus einer Projektbeschreibung für ein anstehendes Projekt wird ein Vorschlag für den Ablauf generiert.

Sowohl *ISIS* als auch *CONSTRUCTION PLANEX* sind sehr auf ihr jeweiliges Anwendungsgebiet spezialisiert. Weicht ein Anwendungsfall geringfügig von dem vorgesehenen Anwendungsgebiet ab, sind umfang-

³¹ [Fox et al., 1983, *ISIS: a constraint-directed reasoning approach to job shop scheduling*]

³² [Hendrickson et al., 1987, *Expert System for Construction Planning*]

³³ [MasterFormat, 2004, Amerikanische Norm zur Gliederung von Ausschreibungsunterlagen im Bauwesen]

³⁴ [Chevallier und Russell, 2001, *Developing a draft schedule using templates and rules*]

reiche Anpassungen in der Wissensbasis erforderlich. Um größere Allgemeingültigkeit zu erlangen, sind Ansätze erforderlich, die nicht auf Vorgangsebene Schlussfolgerungen vornehmen, sondern größere Zusammenhänge berücksichtigen. Wissen wird nicht fallbasiert, sondern modellbasiert repräsentiert. Entsprechend wird die Methodik, die kausale Zusammenhänge bei der Auswahl von Vorgängen und ihren Abhängigkeiten berücksichtigt, als modellbasiertes Schließen (*Model-Based-Reasoning (MBR)*) beschrieben.

Navinchandra et al. [1988]³⁵ beschreiben das Expertensystem *GHOST*, das auf der Grundlage des modellbasierten Schließens arbeitet. Ausgehend von einer Beschreibung des Projekts durch grobe Vorgänge, die alle parallel ausgeführt werden können, erfolgt eine hierarchische Verfeinerung. Die Wissensbasis enthält Regelsätze, um Vorgänge in Teilvorgänge und ihre Abhängigkeiten zu zerlegen. Die Regelsätze werden *critics* genannt. Sie führen zu Anordnungsbeziehungen. Beim Zerlegen können Widersprüche entstehen, weil Teilvorgänge, die aus unterschiedlichen Vorgängen abgeleitet wurden, nicht gleichzeitig stattfinden können. Die Wissensbasis enthält daher darüber hinaus auch Regelsätze, um Widersprüche aufzulösen. Ein Kontrollmechanismus steuert die Reihenfolge, in der die *critics* angewendet werden. Das Ergebnis ist ein Netzplan, der einen konfliktfreien Ablauf sicherstellt. Ressourcen werden nicht berücksichtigt.

Dzeng und Tommelein [1993]³⁶ beschreiben ein weiteres modellbasiertes Verfahren, das Gebäude-Informationen-Modelle als Eingangsgrößen nutzt, für deren Bestandteile aus einer Wissensbasis, die aufbereitete Prozessinformationen enthält, geeignete Teilabläufe ermittelt werden, die zu einem Gesamtablauf zusammengeführt werden.

Darwiche et al. [1988]³⁷ beschreiben *OARPLAN*. *OARPLAN* ist Teil einer integrierten Entwurfs- und Konstruktionsumgebung. CAD-Daten werden in ein objektorientiertes Produktmodell eingelesen. Für das Produktmodell wird ein Herstellungsablauf generiert. Topologische Zusammenhänge und Informationen zu Komponenten werden aus dem Produktmodell ermittelt. Änderungen am Produktmodell wirken sich konsistent auf den Ablauf aus. Ähnlich wie bei *GHOST* erfolgt die Entwicklung eines Ablaufplans von grob nach fein. Jeder Vorgang stellt ein Tripel dar, bestehend aus *Vorgang*, *Gegenstand* und *Ressource*. Ausgehend von gro-

³⁵ [Navinchandra et al., 1988, *GHOST: Project Network Generator*]

³⁶ [Dzeng und Tommelein, 1993, *Using Product Models to Plan Construction*], [Dzeng, 1997, *Boiler Erection Scheduling Using Product Models and Case-Based Reasoning*]

³⁷ [Darwiche et al., 1988, *OARPLAN: generating project plans by reasoning about objects, actions and resources*], [Winstanley, 1995, *A Planning System for Large-Scale Industrial Projects*]

ben Vorgängen, die parallel ausgeführt werden können, wird schrittweise verfeinert. Die Verfeinerung erfolgt auf der Grundlage unterschiedlicher Wissensbasen (knowledge sources KS). Die Auswertung ist abgeschlossen, sobald sich keine weitere Wissensbasis mehr findet, die zu Änderungen des Ablaufplans führt. Das schrittweise Vorgehen erlaubt die Darstellung des Ablaufs auf verschiedenen Detaillierungsstufen.

Die bisher beschriebenen Programmsysteme zeichnen sich durch die Automatisierung bei der Bestimmung von Bauabläufen aus. Gleichzeitig stellt die Automatisierung einen Nachteil dar. Die Verfahrensauswahl wird durch die jeweils zugrunde liegende Wissensbasis vorgenommen und kann nur durch ihre Modifikation beeinflusst werden. Die jeweiligen Eingangsgrößen werden direkt in Abläufe transformiert. Diejenigen Verfahren, die Heuristiken einsetzen, können nicht sicherstellen, dass Ergebnisse reproduzierbar sind.

Fischer und Aalami [1996]³⁸ präsentieren ein modellbasiertes System (MBR), das den Aspekt des Planens und des Gestaltens stärker berücksichtigt. Die Transformation von Bauwerksmodellen auf Bauabläufe erfolgt halbautomatisch. Die Vorgehensweise ist ähnlich wie bei *GHOST* und *OARPLAN*. Vorgänge werden hierarchisch verfeinert. Für jede Hierarchiestufe wird auf der Grundlage der Wissensbasis die Menge der Abhängigkeiten ermittelt. Ein wesentlicher Unterschied besteht in dem Verfeinerungsmechanismus. Die Wissensbasis wird genutzt, um Vorschläge für die Verfeinerung, im Sinne einer Verfahrensauswahl, zu ermitteln. Die Entscheidung für ein konkretes Verfahren wird vom Nutzer verlangt. Dieses Vorgehen erlaubt es, spezifischen Nutzeranforderungen gerecht zu werden. Im Gegensatz zu den bisher vorgestellten Ansätzen ist dieses Verfahren in der Lage, gezielt die Verfahrenswahl zu beeinflussen. Unterschiedliche Abläufe lassen sich für dasselbe Bauwerksmodell entwickeln. Im Rahmen der Ablaufplanung lassen sich unterschiedliche Bauverfahren auf ihre Wirtschaftlichkeit hin untersuchen und ihre Auswirkungen auf Dauern und Termine bestimmen.

In einem Vergleich unterschiedlicher Ansätze zum Generieren von Bauabläufen kommt Levitt [1990]³⁹ zu dem Schluss, dass allgemeine Problemlösungsprogramme aufgrund ihrer Beschränkung in der Beschreibung von Vorgängen keine Bedeutung für komplexe Planungsaufgaben haben. Ebenso sieht er im Einsatz von spezialisierten Expertensystemen Grenzen, da eine Anpassung der Wissensbasis von fallbasierten Systeme-

³⁸ [Fischer und Aalami, 1996, Scheduling with Computer-Interpretable Construction Method Models], [Kuhne et al., 2000, Neue Ansätze zur Projektplanung und Baustellensteuerung]

³⁹ [Levitt, 1990, Knowledge-based planning systems: An engineering perspective]

me nur mit erheblichem Aufwand realisiert werden kann. Das größte Potential sieht Levitt in modellbasierten Systemen (MBR), die einen Mittelweg zwischen allgemeinen Problemlösungsprogrammen und spezialisierten Expertensystemen darstellen.

2.4 Zusammenfassung

Ablaufplanungen werden im Rahmen eines Projekts durch unterschiedliche Beteiligte in unterschiedlichen Projektphasen und mit unterschiedlichen Zielsetzungen erstellt. Mit dem Voranschreiten der Projektphasen steigt der Detaillierungsgrad. Hierbei werden nicht die vorhandenen Modelle sukzessiv verfeinert, sondern es entstehen unterschiedliche Ablaufplanungen, denen - entsprechend der jeweiligen Anforderungen - unterschiedliche Methodiken zugrunde liegen, angepasst an den angestrebten Detaillierungsgrad und die jeweilige Sichtweise. Nach dem *bottom-up* Prinzip - von fein nach grob - ist sicherzustellen, dass die Vorgaben aus übergeordneten Planungen eingehalten werden.

Die Ablaufplanung ist ein dynamischer Prozess. Mehrere Iterationszyklen sind erforderlich, um unterschiedliche Anforderungen zu synchronisieren und Änderungen, die sich erst im Verlauf ergeben, zu berücksichtigen. Zusätzlich zu berücksichtigende Anforderungen und die Anpassung durch Änderungen stellen aber keine Verfeinerung der Ablaufplanung dar, sondern eine Konsolidierung eines Ablaufmodells innerhalb eines betrachteten Detaillierungsgrads. Geeignete Versionierungsansätze sind erforderlich, um ein Fortschreiben eines Ablaufplans zu ermöglichen.

Insbesondere bei gleichartigen oder ähnlichen Bauprojekten können wissensbasierte Systeme zweckmäßig genutzt werden. Sie stoßen jedoch an ihre Grenzen, wenn das aktuelle Projekt über zu viele Besonderheiten verfügt. Für nicht vollständig beschriebene Problembereiche lässt sich keine Lösung bzw. keine korrekte Lösung generieren. Die Vollständigkeit lässt sich nicht formal überprüfen. Bei Anpassungen der Wissensbasis steht wiederum der Mensch vor der Herausforderung, die Komplexität zu beherrschen und korrekte und konsistente Erweiterungen vorzunehmen.

Kapitel 3

Prozessorientiertes Modell für die Bauablaufplanung

Voraussetzung für einen erfolgreichen Projektablauf bei stetig steigenden Anforderungen ist eine detaillierte Kenntnis über die projektspezifischen Abläufe von der Planung über die Ausführung bis hin in den Betrieb. Kurze Realisierungsphasen können nur erreicht werden, wenn die Planung und die Ausführung sich einander so überlappen, dass sie über weite Teile parallel erfolgen können. Die Komplexität, die Bauabläufe auszeichnet, kann nicht mehr allein aus der Anschauung heraus bewältigt werden, so dass Anstrengungen unternommen werden, moderne rechnergestützte Verfahren für das Projektmanagement zu entwickeln.

In diesem Kapitel wird ein formaler Ansatz beschrieben, der ausgerichtet ist auf das formale Gestalten und Entwickeln von Prozessmodellen. Ziel des Ansatzes ist es, methodisch abgeleitete Prozessmodelle für die Netzplantechnik bereitzustellen, die die technologisch zwingenden Abhängigkeiten¹ vollständig enthalten. Das Ergebnis der Modellierung stellt ein Rohgerüst dar, das durch den erfahrenen Planungsingenieur zu überprüfen und um weitere Angaben zu ergänzen ist.

Abschnitt 3.2 geht auf die zugrunde liegende, formale Methodik zum Modellieren von Prozessen ein. Das von Keller, Nüttgens und Scheer² entwickelte Konzept zur Modellierung von Geschäftsprozessen wird aufgegriffen und zur Abbildung von Bauprozessen verwendet. Begriffe aus dem Bereich der Geschäftsprozessmodellierung werden eingeführt.

Abschnitt 3.3 definiert Fachbegriffe, die spezifisch sind für den hier

¹ siehe Abschnitt 2.2.4

² [Keller et al., 1992, Semantische Prozeßmodellierung auf der Grundlage ereignisgesteuerte Prozesskette (EPK)]

beschriebenen Modellierungsansatz. Die Terminologie unterscheidet sich von derjenigen, die Huhnt und Lawrence [2004]³ für Planungsprozesse und Huhnt [2005]⁴ bzw. Huhnt und Enge [2006]⁵ für Ausführungsprozesse verwenden. In dem hier beschriebenen Ansatz wird versucht, Fachbegriffe zu verwenden, die unabhängig sind von der Planungsphase. Darüber hinaus unterscheidet sich der hier beschriebene Ansatz in der Modellierung von Vorgängen. Abschnitt 3.4 beschreibt das Datenmodell und die Vorgehensweise zum methodischen Gestalten und Modellieren von Prozessmodellen für die Netzplantechnik. Das Kapitel endet mit einer Diskussion des Modellierungsansatzes.

3.1 Mathematischer Hintergrund

Prozesse lassen sich formal beschreiben durch Mengen und Verknüpfungen von Elementen der Mengen. Verknüpfungen stellen Tupel dar, die nach festgelegten Regeln Zusammenhänge zwischen Elementen der Mengen erfassen. Gleichartige Verknüpfungen werden in Relationen zusammengefasst. Relationen stellen somit ebenfalls Mengen dar, für die festgelegte Bildungsregeln definiert sind. In der Mathematik sind Mengen und die Regeln der Mengenbildung in der Mengenalgebra definiert. Die Relationenalgebra befasst sich mit den Besonderheiten bei der Mengenbildung und der Definition von Operationen auf Relationen als spezielle Mengen. Die Veranschaulichung von Zusammenhängen, die auf der Grundlage von Mengen und Relationen formal beschrieben sind, führt zu Knoten-Kanten-Diagrammen, wobei die Elemente der Mengen als Knoten und die Elemente der Relationen als Kanten dargestellt werden, die in der Mathematik als Graphen bezeichnet werden. Die Graphentheorie ist ein Teilgebiet der Mathematik, das sich auf der Grundlage der Mengen- und Relationenalgebra mit der Beschreibung der Eigenschaften von Graphen befasst und Operationen auf Graphen definiert. Für endliche Graphen existieren Matrixalgebren, die einen Rahmen bilden, um Graphen auf formaler Grundlage zu behandeln.

Für die Begriffe und Definitionen haben sich in der Graphentheorie noch keine einheitliche Bezeichnungsweise durchgesetzt. In Anhang A sind Begriffe und Definitionen zusammengefasst, um ein einheitliches

³ [Huhnt und Lawrence, 2004, Methodik zur Modellierung konsistenter Soll-Vorgaben für komplexe Planungsprozesse]

⁴ [Huhnt, 2005, Generating Sequences of Construction Tasks]

⁵ [Huhnt und Enge, 2006, Can algorithms support the specification of construction schedules?]

Verständnis zu schaffen. Der Betrachtungsweise liegen die mathematischen Grundlagen von Pahl und Damrath [2000] zugrunde.

3.2 Prozesse: Modellierungsmethodik

Für die formale Beschreibung von Prozessen existiert eine große Zahl an unterschiedlichen Ansätzen. Gadatsch [2007]⁶ liefert für die Modellierung von Geschäftsprozessen einen Überblick und richtet dabei den Fokus auf die Gestaltung und die Dokumentation von Prozessen. Er unterscheidet die formalen Methoden zur Modellierung von Prozessen in skriptbasierte Methoden, die als Skriptsprachen bezeichnet werden, und grafische Methoden, die als Diagrammsprachen bezeichnet werden. Skriptsprachen nutzen zur Beschreibung von Prozessmodellen eine textbasierte Notation. Das Ergebnis sind codeartige Textdateien, die zwar das Modell sehr präzise beschreiben, jedoch eine anschauliche Betrachtung nicht erlauben. Im Gegensatz dazu stehen Diagrammsprachen. Sie erlauben die Modellierung von Prozessen mit Hilfe von graphischen Beschreibungselementen, die ein intuitives Verständnis unterstützen.

In dieser Arbeit wird die graphische Notation der Architektur integrierter Informationssysteme (ARIS)⁷ genutzt. Bei ARIS handelt es sich zum einen um ein Konzept und zum anderen um ein leistungsfähiges Softwareprodukt. An dieser Stelle sei das Konzept angesprochen.

ARIS: Um der Komplexität zu begegnen, werden unterschiedliche Sichten definiert, die jeweils unterschiedliche Aspekte des Prozessmodells abbilden. ARIS sieht eine Zerlegung des Prozessmodells in fünf Sichten vor. Die statischen Beschreibungsobjekte von Geschäftsprozessen werden in der *Organisationssicht*, der *Datensicht*, der *Funktionssicht* und der *Leistungssicht* abgebildet. Die dynamischen Aspekte werden in der *Steuerungssicht* zusammengefasst. Ihr liegt die von Keller, Nüttgens und Scheer⁸ entwickelte ereignisgesteuerte Prozesskette (EPK) zugrunde, die zur Gruppe der kontrollflussorientierten Ansätze⁹ gezählt wird. Im Folgenden werden diejenigen Beschreibungselemente und Sichten aufgegriffen, die im Rahmen dieser Arbeit verwendet werden.

⁶ [Gadatsch, 2007, Methoden der Prozessmodellierung, S. 80 f.]

⁷ [Scheer, 1998, ARIS. Vom Geschäftsprozeß zum Anwendungssystem], [Scheer, 2001, ARIS-Modellierungsmethoden, Metamodelle, Anwendungen]

⁸ [Keller et al., 1992, Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK)]

⁹ [Gadatsch, 2007, Übersicht über ausgewählte Diagrammsprachen, S. 81 ff.]

Datensicht: Die Datensicht in ARIS dient der Beschreibung von Datenstrukturen. Das zugrunde liegende Modell ist ein Entity-Relationship-Model (ERM)¹⁰ bzw. ein erweitertes Entity-Relationship-Model (eERM). Die Modelle unterscheiden sich im Umfang der definierten Elemente. Abbildung 3.1 zeigt diejenigen Elemente aus einem eERM, die im Rahmen dieser Arbeit verwendet werden.

Die Abbildung von Objekten aus der realen Welt auf Objekte im Modell erfolgt auf der Grundlage von Entitäten. Eine Entität stellt eine abstrakte Beschreibung von Objekten dar, die zu derselben Klasse gehören. Die Objekte selbst sind Instanzen einer Entität. Existiert eine Abhängigkeit zwischen Instanzen, so werden diese auf der Ebene von Entitäten über Beziehungen modelliert. In der Darstellung wird eine Beziehung über das Beziehungssymbol und den an ihr anschließenden Kanten zu den Entitäten modelliert. Die Angabe von Kardinalitäten drückt aus, wie viele Instanzen einer Entität mit wie vielen Instanzen einer anderen Entität verknüpft sind. Die Art der Beziehung wird durch die Angabe einer Semantik beschrieben. Für die Angabe von Eigenschaften werden Attribute spezifiziert. Das Konzept der Vererbung erlaubt die Erweiterung von Entitäten.

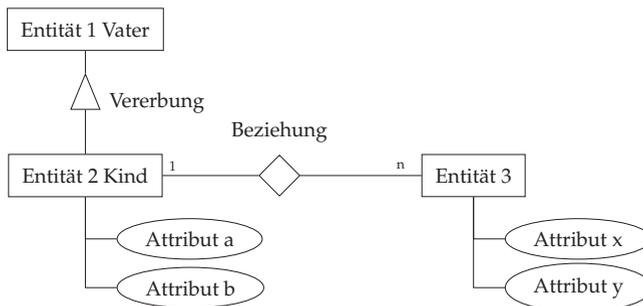


Abbildung 3.1: Elemente einer Datensicht (eERM)

Steuerungssicht: Die Steuerungssicht in ARIS dient der Beschreibung von ablaufbezogenen Zusammenhängen. Die Menge der Beschreibungselemente wird um die Beschreibungselemente *Funktion* und *Ereignis* erweitert sowie um Verknüpfungselemente für Funktionen und Ereignisse,

¹⁰ [Seidlmeier, 2006, Die Datensicht, S. 17 f.]

die eine Modellierung von Abläufen erlauben. Abbildung 3.2 zeigt diejenigen Elemente, die im Rahmen dieser Arbeit verwendet werden.

Funktionen sind Aufgaben bzw. Tätigkeiten, die an einem Objekt ausgeführt werden und der Erbringung einer Leistung in einem Geschäftsprozess dienen. Sie werden durch Ereignisse ausgelöst und enden mit dem Erreichen eines definierten Ergebnisses, das sich ebenfalls durch ein Ereignis beschreiben lässt. Das Modell zur Beschreibung der Steuerungssicht in ARIS, dem die oben genannten Elemente zugrunde liegen, wird als erweiterte ereignisgesteuerte Prozesskette (eEPK)¹¹ bezeichnet.

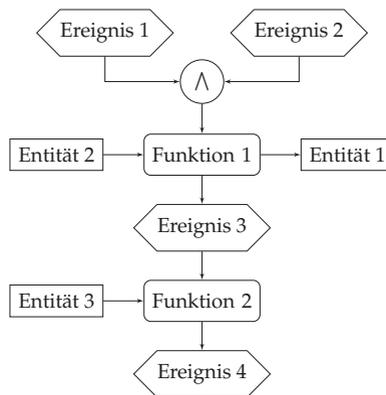


Abbildung 3.2: Elemente einer Prozesskette (eEPK)

Das Konzept, ablaufbezogene Zusammenhänge durch Funktionen und Ereignisse zu beschreiben, erlaubt es, Geschäftsprozesse in getrennten Modellen abzubilden. Ihr Zusammenhang wird durch Ereignisse sichergestellt. Ereignisse, die im Verlauf eines Geschäftsprozesses als das Ergebnis einer Funktion auftreten, können für andere Geschäftsprozesse das auslösende Ereignis darstellen.

Wegen der Analogie, die sowohl zwischen Vorgängen eines Bauablaufs und Funktionen eines Geschäftsprozesses bestehen, als auch zwischen Zuständen von Planungsunterlagen bzw. Bauteilen und Ereignissen, wird im Rahmen dieser Arbeit das Modellierungskonzept von ARIS verwendet, um Bauabläufe formal zu beschreiben.

¹¹ [Seidlmeier, 2006, Steuerungs- (Prozess-)sicht, S. 20 f.]

3.3 Bauablaufplanung: Terminologie und Datenmodell

In diesem Abschnitt werden Begriffe eingeführt, die der fachlichen Beschreibung des abzubildenden Realitätsausschnitts dienen. Die Begriffe werden erläutert und auf die Beschreibungselemente der angewendeten Modellierungsmethodik bezogen.

Abbildung 3.3 zeigt ein einfaches Tragwerk, für das eine Einteilung in die Bauteile *StB-Fundament 1*, *StB-Fundament 2*, *KS-Wand 1*, *KS-Wand 2* und *StB-Decke 1* vorgenommen wurde. Es stellt ein stark vereinfachtes Projekt dar, an dem im Folgenden grundlegende Aspekte erläutert werden.

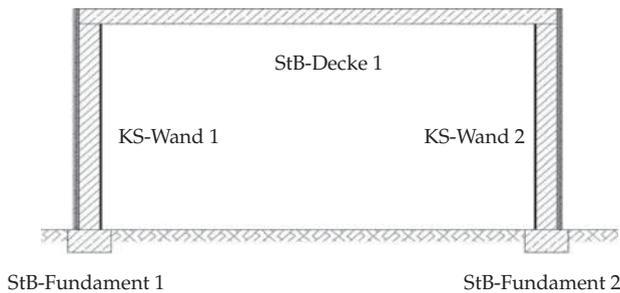


Abbildung 3.3: Projekt Garage

3.3.1 Projektunabhängige und -spezifische Informationen

Grundlage für das Modellierungskonzept ist eine Einteilung des Modellierungsgegenstands in *projektunabhängige* und in *projektspezifische Informationen*. Projektunabhängige Informationen können als Vorlagen angesehen werden, die durch projektspezifische Informationen zu konkretisieren sind.

Betrachtet man ein Modell, das einen Prozess im Bauwesen beschreibt, so besteht dieses aus Informationen und Verknüpfungen der Informationen, die projektspezifisch bedingt sind und solchen, deren Gültigkeit auch unabhängig vom Projekt gegeben sind. Beispielsweise lässt sich der Ablauf für die Entwicklung einer Statik unabhängig von einem konkreten

Projekt isoliert betrachten und durch einen üblichen Ablauf beschreiben. Ebenso lässt sich der Herstellungsprozess einer typischen Trockenbauwand im Vorfeld beschreiben, ohne den konkreten Kontext im Projekt zu kennen.

Abbildung 3.4 zeigt die Entitäten *<Element>vorlage* und *<Element>*. *<Element>* steht als Platzhalter für einen bestimmten Typ. Projektunabhängige Informationen werden in Entitäten *<Element>vorlage* beschrieben und projektspezifische Informationen in Entitäten *<Element>*. Jeder Instanz von *<Element>* ist seine Vorlage bekannt. Jeder Instanz von *<Element>vorlage* können mehrere Instanzen einer Entität *<Element>* zugeordnet sein.

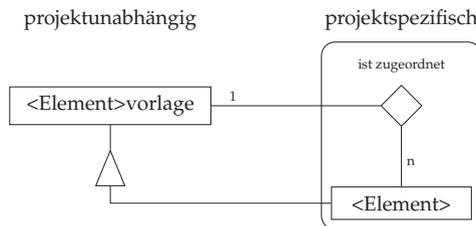


Abbildung 3.4: Projektunabhängige und -spezifische Informationen

Anhand des in Abbildung 3.3 gezeigten Projekts *Garage* sollen in Abbildung 3.5 die Begriffe *Entität*, *Instanz*, *<Element>vorlage* und *<Element>* beispielhaft in einen Zusammenhang gebracht werden.

StB-Fundament 1, *StB-Fundament 2*, *KS-Wand 1*, *KS-Wand 2* und *StB-Decke 1* stellen Instanzen der Entität *Bauteil* dar. Diese sind projektspezifisch. Sowohl *StB-Fundament 1* und *StB-Fundament 2* als auch *KS-Wand 1* und *KS-Wand 2* sind Bauteile, für die bestimmte Merkmale gleich sind. Um den Modellierungsaufwand zu reduzieren, werden in einem ersten Schritt die Instanzen *StB-Fundament*, *KS-Wand* und *StB-Decke* der Entität *Bauteilvorlage* modelliert. Sie legen nur die allgemeinen Merkmale wie Schichtenaufbau und Qualitätsangaben fest. In einem zweiten Schritt folgt das Anlegen der Bauteile. Sie werden als Instanzen der Entität *Bauteil* modelliert. Durch Zuordnen der Vorlagen zu den Bauteilen können die bereits spezifizierten Merkmale automatisch übertragen werden. Die Merkmale werden in entsprechende Attribute der Entität *Bauteil* kopiert. Ein Kopieren der Werte erlaubt die nachträgliche Bearbeitung auf Instanz-

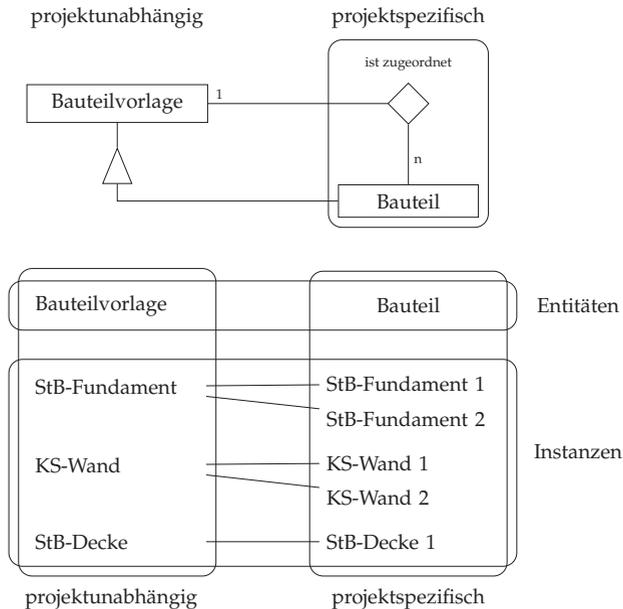


Abbildung 3.5: Zusammenhang der Informationen am Beispiel Garage

ebene, ohne Änderung bei denjenigen Instanzen hervorzurufen, die von derselben Vorlage abgeleitet wurden.

3.3.2 Strukturierung von Informationen

Die Komplexität bei der Modellierung von Prozessen resultiert im Wesentlichen aus der Bewältigung von Massendaten. Vor diesem Hintergrund wird konsequent der Ansatz verfolgt, Instanzen gleicher Entität, also Elemente einer Menge, grundsätzlich durch geeignete Bäume zu strukturieren.

Die Strukturierung einer Menge erfolgt durch die Abbildung ihrer Elemente auf die Knoten eines Strukturierungsbaums. Liegt eine Abbildung¹² $f : M \rightarrow S$ der Elemente einer Menge M auf die Menge S der Strukturierungsknoten eines Baums vor, so definiert ein Strukturierungsknoten $s_i \in S$ eine Teilmenge T_{s_i} aus der Menge M . Abbildung 3.6

¹² Siehe Anhang: A Begriffe aus der Graphentheorie \uparrow Abbildung

zeigt das grundsätzliche Vorgehen zur Strukturierung von $\langle \text{Element} \rangle$ -Instanzen. Die Notation $\langle \text{Baum} \rangle \text{Knoten}$ weist darauf hin, dass es mehrere Baumstrukturen gibt. Einem bestimmten $\langle \text{Baum} \rangle$ kann eine $\langle \text{Element} \rangle$ -Instanz nur einmal zugeordnet werden.

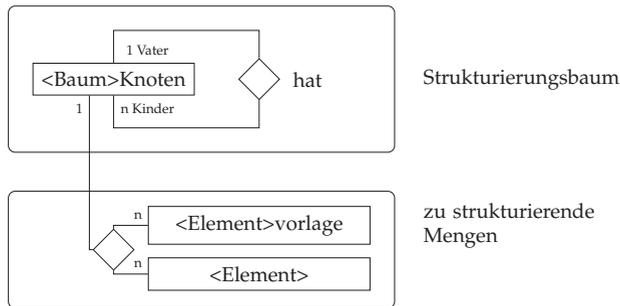


Abbildung 3.6: Strukturieren von Elementen

Eine detaillierte Beschreibung der Anwendung von Strukturierungen auf Informationsmengen wird in einem Beitrag von Huhnt und Enge [2007]¹³ aufgezeigt. Der Beitrag beschreibt, wie die Strukturierung einer Menge M_A auf eine Menge M_B angewendet werden kann, sofern es eine oder auch mehrere Relationen gibt, über die die Elemente der Menge M_A direkt oder indirekt mit den Elementen der Menge M_B verknüpft sind. Dieses Vorgehen erlaubt es, nicht jede Menge einzeln zu strukturieren, sondern nur charakteristische Mengen. Ihre Strukturierung kann auf verknüpfte Mengen übertragen werden.

Die Eigenschaften der Abbildung f entscheiden über die Eigenschaften der Teilmengen T_{s_j} . Stellt die Abbildung eine Äquivalenzrelation¹⁴ dar, dann lässt sich die Menge M auf der Grundlage von S in disjunkte Teilmengen zerlegen. Ist eine Menge über eine Äquivalenzrelation auf Strukturierungsknoten abgebildet, dann ist sichergestellt, dass jedes Element $a \in M$ einem Knoten $s \in S$ zugeordnet ist und dass dasselbe Element a nicht mehrfach zugeordnet werden kann. Ein Strukturierungsbaum, der auf dieser Grundlage Elemente verwaltet, liefert für Strukturierungsknoten, die auf unterschiedlichen Zweigen liegen, disjunkte Teilmengen und

¹³ [Huhnt und Enge, 2007, Consistent Information Management for Structuring Construction Activities]

¹⁴ Siehe Anhang: A Begriffe aus der Graphentheorie \uparrow Äquivalenzrelation

jedes Element ist durch mindestens eine Teilmenge zu erreichen. Mengen können durch unterschiedliche Bäume strukturiert werden. Die Kombination unterschiedlicher Strukturierungsbäume stellt ein effizientes Mittel dar, um gezielt Teilmengen zu bilden.

3.3.3 eERM Leistungsgegenstand und -vorlage

Ein *Leistungsgegenstand* beschreibt das Ergebnis, das sich aus einer Summe an gezielten Handlungen, die nach einer festgelegten Ordnung zu erbringen sind, erzielen lässt. Im Verlauf eines Prozesses werden Leistungen in Form von Informations- bzw. Materialtransformationen erbracht. Bezogen auf einen Prozess im Bauwesen bedeutet die Informationstransformation beispielsweise die Erstellung oder Weiterverarbeitung von digitalen Modellen, Plänen oder Dokumenten und die Materialtransformation, die Herstellung, den Umbau oder den Rückbau von Bauteilen. Digitale Modelle, Pläne oder Dokumente stellen jeweils Teilmengen der Informationen dar, die eine zusammengehörige Einheit bilden und als *Leistungsgegenstand* im Sinne eines Erzeugnisses bzw. eines Produktes aufgefasst werden. Ebenso stellen Bauteile Leistungsgegenstände dar.

Projektgegenstände gleichen Qualität werden zu einer Menge zusammengefasst und durch Entitäten beschrieben. Diese Entitäten werden von der Entität *Leistungsgegenstand* abgeleitet. Die Attribute eines Leistungsgegenstands beschreiben projektspezifische Informationen. Dazu zählen *Mengenangaben* und *Randbedingungen*. Für Bauteile sind dies beispielsweise konkrete Abmessungen und Angaben darüber, wie viele Personen maximal in einer Flucht mauern können.

Die projektunabhängigen Informationen werden durch Entitäten beschrieben, die von der Entität *Leistungsgegenstandsvorlage* abgeleitet sind. Leistungsgegenstandsvorlagen enthalten Qualitätsangaben. Die Attribute *Qualität*, *Menge* und *Randbedingung* sind zusammengesetzte Attribute. Sie werden durch Auswerten der Attribute der Kindentitäten bestimmt. Auf die Darstellung der Attribute der Kindentitäten wird aus Gründen der Übersichtlichkeit verzichtet. Ebenfalls nicht dargestellt ist der Zusammenhang nach Abbildung 3.4, der auch für Kindentitäten gilt.

Für die vollständige Beschreibung eines Leistungsgegenstands sind alle Angaben erforderlich. Ein Leistungsgegenstand ist daher über eine Beziehung mit einer Leistungsgegenstandsvorlage verknüpft. Unterschiedliche Instanzen der Entität *Leistungsgegenstand* können mit derselben Instanz einer *Leistungsgegenstandsvorlage* verknüpft sein. Abbildung 3.7 zeigt ihren Zusammenhang.

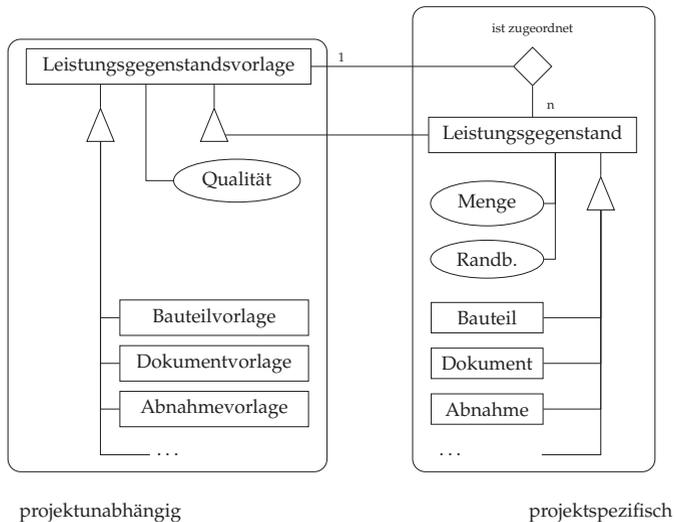


Abbildung 3.7: eERM Leistungsgegenstand und -vorlage

Zum Begriff Leistungsgegenstand: In der Regel kann Produkt oder Erzeugnis als Synonym für Leistungsgegenstand verwendet werden. Dies gilt jedoch nicht immer. Beispielsweise lässt sich das Ergebnis des Ablaufs für die Abnahme einer gebäudetechnischen Anlage nicht einem einzelnen Bauteil oder Dokument zuordnen. Die Begriffe Produkt oder Erzeugnis decken - im Gegensatz zu dem Begriff Leistungsgegenstand - das Ergebnis des Ablaufs nicht mit ab.

3.3.4 eERM Leistungsprozess und -vorlage

Einem Leistungsgegenstand lässt sich ein inhaltlich zusammenhängender Teilprozess zuordnen. Betrachtet man beispielsweise die Decke *StB-Decke 1* des Projekts *Garage* aus Abbildung 3.3, so lässt sich innerhalb des Gesamtprozesses ein Ausschnitt identifizieren, der einen Zusammenhang mit der Decke *StB-Decke 1* hat. Herausgelöst stellt dieser Ausschnitt den Herstellungsprozess der Instanz *StB-Decke 1* dar und wird als *Leistungsprozess* bezeichnet.

Ein Leistungsprozess setzt sich aus projektspezifischen Informationen und projektunabhängigen Informationen zusammen. Die Attribute *Ab-*

lauf allgemein der Entitäten *Leistungsprozessvorlage* und *Ablauf speziell* der Entität *Leistungsprozess* stellen zusammengesetzte Attribute dar. Ein Leistungsprozess umfasst Vorgänge, Zustände und eine Relation in der Menge der Zustände, die zusammen einen Ablauf beschreiben. Instanzen dieser Entitäten beziehen ihre Informationen aus den Instanzen der Kindentitäten. Für Instanzen vom Typ *Leistungsprozessvorlage* beschreibt das Attribut *Ablauf allgemein* einen Ablauf, der keinen Bezug zu projektspezifischen Informationen hat, beispielsweise den allgemeinen

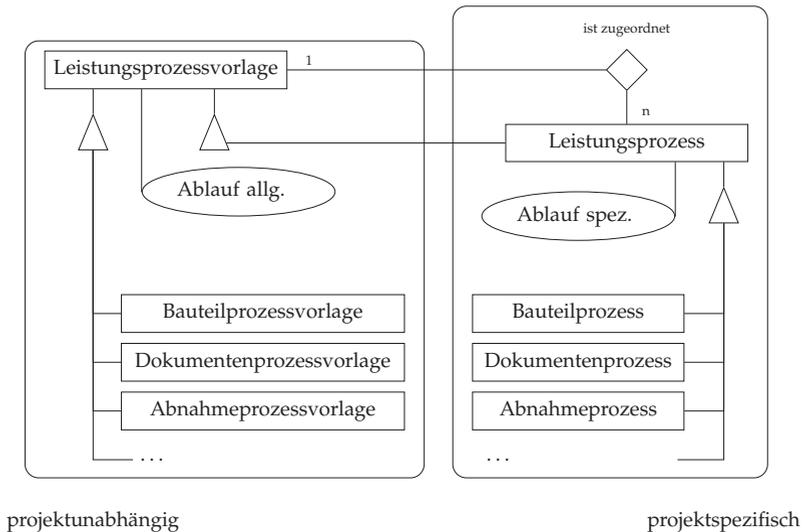


Abbildung 3.8: eERM Leistungsprozess und -vorlage

Herstellungsprozess einer *StB-Decke*. Für Instanzen von *Leistungsprozess* beschreibt das Attribut *Ablauf speziell* einen Ablauf, der für einen konkreten Leistungsgegenstand gilt, beispielsweise den Herstellungsprozess der *StB-Decke 1* des Projekts *Garage*. Mengen, Aufwände und Dauern sind bekannt. Die Beziehungen drücken aus, dass unterschiedliche projektspezifische Instanzen mit derselben projektunabhängigen Instanz verknüpft sein können. Abbildung 3.8 zeigt den Zusammenhang zwischen *Leistungsprozessvorlage* und *Leistungsprozess*. Nicht dargestellt ist der Zusammenhang der Kindentitäten nach Abbildung 3.4.

Der Leistungsgegenstand und der Leistungsprozess werden in unterschiedlichen Entitäten beschrieben. Eine konkrete Wand wird beschrieben durch einen *Leistungsgegenstand* (bzw. genauer: *Bauteil*). Der Herstellungsprozess einer Wand wird beschrieben durch einen *Leistungsprozess* (bzw. genauer: *Bauteilprozess*). Erst die Verknüpfung eines Leistungsgegenstands *KS-Wand 1* mit einem Leistungsprozess (Herstellung der *KS-Wand 1*) ermöglicht die Spezifikation ihres Herstellungsprozesses, für den echte Termine ermittelt werden können und Kosten- und Ressourcenaufwände bestimmbar sind. Abbildung 3.9 zeigt den Zusammenhang zwischen Leistungsgegenstand und Leistungsprozess aus projektunabhängiger Sicht und aus projektspezifischer Sicht. Der Zusammenhang zwischen

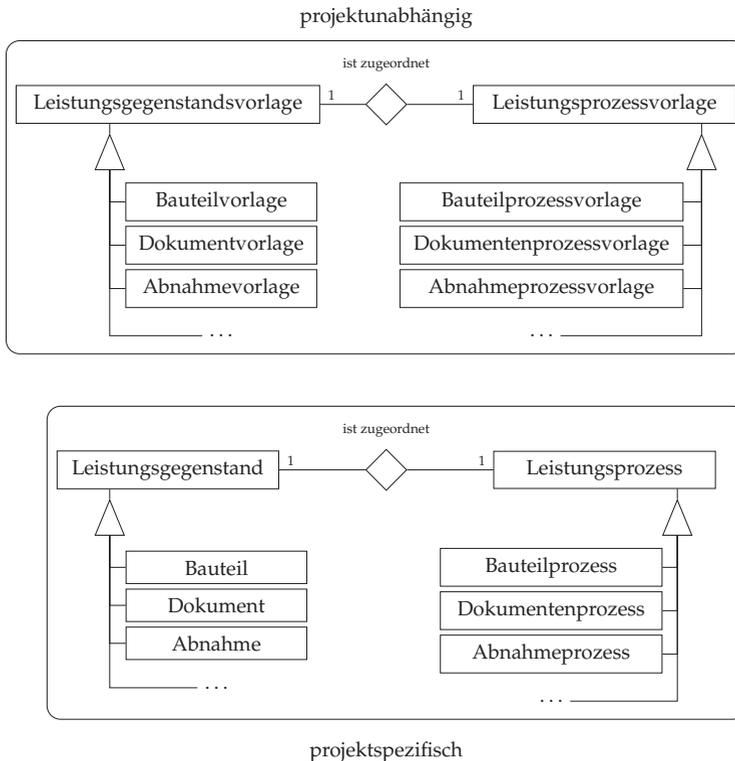


Abbildung 3.9: eERM Leistungsgegenstand und Leistungsprozess

den projektunabhängigen und den projektspezifischen Entitäten wurde bereits in den Abbildungen 3.7 und 3.8 aufgezeigt. Zwischen den Instanzen dieser Entitäten besteht eine 1:1 Beziehung.

3.3.5 eERM Vorgang und -vorlage

Ein *Vorgang* beschreibt die Transformation eines Leistungsgegenstands von einem definierten Ausgangszustand in einen definierten Endzustand. Er stellt ein zeitforderndes Geschehen dar, das eine abgeschlossene Veränderung beschreibt, die bezogen auf den Detaillierungsgrad des Prozesses nicht weiter differenziert werden muss. Er beschreibt sowohl Handlungen, die Ressourcen erfordern, als auch Zustandsänderungen, die keine Ressourcen erfordern. Dies können beispielsweise zeitfordernde Zustandsänderungen wie Abbindevorgänge, Einhalten von Lieferfristen oder die Berücksichtigung von Planvorlaufzeiten sein. Ein Vorgang, der Ressourcen erfordert, wird als *Tätigkeit* bezeichnet.

3.3.6 eERM Vorgangsposition und -vorlage

Der Umfang eines Vorgangs ergibt sich aus $1, \dots, n$ *Vorgangsposition*. Jede Vorgangsposition kapselt Informationen zur Ermittlung von Dauern, Ressourcen, Material und bzw. oder Produktionsmitteln. Als Grundlage für die Ermittlung von Dauern gehen Aufwandswerte A_w bzw. Leistungswerte L_w in die Berechnung ein. Durch Aufsummieren der Anteile aus den Vorgangspositionen lässt sich der Ressourcenbedarf, der Materialverzehr und die erforderlichen Produktionsmittel für einen Vorgang berechnen. Auf der Grundlage dieser Auswertung lassen sich die Vorgangsdauer und die Vorgangskosten verursachungsgerecht ermitteln. Abbildung 3.10 zeigt den Zusammenhang der Entitäten, die einen Vorgang beschreiben. Es gilt der in Abbildung 3.4 gezeigte Zusammenhang zwischen den projektunabhängigen und den projektspezifischen Entitäten.

Aufwands- und Leistungswerte sind für viele Gewerke in ausführlichen Tabellenwerken wie den ARH-Tabellen¹⁵ oder in der einschlägigen Literatur zur Arbeitskalkulation¹⁶ zusammengefasst. Abbildung 3.11

¹⁵ [ARH-Tabellen, 2001, Arbeitszeit-Richtwerte Hochbau (ARH). Die Gesamtausgabe enthält die Tabellen Erdarbeiten, Schalarbeiten, Rahmenschalung, Elementbauteile, Bewehrungsarbeiten, Betonarbeiten und Mauerarbeiten.], [Greiner et al., 2005, Datengrundlage, S. 145]

¹⁶ [Bauer, 2006, Berechnung der Vorgangsdauern, S. 547 ff.], [Kochendörfer et al., 2007, Berechnung von Vorgangsdauern, S. 111 ff.], [Würfele und Bielefeld, 2007, Ermitt-

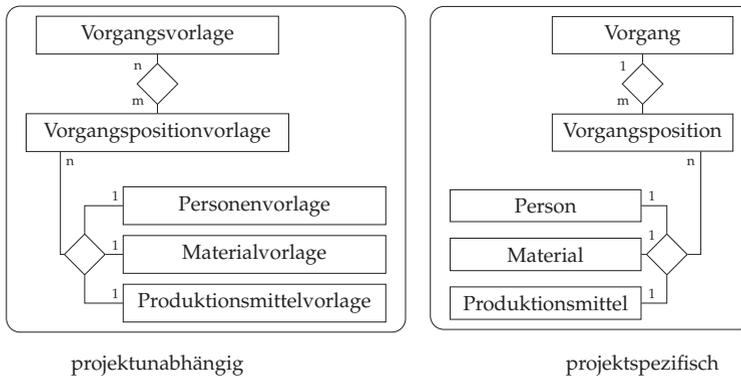


Abbildung 3.10: eERM Vorgang und -vorlage

zeigt das Flussdiagramm aus der DIN 69902 [2009]¹⁷, das den Zusammenhang und die Verknüpfung der Informationen für eine leistungswertbezogene Ermittlung von Aufwänden beschreibt. An einem Zahlenbeispiel werden für ein gegebenes, zu erzielendes Arbeitsergebnis AE und für einen gegebenen Leistungswert EML die erforderlichen Dauer in Stunden ermittelt. Die Dauer wird auf einen Ressourcenbedarf EMB umgerechnet. Ein analoges Diagramm ergibt sich bei der Verwendung von Aufwands- bzw. Zeitbedarfswerten A_w . Lediglich die Ermittlung der Arbeitsmenge AM_h unterscheidet sich. Bei Aufwands- bzw. Zeitbedarfswerten berechnet sich die Arbeitsmenge AM_h nicht aus einem Quotienten, sondern aus dem Produkt aus AE und A_w .

3.3.7 eERM Zustand, Ereignis und -vorlage

Ein *Zustand* beschreibt eine Momentaufnahme eines Leistungsgegenstands im Verlauf eines Prozesses. Er drückt aus, dass ein bestimmter Leistungsgegenstand einen charakteristischen Zwischen- oder Endstand erreicht hat. Die Verknüpfung eines Leistungsgegenstands mit einem Zustand stellt ein *Ereignis* dar. Im Gegensatz zu einem Vorgang ist ein Ereignis nicht zeitfordernd. Jedem Vorgang folgt genau ein Ereignis. Dieses

lung der Ausführungsdauern, S. 50 ff.]

¹⁷ [DIN 69902, 2009, Flussdiagramm, zur Berechnung von Einsatzmitteln und Leistungsbedarf aus Plandaten, S. 3]

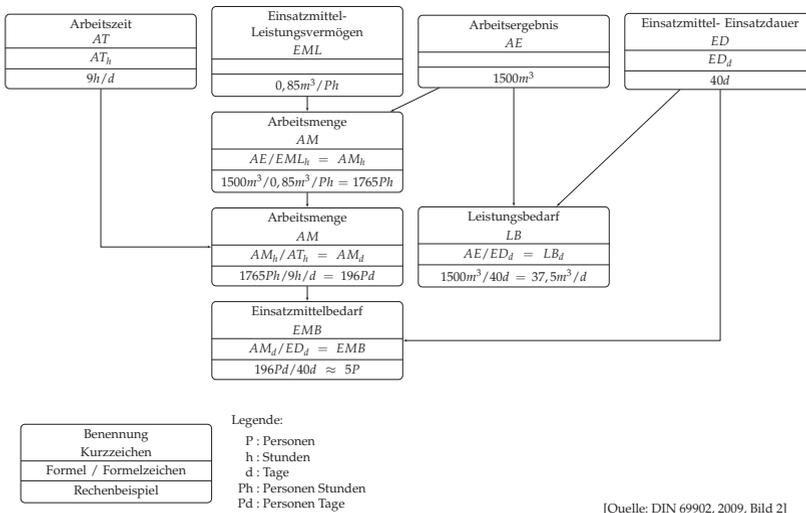


Abbildung 3.11: Flussdiagramm einer Vorgangsposition

markiert das Ende eines Vorgangs. Abbildung 3.12 zeigt den Zusammenhang der Entitäten, die ein Ereignis beschreiben. Es gilt der in Abbildung 3.4 gezeigte Zusammenhang zwischen den projektunabhängigen und den projektspezifischen Entitäten.

Das terminierende Ereignis eines Vorgangs wird als *Ergebnis* bezeichnet. Das Ergebnis eines Vorgangs kann der Auslöser für nachfolgende Vorgänge sein. Für die nachfolgenden Vorgänge stellt dieses Ereignis eine *Voraussetzung* dar. Voraussetzungen und Ergebnisse von Vorgängen werden über Ereignisse beschrieben. Über das Ergebnis hat ein Vorgang immer einen eindeutigen Bezug zu einem Leistungsgegenstand. Der Zusammenhang von Vorgang und Ereignis ist in Abbildung 3.13 gezeigt.

3.3.8 eEPK Funktion und Ereignis

Abbildung 3.13 zeigt die Modellierung der Entität *Vorgang* mit Voraussetzungen und Ergebnis als eERM in der Datensicht. Dieser Modelltyp eignet sich zur Beschreibung von Datenstrukturen und stellt die statischen Zusammenhänge zwischen den Informationen dar. Dynamische Aspekte

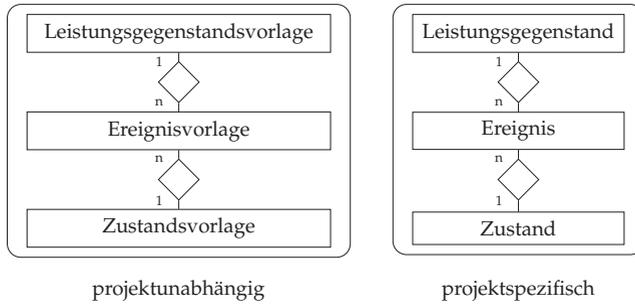


Abbildung 3.12: eERM Ereignis und -vorlage

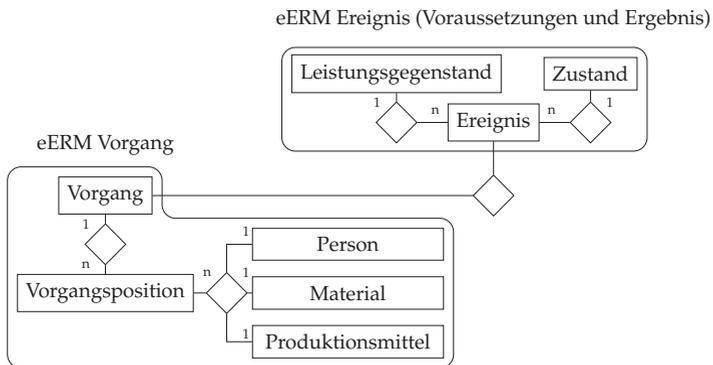


Abbildung 3.13: eERM Vorgang und Ereignis

werden in der Steuerungssicht beschrieben, der eine eEPK zugrunde liegt.

Für den Übergang von der Datensicht auf die Steuerungssicht ist es erforderlich, Beschreibungselemente der eERM auf die Beschreibungselemente der eEPK abzubilden. Die wesentlichen Beschreibungselemente einer eEPK sind die *Funktion* und das *Ereignis*. Die Funktion wird genutzt, um zeitfordernde Geschehen zu modellieren und ein Ereignis, um charakteristische Momente hervorzuheben. Ereignisse können Auslöser aber auch Ergebnis einer Funktion sein.

Ein eERM(*Vorgang*) wird auf eine eEPK(*Funktion*) abgebildet und ein eERM(*Ereignis*) auf ein eEPK(*Ereignis*). Darüber hinaus ist es zulässig, Entitäten aus der Datensicht zu referenzieren. Es ist zu beachten, dass die *Funktion* und das *Ereignis* in der Steuerungssicht eigenständige Beschreibungselemente sind. Dies wird durch die Symbolik in Abbildung 3.14 zum Ausdruck gebracht. Eine Funktion wird als abgerundetes Rechteck dargestellt und ein Ereignis als Sechseck.

Abbildung 3.14 zeigt das Resultat der Abbildung von einem eERM(*Vorgang*) auf eine eEPK(*Funktion*). Bei der Abbildung wird ein Vorgang mit seinen modellierten Voraussetzungen und seinem Ergebnis aus der Datensicht auf die Beschreibungselemente einer eEPK übertragen. Entitäten, die sich nicht verändern, sondern aus der Datensicht referen-

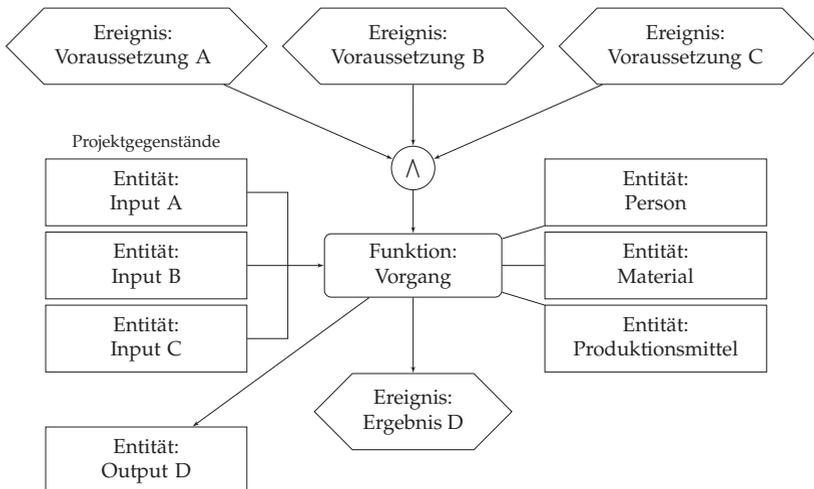


Abbildung 3.14: eEPK Funktion

ziert werden, sind an der rechteckigen Umrandung erkennbar. Durch Auswerten des eERM(*Vorgang*) lässt sich die eEPK(*Funktion*) konkretisieren. Die Abbildung erfolgt nach folgenden Grundsätzen:

Voraussetzungen: Voraussetzungen sind Instanzen der Entität eERM(*Ereignis*). Sie stellen eine Verknüpfung eines Leistungsgegenstands mit einem Zustand dar. Der eEPK(*Funktion*) *Vorgang* wird für jede Voraussetzung der Leistungsgegenstand als Input eingepägt und der zugehörige Zustand als auslösendes eEPK(*Ereignis*). Abbildung 3.14 zeigt dies für die korrespondierenden Paare *Input A - Ereignis A*, *Input B - Ereignis B* und *Input C - Ereignis C*. Ist mehr als ein Ereignis als Auslöser vorhanden, verlangt die Zusammenführung der Ereignisse in der Regel *UND*, dass alle Ereignisse eingetreten sein müssen, bevor eine Funktion beginnt.

Ergebnis: Dem Ergebnis einer eEPK(*Funktion*) *Vorgang* liegt ebenfalls ein eERM(*Ereignis*) zugrunde. Der zugehörige Leistungsgegenstand wird als Output eingepägt und der Zustand als terminierendes eEPK(*Ereignis*). Über die Output-Beziehung ist eine Funktion immer eindeutig mit einem Leistungsgegenstand verknüpft. Im Rahmen dieser Arbeit folgt auf eine Funktion zwingend ein Ereignis.

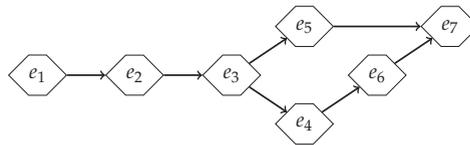
Funktion: Über die Verknüpfung der Entität eERM(*Vorgang*) zu einer Menge an *Vorgangspositionen* und weiter zu den *Personen*, den *Materialien* und den *Produktionsmitteln*, besteht die Möglichkeit, die Dauer der eEPK(*Funktion*) *Vorgang* sowie weitere Angaben zu ermitteln.

3.3.9 Ereignisgraph

Beschreibt man auf der Grundlage des in Abbildung 3.13 dargestellten eERM eine Menge an *Vorgängen*, dann lassen sich diese nach dem in Abschnitt 3.3.8 beschriebenen Ansatz auf Funktionen abbilden. Ist für jeden *Vorgang* die Menge der Voraussetzungen spezifiziert, dann ergibt sich aus der Abbildung eine Prozesskette. Abhängig von der Menge der betrachteten *Vorgänge* beschreibt diese Prozesskette den gesamten Prozess oder einen Teilprozess. Betrachtet man diejenige Teilmenge an *Vorgängen*, bei der jeder *Vorgang* über die Verknüpfung seines Ergebnisses mit demselben Leistungsgegenstand verknüpft ist, dann resultiert ein Teilprozess. Dieser Teilprozess beschreibt einen konkreten Leistungsprozess für einen

Leistungsgegenstand. Bildet man die eEPK(*Ereignisse*) dieses Teilprozesses auf Knoten ab und ihre indirekten Beziehungen auf Kanten, so stellt das Gebilde aus Knoten und Kanten einen gerichteten *Ereignisgraphen* dar.

So, wie sich ein Ereignisgraph aus einer gegebenen Prozesskette ermitteln lässt, kann ein Ereignisgraph auch umgekehrt genutzt werden, um aus den Kanten zwischen den Ereignissen auf die Voraussetzungen für Funktionen und darüber auf die Voraussetzungen für Vorgänge zu schließen. Dieses Vorgehen verlangt, zuerst die Ereignisgraphen aufzustellen und aus der Menge der Kanten die Voraussetzungen für Funktionen abzuleiten. Besteht in einem Ereignisgraphen zwischen zwei Ereignissen e_1 und e_2 eine Kante (e_1, e_2) , dann ist der Funktion f_2 , die mit dem Ereignis e_2 terminiert, e_1 als Voraussetzung einzutragen. Abbildung 3.15 zeigt einen Ereignisgraphen, der den Herstellungsprozess des Bauteils *StB-Decke 1* beschreibt.



Ereignis Leistungsgegenstand : Zustand

- e_1 StB-Decke 1 : eingeschalt
- e_2 StB-Decke 1 : bewehrt
- e_3 StB-Decke 1 : betoniert
- e_4 StB-Decke 1 : Festigkeit zum Abbauen der Unterst tzung erreicht
- e_5 StB-Decke 1 : Festigkeit zum Ausschalen erreicht
- e_6 StB-Decke 1 : ausgeschalt
- e_7 StB-Decke 1 : Unterst tzung abgebaut

Abbildung 3.15: Ereignisgraph

Ein Ereignisgraph besteht aus einer Menge an Ereignissen und einer Relation in der Menge der Ereignisse. Die Relation beschreibt die Vorg nger- und Nachfolgerbeziehungen, aus denen sich Abl ufe ermitteln lassen. Jeder Leistungsprozess nach Abbildung 3.8 l sst sich auf einen Ereignisgraphen  berf hren. Hierf r ist das Attribut *Ablauf* auszu-

werten, das die Voraussetzungen für Vorgänge innerhalb eines Leistungsprozesses enthält.

Das Konzept der Ereignisgraphen wird in dem hier beschriebenen Modellierungsansatz genutzt, um den Spezifikationsaufwand bei der Entwicklung von Prozessmodellen zu reduzieren und um sicherzustellen, dass vergleichbare Leistungsgegenstände durchgängig durch gleiche Leistungsprozesse beschrieben werden. Ein Leistungsprozess wird einmalig, projektunabhängig modelliert und als Leistungsprozessvorlage bereitgestellt. Innerhalb eines Projekts vervielfältigt sich der Ereignisgraph eines Leistungsprozesses entsprechend der Anzahl konkreter Leistungsgegenstände vom Typ einer Leistungsgegenstandsvorlage. Da es für Bauprojekte typisch ist, dass es eine Vielzahl von Leistungsgegenständen mit großer Häufigkeit gibt, führt dieses Vorgehen zu einer erheblichen Reduzierung des Spezifikationsaufwands.

3.4 Bauablaufplanung: Modellierung

Die klassische Ablaufplanung erfolgt in drei Schritten: (1) Die Menge der Vorgänge wird spezifiziert, (2) den Vorgängen werden Dauern zugeordnet und (3) Abhängigkeiten zwischen den Vorgängen werden angegeben.

Zu Punkt (1) und (2): Für die Formulierung eines Vorgangs führt sich der planende Ingenieur gedanklich die herzustellenden Bauteile eines Bauwerks vor Augen. Ein Beispiel für einen Vorgang könnte sein: Außenputzarbeiten. Gedanklich seien alle Außenwandflächen betrachtet, die nach dem Abschluss des Vorgangs verputzt sein sollen. Nicht enthalten sei der Sockelputz, der in einem getrennten Vorgang modelliert wird. Dem Vorgang wird eine Dauer zugewiesen. Wenn nicht durch Kommentare der Vorgang Außenputzarbeiten beschrieben wird, bleiben die Annahmen des Ingenieurs zu dem Arbeitsergebnis implizit. Nur durch Überprüfen aller Vorgänge kann festgestellt werden, ob der Vorgang auch den Sockelputz umfasst oder nicht. Der hier verfolgte Ansatz macht das gedanklich vorgehaltene Bauwerksmodell explizit. Bauteilzustände stehen im Vordergrund. Sie werden als Ereignisse modelliert. Für ein Ereignis wird ein treffender Vorgang formuliert. Vorgangspositionen werden in dem Maße einbezogen, wie sie für das Erreichen des Ereignisses erforderlich sind. Dieses Vorgehen ist nicht auf Bauteile begrenzt, sondern gilt für Leistungsgegenstände im Allgemeinen. Da sich Ereignisse auf konkrete Leistungsgegenstände beziehen, geht beispielsweise aus einem Ausführungsvorgang explizit hervor, welche konkreten Bauteile welchen Zustand erreichen.

Zu Punkt (3): Betrachtet wird die Planung der Vergabe von Nachunter-

nehmerleistungen. Folgende Vorgänge werden nach dem klassischen Ansatz modelliert: (1) LV erstellen, (2) Versand der Unterlagen, (3) Rücklauf der Angebote, (4) Auswertung/Verhandlung/Vergabe und (5) Beauftragung. Ein typisches Vorgehen ist, diese Vorgänge durch fixe Termine sequenziell anzuordnen. Ändert man im Rahmen der Bearbeitung des Prozessmodells jedoch einen Vorgang, dessen Ergebnis eine Planungsunterlage betrifft, die einer der genannten Bearbeitungsschritte (1)-(5) erfordert, entstehen Fehler. Diese sind nur schwer zu ermitteln, da die Benennung der Vorgänge nicht immer Aufschluss über ihren exakten Umfang gibt.

In diesem Abschnitt wird ein Modellierungskonzept beschrieben, das das Gedankenmodell des planenden Ingenieurs explizit macht. Vorgänge werden nicht direkt verknüpft, sondern stehen indirekt über Ereignisse in Beziehung. Darüber hinaus wird der Gesamtprozess zur Beschreibung eines Bauprojekts aus vordefinierten Teilprozessen systematisch aufgebaut. Dieses Vorgehen reduziert den Spezifikationsaufwand und stellt sicher, dass vergleichbare Projektausschnitte konsistent beschrieben werden.

Die Modellierung erfolgt auf der Grundlage der in Abschnitt 3.3 beschriebenen Datenstrukturen. Am Beispiel des in Abbildung 3.3 dargestellten Projekts *Garage* wird das Vorgehen erläutert. Zunächst werden die erforderlichen projektunabhängigen Informationen modelliert. Diese werden auf die Leistungsgegenstände des Projekts angewendet. Es entsteht ein Gebilde aus unabhängigen Teilprozessen, die untereinander in Beziehung zu setzen sind. Dies erfolgt auf der Grundlage einer vorgangsorientierten Herangehensweise. Sie erlaubt es, für die Spezifikation lediglich einen lokalen Ausschnitt aus dem Gesamtprozess zu betrachten. Das Ergebnis der Modellierung ist ein Prozessmodell, das alle technologischen Randbedingungen vollständig und korrekt berücksichtigt. Auf die Modellierung von Strukturierungsbäumen wird nicht eingegangen. Sie werden als gegeben betrachtet und in den Ausschnitten dargestellt, die für die Strukturierung der spezifizierten Mengen wesentlich sind.

3.4.1 Modellieren projektunabhängiger Informationen

Zur Identifikation werden Instanzen, die eine Vorlage darstellen, mit einem hochgestellten v markiert.

Leistungsgegenstandsvorlagen: Die Spezifikation von Vorgangsvorlagen, z.B. *betonieren*, erfolgt für konkrete Leistungsgegenstandsvorlagen, z.B. *StB-Decke*. Ohne den Bezug zu einer konkreten Leistungsgegenstandsvorlage ist es nicht möglich, zwischen den Vorgangsvorlagen *be-*

tonieren einer Decke und betonieren eines Fundaments unterscheiden zu können. Die Unterscheidung ist aber essentiell, da sich ihre Vorgangspositionen und damit ihre Aufwände und Randbedingungen unterscheiden. Bevor eine Instanz einer Vorgangsvorlage modelliert werden kann, muss die ihr zugrunde liegende Leistungsgegenstandsvorlage existieren. Die Modellierung beginnt mit der Spezifikation von Leistungsgegenstandsvorlagen.

Abbildung 3.16 zeigt die modellierten Leistungsgegenstandsvorlagen *StB-Fundament*, *KS-Wand* und *StB-Decke*. Sie sind den Knoten des Strukturierungsbaums DIN 276-1 [2008]¹⁸ zugeordnet.

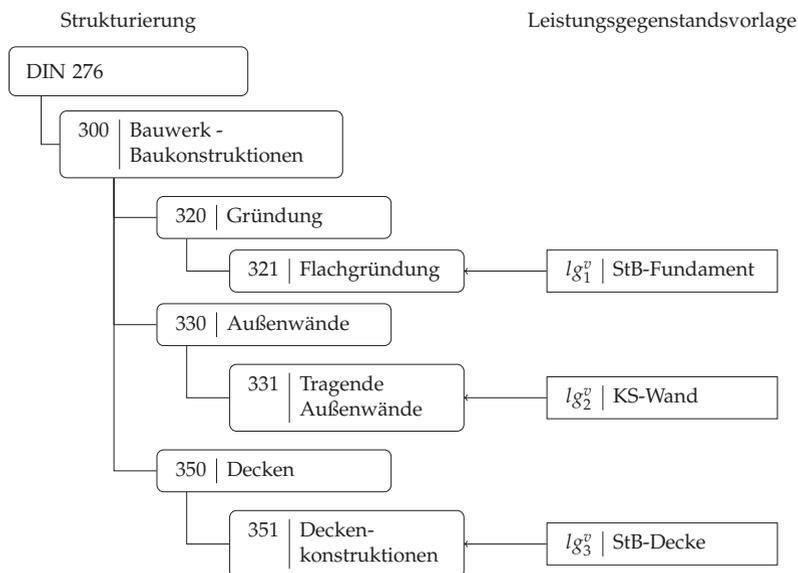


Abbildung 3.16: Projekt Garage - Leistungsgegenstandsvorlagen

Ereignisvorlagen und Vorgangsvorlagen: Abbildung 3.16 zeigt die für dieses Projekt relevanten Bauteilvorlagen (Leistungsgegenstandsvorlagen). Für diese Bauteilvorlagen sind Leistungsprozessvorlagen zu spezifizieren. Bevor eine Prozessvorlage modelliert werden kann, ist die Men-

¹⁸ [DIN 276-1, 2008, Kapitel 4: Kostengliederung, S. 10 ff.]

ge der Vorgangsvorlagen und der Ereignisvorlagen anzulegen. Jeweils eine Vorgangsvorlage ist mit genau einer Ereignisvorlage zu verknüpfen. Die Einteilung eines Prozesses in Vorgänge ist abhängig vom gewünschten Detaillierungsgrad. Da es sich hier um Ausführungsvorgänge handelt, wird ein Detaillierungsgrad angestrebt, der eine Einteilung der Vorgänge nach Bauarbeitsschlüssel¹⁹ erlaubt. Entsprechend wird zur Strukturierung der Vorgänge ein Baum nach Bauarbeitsschlüssel angelegt. Ist ein Bezug der Vorgänge auf Positionen des Leistungsverzeichnisses aus baubetrieblicher Sicht erforderlich, müssen die Vorgangsvorlagen zusätzlich in einem Baum strukturiert werden, der eine Gliederung nach Standardleistungsbuch für das Bauwesen (STLB)²⁰ enthält.

Abbildung 3.17 zeigt die spezifizierten Vorgangsvorlagen, strukturiert nach Bauarbeitsschlüssel, die für das Projekt *Garage* erforderlich sind. Die Spezifikation der Vorgangsvorlagen sei am Beispiel des *StB-Fundaments* beschrieben. Für ein *StB-Fundament* sind bei einem Detaillierungsgrad nach Bauarbeitsschlüssel die Zustände *eingeschalt*, *bewehrt*, *betoniert*, *Festigkeit zum Ausschalen erreicht* und *ausgeschalt* wesentlich. Für jeden genannten Zustand wird eine Zustandsvorlage angelegt. Durch die Kombination der Leistungsgegenstandsvorlage *StB-Fundament* mit jeweils einer Zustandsvorlage erhält man eine Ereignisvorlage. Die Legende von Abbildung 3.17 zeigt diesen Zusammenhang. Aus Gründen der Übersichtlichkeit sind die Ereignisvorlagen nicht in der Abbildung dargestellt.

Ereignisvorlagen sind automatisch nach DIN 276-1 [2008]²¹ strukturiert, da über die Leistungsgegenstandsvorlage *StB-Fundament* eine Abbildung auf die Knoten des entsprechenden Strukturierungsbaums vorliegt. Ist eine Ereignisvorlage spezifiziert, kann eine Vorgangsvorlage angelegt werden. Für die Ereignisvorlage *StB-Fundament: eingeschalt* wird die Vorgangsvorlage *einschalen* angelegt. Da bekannt ist, dass sich die Tätigkeit *einschalen* auf ein *StB-Fundament* bezieht, lassen sich Vorgangspositionen zuordnen. Aufwände, Ressourcen, Material und Produktionsmittel der Vorgangspositionen beziehen sich auf das *StB-Fundament*. Analog werden die Ereignisvorlagen für die Zustände *bewehrt*, *betoniert* und *ausgeschalt* mit Vorgangsvorlagen verknüpft. Für die Ereignisvorlage *StB-Fundament: Festigkeit zum Ausschalen erreicht* wird eine Vorgangsvorlage *abbinden* angelegt. Es handelt sich um eine Vorgangsvorlage, die keine Ressourcen erfordert. Der resultierende Zustand ist jedoch für nachfolgen-

¹⁹ [Hoffmann, 2002; BAUORG, 1998, Bauarbeitsschlüssel]

²⁰ [Schiller, 2003, Leistungsbeschreibung mit STLB-bau Dynamische Baudaten, S. 19 ff.]

²¹ [DIN 276-1, 2008, Kostengliederung, S. 10 ff.]

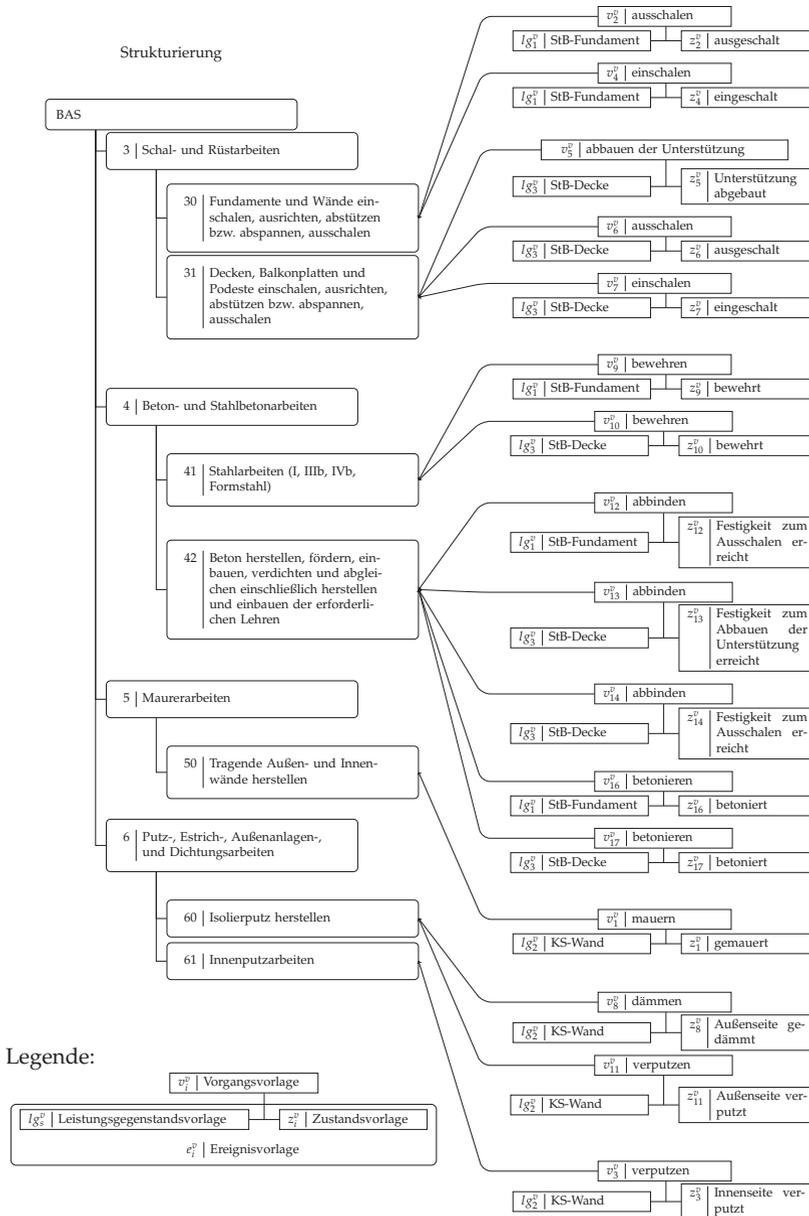


Abbildung 3.17: Projekt Garage - Ereignisvorlagen und Vorgangsvorlagen

de Vorgänge eine Voraussetzung und muss daher modelliert werden. Gibt es verschiedene Abbindefristen, die für Folgevorgänge die Auslöser sind, müssen mehrere Abbindevorgänge getrennt angelegt werden, die über ihre Vorgangspositionen unterschiedliche Dauern berücksichtigen. Dies ist beispielsweise für eine *StB-Decke in Ortbeton* der Fall, bei der für das *aus-schalen* und für das *entfernen der Unterstützung* unterschiedliche Abbindefristen vorgeschrieben sind. Auf die Spezifikation der Vorgangspositionen wird hier nicht weiter eingegangen.

Leistungsprozessvorlagen: Abbildung 3.17 enthält dieselben Vorgangs-Ereignis-Paare wie Abbildung 3.18, nur sind diese anders geordnet. Die Menge der Ereignisvorlagen lässt sich anhand ihrer Leistungsgegenstandsvorlage in disjunkte Teilmengen zerlegen. Für jede Teilmenge wird eine Leistungsprozessvorlage angelegt. Innerhalb jeder Teilmenge werden die Voraussetzungen für die Vorgänge spezifiziert und in dem Attribut *Ablauf* registriert. Auf der Grundlage der Voraussetzungen und der spezifizierten Tupel aus Vorgangsvorlage und Ereignisvorlage lässt sich eine eEPK generieren.

3.4.2 Modellieren projektspezifischer Informationen

Die Informationen, die sich auf ein konkretes Projekt beziehen, werden als projektspezifisch bezeichnet. Ihre Modellierung erfolgt auf der Grundlage von Vorlagen. Jede projektspezifische Ausprägung wird aus einer allgemeinen Vorlage abgeleitet, die projektunabhängig formuliert ist.

Strukturierung: Die Gestaltung und Modellierung eines Projekts beginnt mit der Entwicklung eines Projektstrukturplans (PSP). Nach der Definition von PMBOK [2003]²² ist ein Projektstrukturplan:

„eine an Liefergegenständen orientierte hierarchische Strukturierung der durch das Projektteam auszuführenden Arbeit, um die Projektziele zu erfüllen und die erforderlichen Liefergegenstände zu erstellen.“

Verschiedene Arten von PSP werden unterschieden. Der ablauforientierte PSP dient der Gliederung des Projekts nach ablaufbedingten Gesichtspunkten. Der aufbauorientierte PSP dient der Gliederung des Pro-

²² [PMBOK, 2003, Work Breakdown Structure (WBS) / Projektstrukturplan (PSP), S. 373]

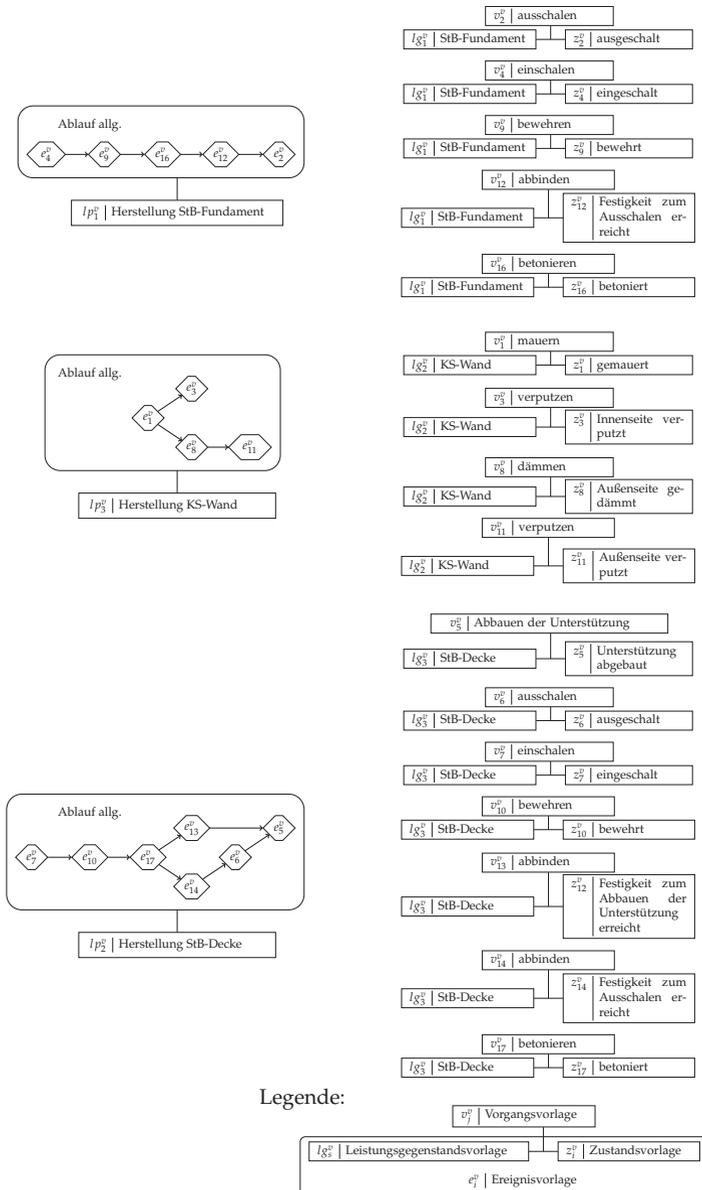


Abbildung 3.18: Projekt Garage - Leistungsprozessvorlagen

jekts nach seiner Objektstruktur. Mischformen, die beide Ansätze in einer Struktur kombinieren, sind üblich. Ziel der Strukturierung ist es, eine überschaubare Aufgliederung des Projektes in Teilprojekte bis hin zu Arbeitspaketen zu erhalten, auf deren Grundlage Kosten, Termine und Verantwortlichkeiten zugeordnet werden können. Arbeitspakete stellen die feinste Gliederungseinheit dar.

Neben der Struktur umfasst ein PSP auch weitere Informationen. An dieser Stelle wird nur auf die hierarchische Gliederung eingegangen. Sie unterstützt den planenden Ingenieur bei der Zerlegung des Projekts in überschaubare Einheiten. Die oberste Ebene wird durch den Projektnamen beschrieben. Abhängig vom gewählten Strukturierungsansatz folgen Gliederungsebenen, die das Projekt hierarchisch in kleinere Einheiten einteilt. Als kleinste Gliederungseinheit werden die Instanzen der Leistungsgegenstandsvorlagen in den Baum eingehängt. Hierdurch wird gewährleistet, dass ein einheitlicher Detaillierungsgrad über das gesamte Projekt eingehalten wird. Innerhalb eines Projektstrukturbaums darf eine Leistungsgegenstandsvorlage mehrfach zugeordnet werden. Für den Baum stellen die Leistungsgegenstandsvorlagen einen Verzweigungsknoten dar, nicht das Element einer Menge, das es zu strukturieren gilt. Abbildung 3.19 zeigt das Prinzip der Gliederung.

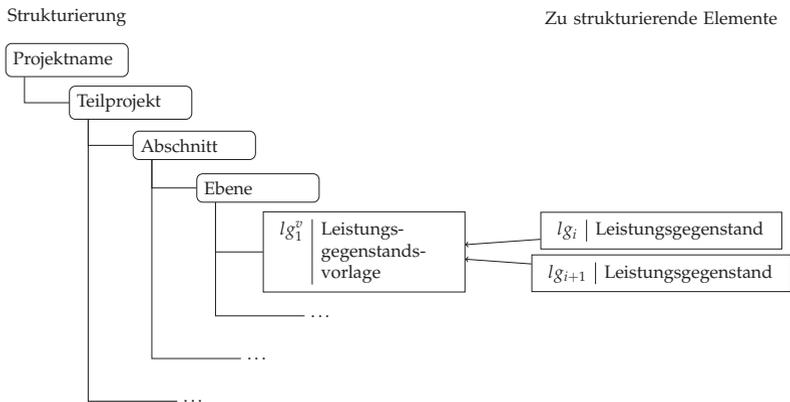


Abbildung 3.19: Prinzip Projektstrukturbaum

Leistungsgegenstände: Den Leistungsgegenstandsvorlagen, als feinste Gliederungseinheit, werden konkrete Leistungsgegenstände zugeordnet. Für das Projekt *Garage* sind dies die Bauteile *StB-Fundament 1*, *StB-Fundament 2*, *KS-Wand 1*, *KS-Wand 2* und *StB-Decke 1*. Abbildung 3.19 zeigt die Strukturierung der Bauteile für das Projekt *Garage*.

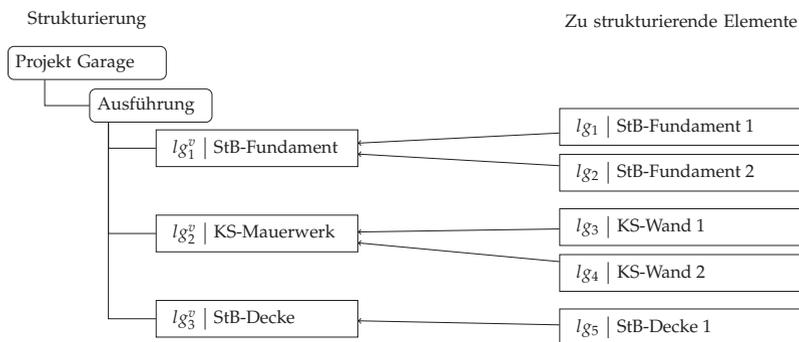


Abbildung 3.20: Projekt Garage - Projektstrukturbaum

Leistungsprozesse: Mit der Zuordnung der Leistungsgegenstände zu den Leistungsgegenstandsvorlagen wird der Umfang des Prozessmodells festgelegt. Auf der Grundlage der Leistungsprozessvorlagen, die jeweils mit einer Leistungsgegenstandsvorlage verknüpft ist, lassen sich für die spezifizierten Leistungsgegenstände Teilprozesse ableiten und in das Prozessmodell übertragen. Abbildung 3.21 zeigt das Ergebnis, nachdem für jedes Bauteil ein entsprechender Teilprozess angelegt wurde. Dargestellt sind nur die Ereignisgraphen. Jedes Ereignis wurde von einer Ereignisvorlage abgeleitet. Um den Weg zurückverfolgen zu können, sind in der Darstellung unterhalb der Ereignisse die Identifikatoren ihrer Vorlage erwähnt. So, wie die Ereignisse, werden auch Zustände und Vorgänge mit ihren Vorgangspositionen generiert.

Zu diesem Zeitpunkt besteht das Prozessmodell aus unabhängigen Teilprozessen, die nicht miteinander verknüpft sind. Jeder Teilprozess beschreibt den Leistungsprozess eines Leistungsgegenstands. Zusätzliche Verknüpfungen müssen spezifiziert werden.

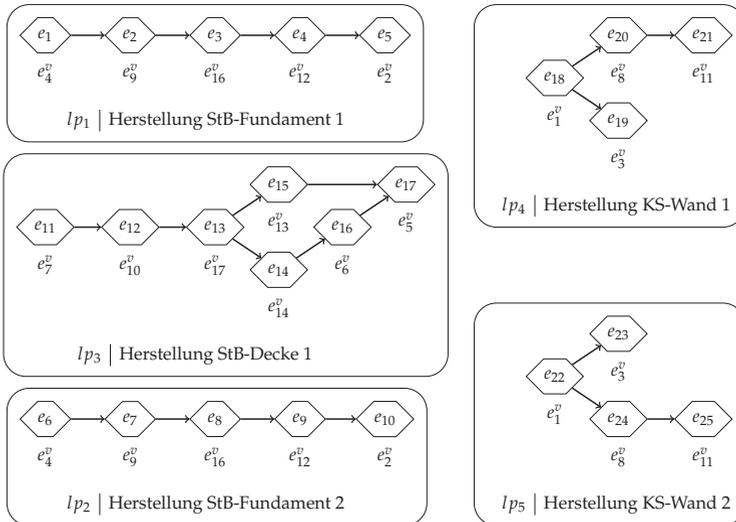


Abbildung 3.21: Projekt Garage - Leistungsprozesse

3.4.3 Spezifikation von Voraussetzungen

Die Spezifikation der zusätzlichen Verknüpfungen erfolgt durch die Angabe von Voraussetzungen für Vorgänge. Es müssen nur diejenigen Voraussetzungen angegeben werden, die sich nicht auf denselben Leistungsgegenstand beziehen. Ein Vorgang v_i hat über sein Ergebnis e_i immer einen Bezug zu einem Leistungsgegenstand lg_s . Ist das Erreichen des Zustands z_k eines Leistungsgegenstands $lg_t \neq lg_s$ aus technologischer Sicht für die Bearbeitung von v_i erforderlich, dann muss für den Vorgang v_i das Ereignis $e_k(lg_t, z_k)$ als Voraussetzung spezifiziert werden.

In dem Projekt *Garage* sind folgende Voraussetzungen anzugeben: Mit dem Mauern kann erst begonnen werden, wenn die Fundamente die Festigkeit zum Ausschalen erreicht haben. Das Einschalen der Decke verlangt, dass die Wände gemauert sind. Mit dem Verputzen der Innenwände kann erst begonnen werden, wenn die Decke ausgeschalt ist. Die Spezifikation von Voraussetzungen erfolgt für Vorgänge. Abbildung 3.22 stellt dies für die Vorgänge *mauern der KS-Wand 1+2*, *einschalen der StB-Decke 1* und *verputzen der KS-Wand 1+2* dar.

Durch die zusätzlichen technologischen Abhängigkeiten werden die

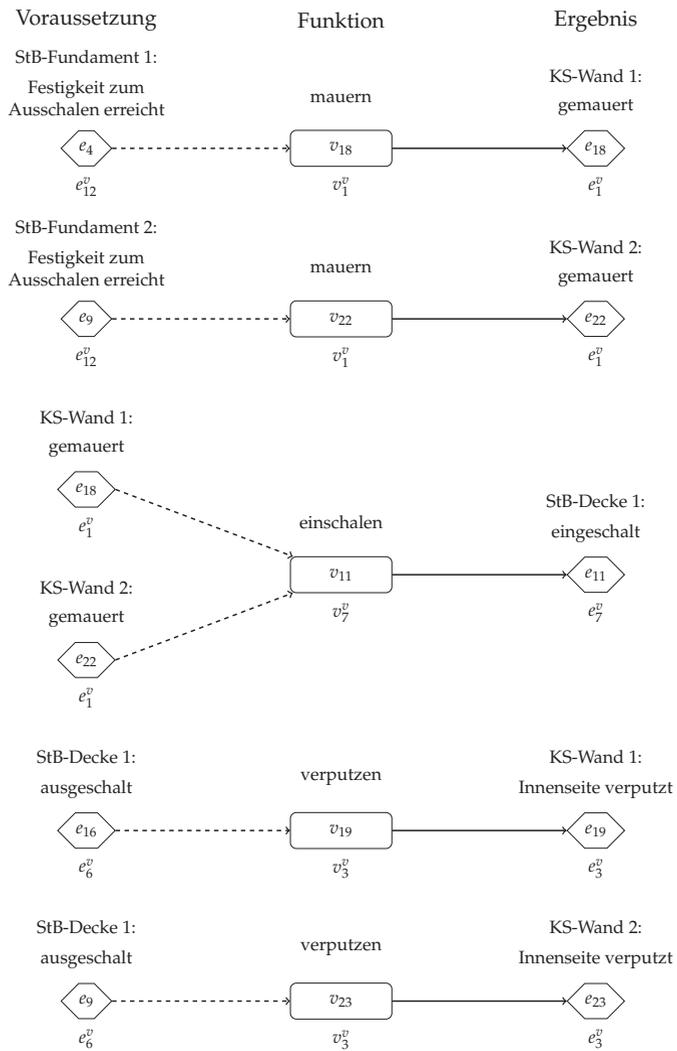


Abbildung 3.22: Projekt Garage - Voraussetzungen

unabhängigen Teilprozesse miteinander verknüpft. Auf der Grundlage der Leistungsgegenstände, den ihnen zugeordneten Leistungsprozessen und den zusätzlich spezifizierten Voraussetzungen ergibt sich der in Abbildung 3.23 gezeigte Prozess, dargestellt als eEPK. Ausgehend von diesem Prozessmodell lassen sich die erforderlichen Graphen für die Netzplantechnik durch geeignete Abbildungen bestimmen.

3.4.4 Spezifikationsaufwand

Abhängig von der Einteilung des Projekts in Leistungsgegenstände ergibt sich für die Modellierung des Prozessmodells ein konkreter Spezifikationsaufwand. Unterschiedliche Einteilungen können zu unterschiedlichen Prozessmodellen mit entsprechend unterschiedlichen Spezifikationsaufwänden führen. Vergleicht man verschiedene Prozessmodelle, die sich im Ergebnis auf denselben Ablauf abbilden lassen, dann ist der Unterschied im Spezifikationsaufwand auf das Anlegen der Leistungsgegenstände, das Entwickeln der für die Leistungsgegenstände erforderlichen Leistungsprozessvorlagen und die Angabe der projektspezifischen Voraussetzungen für Vorgänge zurückzuführen. Um den Spezifikationsaufwand unterschiedlicher Prozessmodelle miteinander vergleichen zu können, wird eine Abbildung²³ $a_\omega : \Omega \rightarrow N$ eingeführt, die ein Prozessmodell ω auf die Menge der natürlichen Zahlen abbildet. Ist der Ausdruck $a_\omega(\omega_1) < a_\omega(\omega_2)$ wahr, dann bedeutet dies, dass der Spezifikationsaufwand für das Prozessmodell ω_1 kleiner ist als für das Prozessmodell ω_2 .

$$a_\omega : \Omega \rightarrow N \quad (3.1)$$

a_ω Spezifikationsaufwand
 Ω Menge der Prozessmodelle
 N Menge der natürlichen Zahlen

Die Berechnung von Aufwänden erfolgt durch Aufsummieren der Mächtigkeiten von Mengen. Die Mächtigkeit einer endlichen Menge A beschreibt die Anzahl der in einer Menge A enthaltenen Elemente und wird mit $|A|$ ausgedrückt. Betrachtet man eine Relation R als Menge an Verknüpfungen, dann ergibt $|R|$ die Anzahl der Verknüpfungen in der Relation R .

²³ Siehe Abschnitt A↑ Abbildung

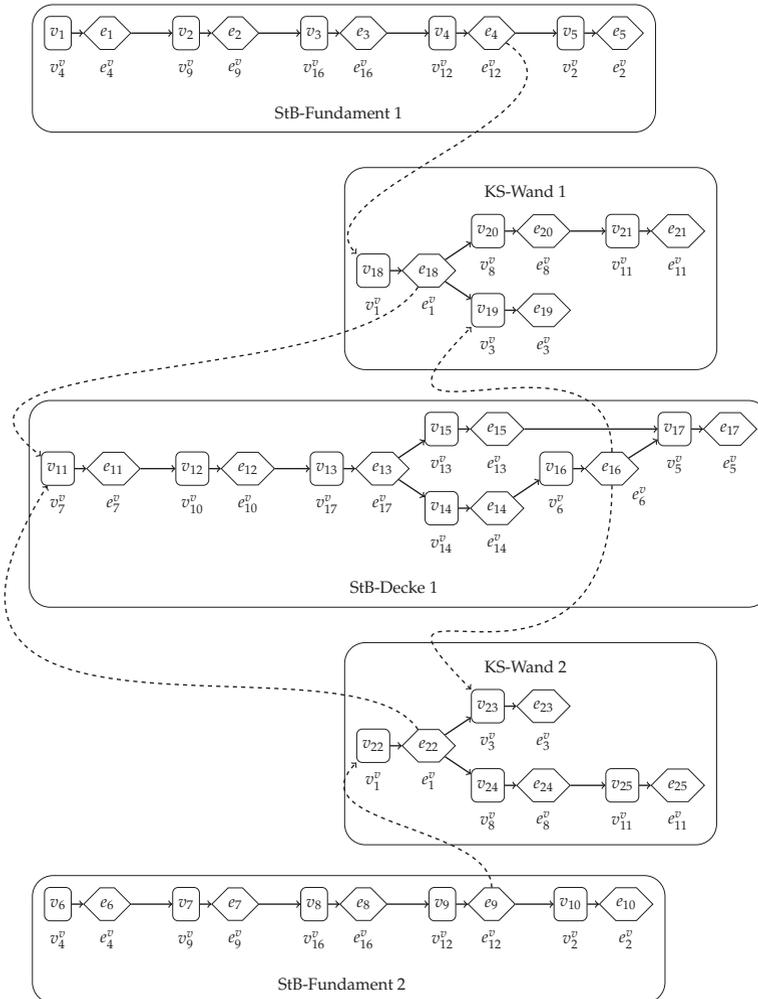


Abbildung 3.23: Projekt Garage - Gesamtprozess als ePK

Der Spezifikationsaufwand wird definiert als die Summe aus drei Anteilen: (1) der Anzahl der spezifizierten Leistungsgegenstände, (2) der Summe der Aufwände der verwendeten Leistungsprozessvorlagen und (3) der Anzahl der zusätzlichen projektspezifischen Voraussetzungen. Der Aufwand zur Spezifikation einer Leistungsprozessvorlage berechnet sich als Summe aus der Anzahl der Knoten und der Anzahl der Kanten seines Ereignisgraphen:

$$a_{\omega}(\omega) = |LG| + \sum_{i=1}^n a(lp_i^v) + |R_{E-V}| \quad (3.2)$$

$a_{\omega}(\omega)$	Spezifikationsaufwand eines Prozessmodells ω
LG	Menge der Leistungsgegenstände
$a(lp_i^v)$	Aufwand für eine Leistungsprozessvorlage lp_i^v
R_{E-V}	Relation der projektspezifischen Voraussetzungen für Vorgänge
E	Menge der Ereignisse
V	Menge der Vorgänge

$$a(lp_i^v) = |EV_i| + |Ablauf_i| \quad (3.3)$$

EV_i	Menge der Ereignisvorlagen einer Leistungsprozessvorlage lp_i^v
$Ablauf_i$	Menge der Voraussetzungen innerhalb einer Leistungsprozessvorlage lp_i^v

Der so definierte Spezifikationsaufwand ist dimensionslos. Er berücksichtigt nicht alle zu spezifizierenden Instanzen. Seine Aussagekraft bezieht sich auf den Vergleich von unterschiedlichen Prozessmodellen, die sich auf identische Abläufe abbilden lassen. Der Vergleich des Spezifikationsaufwands von Prozessmodellen, die zu unterschiedlichen Abläufen führen, ist aussagenlos. Für das Projekt *Garage* ergibt sich wie folgt ein Spe-

zifikationsaufwand von 41:

$$\begin{aligned}
 |LG| &= 5 && \text{Abbildung 3.20} \\
 a(lp_1^v) &= 5 + 4 = 9 && \text{Abbildung 3.18 - StB-Fundament} \\
 a(lp_2^v) &= 4 + 3 = 7 && \text{Abbildung 3.18 - KS-Wand} \\
 a(lp_3^v) &= 7 + 7 = 14 && \text{Abbildung 3.18 - StB-Decke} \\
 \sum_{i=1}^n a(lp_i^v) &= 9 + 7 + 14 = 30 \\
 |R_{E-V}| &= 6 && \text{Abbildung 3.22} \\
 a_\omega(\textit{Garage}) &= 5 + 30 + 6 = 41
 \end{aligned}$$

3.5 Diskussion

In diesem Kapitel wurde eine Methodik beschrieben, die das Gestalten und Modellieren des Prozesses zur formalen Beschreibung von Bauabläufen in den Mittelpunkt rückt. Das Ergebnis ist ein Prozessmodell, das die technologisch zwingenden Abhängigkeiten vollständig und konsistent beschreibt. Es kann auf einen Netzplan abgebildet und nach den bekannten Verfahren der Netzplantechnik ausgewertet werden. Aufgrund der formalen Herangehensweise sind die Eigenschaften des Prozessmodells mathematisch begründet und die Vollständigkeit und Korrektheit kann konsistent zur Eingabe gewährleistet werden.

Grundlage für das Modellierungskonzept ist eine Einteilung des Modellierungsgegenstands in projektunabhängige und in projektspezifische Informationen. Teilabläufe werden projektunabhängig beschrieben und als Vorlagen bereitgestellt. Innerhalb eines Projekts werden die Vorlagen genutzt, um für konkrete Leistungen Teilprozesse abzuleiten. Über die Angabe von Voraussetzungen werden Teilprozesse miteinander verknüpft. Ziel der Methodik ist es, bei der Modellierung mit einem Minimum an Eingangsinformationen auszukommen und formale Mechanismen zur Berechnung des Zusammenhangs anzuwenden.

Der Spezifikationsaufwand für die Entwicklung von Prozessmodellen nach dem vorliegenden Ansatz hängt stark von den Teilprozessen ab, die als Leistungsprozessvorlagen in die Modellierung einfließen. Eine Leistungsprozessvorlage beschreibt einen Teilprozess, bestehend aus Vorgängen und Ereignissen, zur Erbringung einer Leistung. Die zu erbringende Leistung wird beschrieben durch eine Leistungsgegenstandsvorlage. Was zu dem Umfang einer Leistungsgegenstandsvorlage gehört, wird aus der Anschauung heraus beurteilt und kann beliebig festgelegt werden. Ob

beispielsweise eine Trockenbauwand alleine betrachtet wird oder zusammen mit Elektroleitungen und -installationen, bzw. mit Wandbekleidung wie Fliesen oder Tapete, entscheidet der Betrachter. Entsprechend ergeben sich unterschiedliche Möglichkeiten zur Gestaltung von Leistungsprozessvorlagen, mit denen sich im Ergebnis derselbe Prozess beschreiben lässt. Abhängig von den eingesetzten Leistungsprozessvorlagen und der Einteilung des Projekts in Leistungsgegenstände, kann der Aufwand für die Spezifikation desselben Prozesses variieren. Bei der Komplexität von Bauabläufen lässt sich im Vorfeld nicht erfassen, welche Einteilung sich als die günstigste erweist. Ein Vorschlag für eine optimale Einteilung lässt sich nicht realisieren.

Rückblickend lässt sich aber beurteilen, wie gut eine Einteilung war. Dieses Wissen kann dazu genutzt werden, um eine Basis an Leistungsprozessvorlagen zu verbessern. Führt man für einen konkreten, spezifizierten Prozess, der auf der Grundlage der hier beschriebenen Modellierungstechnik entwickelt wurde, eine Mustererkennung durch, dann lässt sich dieser auf unterschiedliche Art und Weise in disjunkte Teilprozesse zerlegen. Jeder Zerlegung lässt sich ein Spezifikationsaufwand zuordnen. Bestimmt man in der Menge der möglichen Zerlegungen diejenige, für die der Spezifikationsaufwand minimal ist, dann erhält man eine Menge an optimalen Zerlegungen. Jede Zerlegung stellt eine Variante dar, denselben Prozess durch unterschiedliche Leistungsgegenstandseinteilungen aufzubauen. Jeder Teilprozess stellt einen Leistungsprozess dar. Gleiche Leistungsprozesse gehören zu derselben Leistungsprozessvorlage. Der Umfang der jeweiligen Leistungsprozessvorlage wird auf eine Leistungsgegenstandsvorlage abgebildet. Nur das Rückrechnen von optimalen Leistungsprozessvorlagen erlaubt eine Aussage darüber, wie effizient eine Menge an eingesetzten Leistungsprozessvorlagen für einen gegebenen Prozess war.

Da der Erfolg und der Nutzen für das Modellieren komplexer Bauabläufe auf der Grundlage des hier beschriebenen Ansatzes entscheidend vom Spezifikationsaufwand abhängt, wird der kontinuierlichen Anpassung und Verbesserung der Basis an Leistungsprozessvorlagen große Aufmerksamkeit beigemessen. Kern dieser Arbeit ist es, diesem Anspruch zu genügen. In Kapitel 4 wird eine Mustererkennungsmethodik beschrieben, die es erlaubt, auf der Grundlage eines Branch-and-Bound-Verfahrens die Lösungsmenge für das Optimierungsproblem zur Bestimmung optimaler Leistungsprozessvorlagen für gegebene Prozessmodelle zu bestimmen.

Kapitel 4

Muster in Bauabläufen

In Kapitel 3 wurde eine Herangehensweise zum Gestalten und Modellieren von Bauprozessen beschrieben, die eine Einteilung des Modellierungsgegenstands in projektunabhängige und projektspezifische Informationen vornimmt. Diese Einteilung dient der Reduzierung des Spezifikationsaufwands. Ziel ist es, Teilabläufe einmalig und projektneutral zu beschreiben und sie dann als Leistungsprozessvorlagen in der Planung konkreter Projekte zu nutzen. Eine Reduzierung des Spezifikationsaufwands ergibt sich aus zwei Gründen:

1. Ein Projekt wird in Leistungsgegenstände zerlegt. Für jeden Leistungsgegenstand wird ein Leistungsprozess aus einer Leistungsprozessvorlage abgeleitet. Je größer die Anzahl der Leistungsprozesse ist, die von derselben Vorlage abgeleitet werden, desto größer ist die Einsparung des Spezifikationsaufwands.
2. Die Anzahl der Vorlagen nimmt mit jedem realisierten Projekt zu. Immer weniger Vorlagen müssen für ein konkretes Projekt neu entwickelt werden.

Abhängig von der Einteilung des Projekts in Leistungsgegenstände, ergibt sich für die Modellierung des Prozessmodells ein konkreter Spezifikationsaufwand. Dieser ist auf das Anlegen der projektunabhängigen Leistungsprozessvorlagen und die Spezifikation der projektspezifischen Voraussetzungen für Vorgänge zurückzuführen. Unterschiedliche Einteilungen des Projekts in Leistungsgegenstände können im resultierenden Spezifikationsaufwand variieren.

Die Einteilung des Projekts *Garage* in die fünf Bauteile *StB-Fundament 1*, *StB-Fundament 2*, *KS-Wand 1*, *KS-Wand 2* und *StB-Decke 1* führte beispielsweise zu einem Aufwand von 41. Anstatt eine Einteilung

in fünf Bauteile vorzunehmen, wäre auch eine Einteilung in drei Bauteile, KS-Wand mit *StB-Fundament 1*, KS-Wand mit *StB-Fundament 2* und *StB-Decke 1* plausibel gewesen und hätte einen Aufwand von 36 verursacht. Eine Prognose, welche Einteilung zu einem Minimum an Spezifikationsaufwand führt, ist für komplexe Bauabläufe nicht möglich. Erst nachdem ein Prozess spezifiziert ist, lässt sich rückblickend ermitteln, welche Einteilung zu einem Minimum an Spezifikationsaufwand führt.

Die Ermittlung einer optimalen Einteilung des Projekts in Leistungsgegenstände stellt eine Umkehrung des in Kapitel 3 vorgestellten Modellierungsansatzes dar. Der dort beschriebene Ansatz zeigt, wie schrittweise das Prozessmodell entwickelt wird, um daraus den Prozess abzubilden. Die Umkehrung des Ansatzes bedeutet, dass von einem gegebenen Prozess ausgegangen wird, für den ein Prozessmodell mit bestimmten Eigenschaften gesucht ist. Die Zielmenge sind alle Prozessmodelle, die sich auf den gegebenen Prozess abbilden lassen. Die gesuchte Lösungsmenge umfasst diejenigen Prozessmodelle der Zielmenge, für die der Spezifikationsaufwand nach Gleichung 3.2 minimal ist.

Ziel der Rückrechnung ist es, anhand der Muster von Teilprozessen disjunkte Zerlegungen des Gesamtprozesses zu ermitteln, wobei jede unterschiedliche Zerlegung eine Variante eines geeigneten Prozessmodells darstellt. Jede gefundene Zerlegung kann somit durch die Abbildung auf einen Spezifikationsaufwand gewichtet werden. Unterschiedliche Zerlegungen lassen sich vergleichen.

Die Ermittlung eines optimalen Prozessmodells für einen gegebenen Prozess lässt sich als Suche nach Mustern in einem Graphen auffassen. Um das Verfahren zu beschreiben, ist es zweckmäßig, die fachlich begründeten Begriffe, wie Prozess, Leistungsprozess, Leistungsprozessvorlage u.s.w. zu abstrahieren und die Problemstellung graphentheoretisch zu betrachten.

Das Vorgehen wird am Beispiel des Projekts *Garage* erläutert. Zugrunde liegt der in Abbildung 3.23 dargestellte Prozess.

4.1 Mustererkennung

Der Begriff Mustererkennung beschreibt die Fähigkeit, in einer Menge an Informationen Ähnlichkeiten, Regelmäßigkeiten oder Wiederholungen zu erkennen. Ein typisches Beispiel ist die menschliche Wahrnehmung. Sinnesreize werden kontinuierlich erfasst, abstrahiert und ausgewertet. Diese kognitive Fähigkeit manifestiert sich u.a. in Form von Gesichtserkennung, Spracherkennung und der Texterkennung.

In der Informatik wird der Begriff Mustererkennung im Zusammenhang mit verschiedenartigen Zielsetzungen verwendet. Ein Richtung befasst sich mit dem automatisierten Erkennen, Klassifizieren und Verarbeiten von Signalen. Beispiele für die Anwendung von Mustererkennung sind Spracherkennung, optische Zeichenerkennung, Robotik, Biometrie (Gesichtserkennung, Fingerabdrücke...) und Bilderkennung. Mustererkennung in Datenbanken wird auch in der Datenanalyse oder im Data-Mining genutzt.

Diese Verfahren zielen darauf ab, in einem ersten Schritt die gegebene Eingangsinformationen in eine verarbeitbare Form zu transformieren, um sie dann in einem anschließenden Schritt anhand von charakteristischen Merkmalen mit bekannten Daten zu vergleichen. Die Merkmalsvergleiche werden entsprechend ihrer Übereinstimmungen gewichtet. Liegen die Gewichtungen innerhalb festgelegter Grenzen, kann anhand der bekannten Eigenschaften der Vergleichsmerkmale eine Klassifikation des Untersuchungsgegenstands erfolgen. Die verschiedenen Verfahren unterscheiden sich im Wesentlichen in ihren Klassifikationsverfahren. Drei wesentliche Ansätze werden unterschieden¹.

- *Syntaktische Mustererkennung*: Bei der syntaktischen Mustererkennung werden die zu vergleichenden Objekte durch Folgen von Symbolen beschrieben, so dass Objekte der gleichen Kategorie die gleichen Beschreibungen aufweisen.
- *Statistische Mustererkennung*: Bei der statistischen Mustererkennung werden für die Objekte Wahrscheinlichkeiten bestimmt. Die Wahrscheinlichkeit gibt an, ob das Objekt eher der einen oder der anderen Kategorie angehört. Das Objekt wird dann in die Kategorie mit der höchsten Wahrscheinlichkeit eingeordnet.
- *Strukturelle Mustererkennung*: Die strukturelle Mustererkennung fasst die syntaktische und die statistische Mustererkennung in einem Verfahren zusammen, indem zunächst Merkmale für die Objekte festgelegt werden. Anschließend führen diese dann durch statistische Verfahren zu einer Zuordnung des Objektes zu einer Kategorie.

Da es im Rahmen dieser Arbeit nicht um eine Form der Informationsabstraktion mit Merkmalerkennung und anschließender Klassifikation geht, wird auf diese Ansätze nicht weiter eingegangen.

¹ [Niemann, 2003, In einer umfassenden Darstellung werden alle Aspekte der Vorverarbeitung, Merkmalgewinnung und Klassifikation behandelt.]

Eine weitere Richtung befasst sich mit der Betrachtung von Isomorphieproblemen auf Graphen. Hierbei steht das Erkennen von Strukturen in Graphen im Mittelpunkt. Werden beispielsweise Graphen zur Visualisierung eingesetzt, so erschließt sich einem Betrachter der Inhalt erheblich einfacher, wenn strukturgleiche Ausschnitte gleichartig dargestellt werden. Das Identifizieren von strukturgleichen Ausschnitten ist eine Problematik, die durch Subgraphisomorphismen untersucht wird. Ein weiteres Anwendungsgebiet ist die Chemie, wo es darum geht Moleküle auf das Vorkommen typischer Strukturen hin zu untersuchen. Eine Übersicht über Subgraphisomorphismen liefert Messmer und Bunke [2000]².

Isomorphismus: Gegeben seien die Strukturen³ $(A; R_a)$ und $(Z; R_z)$. Eine Abbildung⁴ $f : A \rightarrow Z$ wird als strukturerhaltend bezeichnet, wenn folgender Ausdruck wahr ist: $f(a_1 R_a a_2) = f(a_1) R_z f(a_2)$. Ist eine Abbildung strukturerhaltend und bijektiv, dann bezeichnet man die Strukturen als isomorph. Die Abbildung f wird in diesem Fall als *Isomorphismus* bezeichnet. Durch die Betrachtung von Isomorphismen werden mathematische Strukturen, u.a. Graphen, vergleichbar gemacht.

Ein Subgraphisomorphismus $f : A \rightarrow Z$ zwischen zwei Graphen $G_a = (A; R_a)$ und $G_z = (Z; R_z)$ ist ein Isomorphismus, der den Graphen G_a auf einen Untergraphen von G_z abbildet.

Im Rahmen dieser Arbeit werden Isomorphismen betrachtet, um die Gleichheit der Muster⁵ von Teilprozessen zu ermitteln. Es wird auf die Anwendung bekannter Verfahren zur Bestimmung von Untergraphen verzichtet, um den spezifischen Eigenschaften der gegebenen Problemstellung besser gerecht werden zu können. Insbesondere wird auf die Ermittlung der Untergraphen, für die die Isomorphismen angewendet werden, eine an die Problemstellung angepasste Vorgehensweise entwickelt. Die Anzahl der zu betrachtenden Untergraphen soll möglichst klein gehalten werden, um den Aufwand in Grenzen zu halten. Gleichzeitig ist sicherzustellen, dass der Lösungsraum vollständig betrachtet wird.

² [Messmer und Bunke, 2000, Efficient subgraph isomorphism detection: a decomposition approach]

³ Siehe Anhang: A Begriffe aus der Graphentheorie ↑ Struktur

⁴ Siehe Anhang: A Begriffe aus der Graphentheorie ↑ Abbildung

⁵ siehe Abschnitt 4.2.5

4.2 Abstraktion der Problemstellung

In diesem Abschnitt wird die fachliche Problemstellung auf eine graphentheoretische Problemstellung abgebildet. Wesentliche Grundbegriffe werden eingeführt, auf die bei der Beschreibung des Ansatzes Bezug genommen wird.

4.2.1 Der Prozess als Ereignisgraph

Ausgangspunkt der Betrachtung sind realisierte Prozesse, die sich in ihrer Modellierung als fachlich richtig und für das Controlling als zweckmäßig herausgestellt haben. Als charakteristisches Merkmal realisierter Prozesse wird die Eigenschaft festgelegt, gerichtet und zyklensfrei zu sein. Stellvertretend wird hier der in Abbildung 3.23 gezeigte Prozess zur Veranschaulichung herangezogen. Dargestellt ist eine eEPK, bestehend aus den Elementen vom Typ Ereignis und Funktion, sowie den Kanten, die diese Elemente verknüpfen. Fasst man die Ereignisse in der Menge M_E , die Funktionen in der Menge M_F , die Verknüpfungen von Ereignissen und Funktionen in der Relation R_{F-E} und die Verknüpfungen von Funktionen und Ereignissen in der Relation R_{E-F} zusammen, dann lässt sich der Prozess als bipartiter Graph ausdrücken.

$$\begin{aligned} G &= (M_F, M_E; R_{E-F}, R_{F-E}) \\ R_{E-F} &= \{(e, f) \in M_E \times M_F \mid e \text{ ist Voraussetzung von } f\} \\ R_{F-E} &= \{(f, e) \in M_F \times M_E \mid f \text{ führt zu } e\} \end{aligned} \quad (4.1)$$

G	Bipartiter Graph
M_F	Menge der Funktionen
M_E	Menge der Ereignisse
R_{E-F}	Menge an Verknüpfungen, die ein Ereignis als Voraussetzung für eine Funktion definieren
R_{F-E}	Menge an Verknüpfungen, die ein Ereignis als Ergebnis für eine Funktion definieren

Der bipartite Graph G lässt sich überführen in einen schlichten Ereignisgraphen G_E . Dieser enthält genau diejenigen Knotenpaare $(e_1, e_2) \in M_E \times M_E$, für die es jeweils einen Knoten $f \in M_F$ gibt, der die Knoten e_1 und e_2 über die Kanten (e_1, f) und (f, e_2) in G verbindet. Die Knotenpaare werden in der Relation R_{E-E} zusammengefasst. Weil eine Funktion immer

nur ein Ereignis als Nachfolger hat, geht durch diese Umformung keine Strukturinformation des Prozesses verloren. Die Betrachtungen, die am Ereignisgraphen durchgeführt werden, gelten gleichermaßen für den Prozess. Jeder Ereignisgraph ist ein endlicher, azyklischer, schlichter Graph.

$$\begin{aligned}
 G_E &= (M_E ; R_{E-E}) \\
 R_{E-E} &= \{(e_1, e_2) \in M_E \times M_E \mid \\
 &\quad \bigvee_{f \in M_F} [(e_1, f) \in R_{E-F} \wedge (f, e_2) \in R_{F-E}]\}
 \end{aligned}
 \tag{4.2}$$

G_E Ereignisgraph

M_E Menge der Knoten des schlichten Graphen

R_{E-E} Menge der Kanten des schlichten Graphen

Ein Ereignisgraph kann aus einer oder aus mehreren Zusammenhangskomponenten bestehen. Abbildung 4.1 zeigt den Ereignisgraphen G_E für

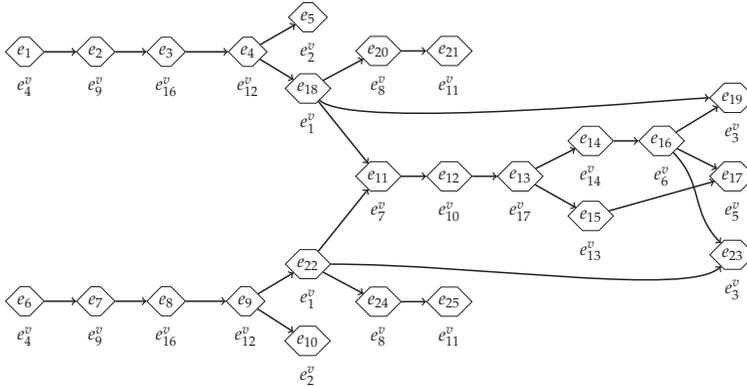


Abbildung 4.1: Projekt Garage - Ereignisgraph

das Projekt *Garage*. In der Darstellung des Ereignisgraphen sind innerhalb der Knoten die Identifikatoren der Ereignisse abgebildet. Unterhalb jedes Knotens ist eine Beschriftung abgebildet. Die Beschriftung eines Knotens ergibt sich aus dem Identifikator seiner Ereignisvorlage. Da unterschiedliche Ereignisse von derselben Vorlage abstammen können, gibt es unterschiedliche Knoten mit gleicher Beschriftung.

Nicht in der Abbildung 4.1 dargestellt sind die Identifikatoren und Beschriftungen von Kanten. Als Kantenidentifikator bzw. -beschriftung werden die Identifikatoren bzw. Beschriftungen der inzidenten Knoten verwendet. Ein Identifikator ergibt sich durch Aneinanderreihen der Identifikatoren, getrennt durch ein definiertes Trennzeichen (siehe Abschnitt 4.2.4). Analog lässt sich eine Beschriftung erzeugen. Dieses Vorgehen stellt sicher, dass alle Kanten eindeutig identifiziert werden können, ihre Beschriftung sich aber für diejenigen Kanten wiederholt, für die die Vorlagen der in Beziehung stehenden Ereignisse gleich waren.

Für jeden Knoten und für jede Kante existiert somit ein Identifikator und eine Beschriftung. Identifikatoren und Beschriftungen werden genutzt, um beliebige Teilgraphen oder Untergraphen eines Ereignisgraphen auf Zeichenketten abzubilden. Diese Zeichenketten können zur Identifikation von Teilprozessen oder zur Strukturanalyse verwendet werden. Eindeutige Bezeichner sind die Voraussetzung für die Betrachtung von Wegen in Graphen und für die Anwendung von Wegalgebren⁶.

4.2.2 Zerlegung eines Ereignisgraphen

Sei $G_E = (M_E; R_{E-E})$ der Ereignisgraph eines Projekts. Entfernt man aus R_{E-E} eine Menge S_{E-E} an Kanten, so entsteht ein Teilgraph T_E . Für unterschiedliche Mengen $S_{E-E,s}$ entstehen unterschiedliche Teilgraphen $T_{E,s}$. Der Index s stellt einen Bezug zu der konkreten Menge $S_{E-E,s}$ her. Vergrößert sich durch das Entfernen der Kanten die Anzahl der Zusammenhangskomponenten⁷, so stellt der Teilgraph $T_{E,s}$ eine Zerlegung $Z_{E,s}$ des ursprünglichen Ereignisgraphen G_E dar. Eine Zerlegung ist ein Ereignisgraph mit der Knotenmenge M_E und der Kantenmenge $Z_{E-E,s}$. Die Kantenmenge Z_{E-E} berechnet sich als das Komplement der Menge $S_{E-E,s}$ bezüglich R_{E-E} .

Abbildung 4.2 zeigt einen Teilgraphen⁸ $T_{E,s}$ des Ereignisgraphen G_E des Projekts *Garage*. Entfernt wurden die Kanten (e_{16}, e_{19}) , (e_{16}, e_{23}) , (e_{18}, e_{11}) und (e_{22}, e_{11}) . Diese Kanten werden in der Menge $S_{E-E,s}$ zusammengefasst. Der Teilgraph $T_{E,s}$ stellt eine Zerlegung $Z_{E,s}$ von G_E in drei Zusammenhangskomponenten dar.

$$\begin{aligned} Z_{E,s} &= (M_E; Z_{E-E,s}) \\ Z_{E-E,s} &= R_{E-E} \setminus S_{E-E,s} \end{aligned} \tag{4.3}$$

⁶ [Pahl und Damrath, 2000, Wegalgebra, S. 614 ff.]

⁷ Siehe Abschnitt A↑ Zusammenhang

⁸ Siehe Abschnitt A↑ Teilgraph

- G_E Ereignisgraph
- $Z_{E,s}$ eine Zerlegung von G_E
- R_{E-E} Menge der Kanten des Ereignisgraphen G_E
- $S_{E-E,s}$ eine Menge an Kanten, die aus G_E entfernt wurden
- $Z_{E-E,s}$ Komplement der Menge $S_{E-E,s}$ bezüglich R_{E-E}

Zusammenhangskomponenten werden im Folgenden als Komponenten bezeichnet. Jede Komponente einer Zerlegung ist ein Untergraph von G_E . Um Komponenten unterscheiden zu können, wird der Index k eingeführt. Eine Komponente wird mit $K_{E,sk}$ bezeichnet. Die Komponenten aus Abbildung 4.2 seien von oben nach unten mit $K_{E,s1}$, $K_{E,s2}$ und $K_{E,s3}$ bezeichnet und in der Menge $M_{K,s}$ zusammengefasst.

$$K_{E,sk} = (M_{E,sk} ; Z_{E-E,sk}) \quad (4.4)$$

- $M_{E,sk}$ Menge der Knoten der Komponente, Teilmenge von M_E
- $Z_{E-E,sk}$ Menge der Kanten der Komponente, Teilmenge von $Z_{E-E,s}$

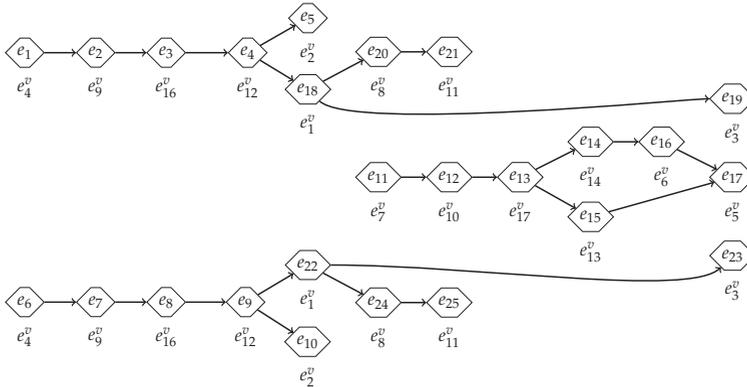


Abbildung 4.2: Ein Teilgraph von G_E , bestehend aus drei Untergraphen

Es gilt:

$$M_E = \bigcup_{K_{E,sk} \in M_{K,s}} M_{E,sk}$$

$$Z_{E-E,s} = \bigcup_{K_{E,sk} \in M_K} Z_{E-E,sk}$$

Zum Unterschied zwischen Teilgraph und Untergraph: Ein Teilgraph entsteht aus einem Graphen durch Entfernen von Kanten. Ein Untergraph entsteht aus einem Graphen durch Entfernen von Knoten und Kanten. Durch das Entfernen eines Knotens werden auch inzidente Kanten entfernt.

4.2.3 Prozessmodell ausgedrückt als Zerlegung

Betrachtet man einen Ereignisgraphen G_E und eine seiner Zerlegungen $Z_{E,s}$, dann beschreibt das Gebilde aus $(G_E, Z_{E,s})$ ein Prozessmodell ω . Jede Komponente $K_{E,sk}$ aus $M_{K,s}$ stellt einen Teilprozess dar. Jede Kante aus $S_{E-E,s}$ stellt eine zusätzliche Verknüpfung dar, die zu dem Gesamtprozess dazugehört, aber nicht durch die Teilprozesse berücksichtigt ist. Die Menge Ω sei definiert als diejenige Menge, die alle Prozessmodelle ω enthält, die sich aus unterschiedlichen Zerlegungen von G_E beschreiben lassen.

$$\omega = (G_E, Z_{E,s}) \tag{4.5}$$

ω ein von der Zerlegung $Z_{E,s}$ abhängiges Prozessmodell

Ω Menge aller Prozessmodelle von G_E

4.2.4 Identifikation von Ereignisgraphen

Bei der formalen Analyse eines Prozesses, der als Ereignisgraph G_E gegeben ist, entstehen zur Laufzeit von Algorithmen unterschiedliche Zerlegungen $Z_{E,s}$. Jede Zerlegung $Z_{E,s}$ teilt den Prozess in unterschiedliche Komponenten $M_{K,s}$ ein. Prozesse, Zerlegungen und Komponenten lassen sich alle als Ereignisgraphen darstellen. Um unterschiedliche Prozesse, Zerlegungen oder Komponenten zur Laufzeit von Algorithmen auf ihre Identität hin überprüfen zu können, ist es erforderlich, eine Abbildungsvorschrift zu formulieren, die einen Ereignisgraphen reproduzierbar auf denselben Identifikator abbildet.

In diesem Abschnitt wird eine Abbildung f_I eingeführt, die einen beliebigen Ereignisgraphen eindeutig und reproduzierbar auf eine Zeichenkette abbildet. Voraussetzung für das Generieren von eindeutigen Identifikatoren ist, dass jeder Knoten einen eindeutigen Identifikator hat. Dies ist für Knoten, die zu demselben Prozess gehören, der Fall. Bildet man den Prozess auf einen Ereignisgraphen G_E ab, so beziehen sich die unterschiedlichen Zerlegungen bzw. Komponenten immer auf dieselbe Knotenmenge M_E . Die Kantenidentifikatoren brauchen nicht berücksichtigt zu werden. Diese setzen sich aus den Knotenidentifikatoren zusammen.

Jeder Ereignisgraph ist ein endlicher, azyklischer, schlichter Graph. Ein eindeutiger und reproduzierbarer Identifikator muss die Struktur des Graphen berücksichtigen. Um die Struktur des Graphen zu berücksichtigen, werden Wege betrachtet. Jeder Weg lässt sich als Folge $\langle a, \dots, e \rangle$ von Knoten beschreiben. Sei $S \subseteq M_E$ die Menge der Knoten ohne Vorgänger (Startknoten), und sei $E \subseteq M_E$ die Menge der Knoten ohne Nachfolger (Endknoten), dann existiert in der transitiven Hülle $\langle R_{E-E} \rangle_t$ eine endliche Menge an Wegen M_W , die die Menge der Startknoten S mit der Menge der Endknoten E verbindet. Diese Wegmenge beschreibt einen Ereignisgraphen eindeutig. Die transitive Hülle $\langle R_{E-E} \rangle_t$ enthält alle Wege von einem Knoten x zu einem Knoten y über beliebig viele Zwischenstationen.

$$M_W = \{ \langle a, \dots, e \rangle \mid (a, e) \in \langle R_{E-E} \rangle_t \wedge a \in S \wedge e \in E \} \quad (4.6)$$

M_W	Menge der Wege
$\langle a, \dots, e \rangle$	ein Weg von Knoten a nach Knoten e
$\langle R_{E-E} \rangle_t$	transitive Hülle von R_{E-E}
S	Menge der Startknoten
E	Menge der Endknoten

Ordnet man die Knotenidentifikatoren entsprechend ihrer Position in der Folge $\langle a, \dots, e \rangle$ in einer Zeichenkette an, so erhält man für jeden Weg einen Wegidentifikator. Sortiert man die Wegidentifikatoren alphabetisch, erhält man einen eindeutigen, reproduzierbaren Identifikator für einen Ereignisgraphen. Unterschiedliche Trennzeichen werden verwendet, um Knoten bzw. Wege voneinander zu trennen. Gleichung 4.7 zeigt die Abbildung f_I von Ereignisgraphen einer Zerlegung auf Identifikatoren.

$$f_I : M_{G_E} \rightarrow I \quad (4.7)$$

- M_{G_E} Menge an Ereignisgraphen, zulässige Elemente von M_{G_E} sind: G_E , $Z_{E,s}$ und $K_{E,sk}$
- I Menge der Identifikatoren

Stellvertretend wird für die Komponente $K_{E,1}$ aus Abbildung 4.2 der Identifikator bestimmt. Die Komponente hat einen Startknoten e_1 und drei Endknoten e_5, e_{19}, e_{21} . Hieraus ergeben sich drei Wege. Sortiert man die Wegidentifikatoren alphabetisch, erhält man den Komponentenidentifikator.

$$\begin{aligned}
 f_I(K_{E,1}) = & e_1 - e_2 - e_3 - e_4 - e_5, \\
 & e_1 - e_2 - e_3 - e_4 - e_{18} - e_{19}, \\
 & e_1 - e_2 - e_3 - e_4 - e_{18} - e_{20} - e_{21}
 \end{aligned}$$

Vergleicht man zwei Komponenten, die sich aus unterschiedlichen Zerlegungen ergeben, dann lässt sich anhand ihrer Identifikatoren ermitteln, ob die betrachteten Komponenten gleich sind oder nicht.

4.2.5 Muster von Ereignisgraphen

Jedem Ereignisgraphen G_E lässt sich ein Muster zuordnen. Das Muster eines Ereignisgraphen ist eine Zeichenkette, die sich aus den Knotenbeschriftungen abbilden lässt. Betrachtet werden alle Wege M_W analog zu Abschnitt 4.2.4, die die Menge der Startknoten S von G_E mit der Menge der Endknoten E von G_E verbindet. Verwendet man bei der Abbildung eines Weges auf eine Zeichenkette nicht die Knotenidentifikatoren, sondern die Knotenbeschriftungen, so erhält man für jeden Weg ein Wegmuster. Sortiert man die Wegmuster alphabetisch, erhält man ein reproduzierbares Graphenmuster. Gleichung 4.8 zeigt die Abbildung f_P von Ereignisgraphen auf Muster. Das Beispiel zeigt das Muster der Komponente $K_{E,1}$.

$$f_P : M_{G_E} \rightarrow P \quad (4.8)$$

- M_{G_E} Menge an Ereignisgraphen, zulässige Elemente von M_{G_E} sind: G_E , $Z_{E,s}$ und $K_{E,sk}$
- P Menge der Muster (Pattern)

Beispiel:

$$\begin{aligned} f_P(K_{E,1}) &= e_4^v - e_9^v - e_{16}^v - e_{12}^v - e_1^v - e_2^v, \\ &e_4^v - e_9^v - e_{16}^v - e_{12}^v - e_1^v - e_3^v, \\ &e_4^v - e_9^v - e_{16}^v - e_{12}^v - e_1^v - e_8^v, -e_{11}^v \end{aligned}$$

4.2.6 Klassen von Komponenten

Sei $M_{K,s}$ eine Menge an Komponenten einer Zerlegung $Z_{E,s}$. Die Abbildung f_P nach Gleichung 4.8 kann als eine Relation aufgefasst werden, die eine Komponente $K_{E,sk}$ mit einem Muster verknüpft. Diese Relation kann genutzt werden, um Komponenten in Äquivalenzklassen einzuteilen. Jede Äquivalenzklasse stellt diejenige Teilmenge von $M_{K,s}$ dar, für die die Abbildung f_P gleich ist. Die Abbildung f_P ist surjektiv. Das heißt, das Produkt $f_P \circ f_P^T$ für die Menge $M_{G_E} = M_{K,s}$ liefert eine Äquivalenzrelation $\ddot{A}_{K,s} \subseteq M_{K,s} \times M_{K,s}$. Enthält die Äquivalenzrelation $\ddot{A}_{K,s}$ ein Paar $(M_{E,sk}, M_{E,sj})$, so lassen sich die Komponenten $M_{E,sk}$ und $M_{E,sj}$ auf dasselbe Muster abbilden und werden als äquivalent betrachtet. Eine Äquivalenzklasse wird durch einen beliebigen Repräsentanten beschrieben.

$$[K_{E,sk}] := \{x \in M_{K,s} \mid (K_{E,sk}, x) \in \ddot{A}_{K,s}\} \quad (4.9)$$

$[K_{E,sk}]$ Äquivalenzklasse
 $K_{E,sk}$ eine Komponente (Repräsentant)
 $M_{K,s}$ Menge der Komponenten einer Zerlegung $Z_{E,s}$
 $\ddot{A}_{K,s}$ Äquivalenzrelation

Beispiel:

$$\begin{aligned} [K_{E,s1}] &= \{K_{E,s1}, K_{E,s3}\} \\ [K_{E,s2}] &= \{K_{E,s2}\} \end{aligned}$$

Die Quotientenmenge $M_{K,s} / \ddot{A}_{K,s}$ ist eine Menge, die alle Äquivalenzklassen einer Zerlegung als Elemente enthält.

$$M_{K,s} / \ddot{A}_{K,s} := \{[x] \mid x \in M_{K,s}\} \quad (4.10)$$

$M_{K,s} / \ddot{A}_{K,s}$ Quotientenmenge
 x eine Komponente (Repräsentant)
 $M_{K,s}$ Menge der Komponenten einer Zerlegung
 $\ddot{A}_{K,s}$ Äquivalenzrelation

Beispiel:

$$M_{K,s}/\dot{A}_{K,s} = \{[K_{E,1}], [K_{E,2}]\}$$

Jede Komponente $K_{E,sk}$ aus $M_{K,s}$ stellt einen Teilprozess dar. Jede Äquivalenzklasse $[K_{E,sk}]$ stellt eine Teilmenge von Teilprozessen dar, die sich aus derselben Vorlage ableiten lässt. Die Anzahl der Elemente in einer Äquivalenzklasse gibt an, wie viele Teilprozesse dieselbe Vorlage nutzen können. Die Anzahl der Elemente einer Quotientenmenge $M_{K,s}/\dot{A}_{K,s}$ gibt an, wie viele Prozessvorlagen für eine Zerlegung $Z_{E,s}$ erforderlich sind.

4.2.7 Bewertung eines Ereignisgraphen

Als Bewertung eines Ereignisgraphen wird seine Abbildung a auf eine dimensionslose Ganzzahl eingeführt. Die Abbildung berechnet sich als Summe aus der Anzahl der Knoten und der Anzahl der Kanten. Die Bewertung a ist ein Maß für den Aufwand ohne Berücksichtigung von gleichartigen Teilprozessen.

$$a : M_{G_E} \rightarrow N \tag{4.11}$$

$$a(G_E) = |M_E| + |R_{E-E}|$$

- a Bewertung eines Ereignisgraphen
- M_{G_E} Menge an Ereignisgraphen
- G_E ein Ereignisgraph
- N Menge der natürlichen Zahlen
- M_E Menge der Ereignisse eines Ereignisgraphen
- R_{E-E} Menge der Kanten eines Ereignisgraphen

Beispiele:

$$a(G_{Garage}) = 23 + 27 = 50$$

$$a(Z_{E,s}) = 25 + 23 = 48$$

$$a(K_{E,s1}) = 8 + 7 = 15$$

4.2.8 Bewertung einer Zerlegung

Eine Zerlegung besteht aus einer Menge an Komponenten $M_{K,s}$, die sich in ein Mengensystem aus Äquivalenzklassen einteilen lässt. Jede Äquivalenzklasse stellt eine Teilmenge von $M_{K,s}$ dar. Jedes Element von $M_{K,s}$ ist

in genau einer Äquivalenzklasse enthalten, die Teilmengen sind disjunkt und keine der Teilmengen ist leer. Ein Repräsentantensystem $M_{R,s}$ ist eine Teilmenge von $M_{K,s}$, die genau eine Komponente aus jeder Äquivalenzklasse enthält.

Als Bewertung einer Zerlegung wird ihre Abbildung a_Z auf eine dimensionslose Ganzzahl eingeführt. Die Bewertung berechnet sich aus der Summe der Aufwände der Komponenten des Repräsentantensystems, erweitert um die Anzahl der Komponenten in $M_{K,s}$. Die Bewertung a_Z ist ein Maß für den Aufwand, unter Berücksichtigung sich wiederholender Teilprozesse.

$$a_Z : M_{Z_{E,s}} \rightarrow N$$

$$a_Z(Z_{E,s}) = |M_{K,s}| + \sum_{x \in M_{R,s}} a(x) \quad (4.12)$$

a_Z	Bewertung einer Zerlegung
$M_{Z_{E,s}}$	Menge an Zerlegungen
$Z_{E,s}$	eine Zerlegung
N	Menge der natürlichen Zahlen
$M_{K,s}$	Menge an Komponenten einer Zerlegung
$M_{R,s}$	Repräsentantensystem

Beispiel:

$$a_Z(Z_{E,s}) = |M_{K,s}| + a(K_{E,1}) + a(K_{E,2}) = 3 + 15 + 14 = 32$$

4.2.9 Normierte Bewertung einer Zerlegung

Um Zerlegungen, die sich auf unterschiedliche Knoten und Kantenmengen beziehen, zweckmäßig vergleichen zu können, wird eine Normierung eingeführt. Sei $M_{K,s}$ eine Menge an Komponenten einer Zerlegung $Z_{E,s}$. Jede Komponente in $M_{K,s}$ stellt einen Teilprozess dar. Die Zerlegung $Z_{E,s}$ lässt sich auf unterschiedliche Art und Weise bewerten. Eine Bewertung nach Gleichung 4.12 beschreibt den Spezifikationsaufwand unter Berücksichtigung von wiederkehrenden Teilprozessen. Eine Bewertung nach Gleichung 4.11 beschreibt den Spezifikationsaufwand ohne die Berücksichtigung von wiederkehrenden Teilprozessen. Bezieht man die genannten Bewertungen einer Zerlegung $Z_{E,s}$ aufeinander, so erhält man eine Normierung n_Z des Spezifikationsaufwands auf den Wertebereich von 0 bis 1, wobei 0 nicht erreicht wird. Die normierte Bewertung n_Z ist ein Maß,

das den relativen Effekt beschreibt, der sich aus der Berücksichtigung wiederkehrender Teilprozesse ergibt.

$$n_Z : M_{Z_{E,s}} \rightarrow \mathbb{R} \quad \text{mit } 0 < n_Z \leq 1$$

$$n_Z(Z_{E,s}) = \frac{a_Z(Z_{E,s})}{a(Z_{E,s})} \quad \text{wobei } a(Z_{E,s}) \geq a_Z(Z_{E,s}) \quad (4.13)$$

n_Z normierte Bewertung einer Zerlegung
 $M_{Z_{E,s}}$ Menge an Zerlegungen
 $Z_{E,s}$ eine Zerlegung
 \mathbb{R} Menge der reellen Zahlen zwischen 0 und 1

a_Z Bewertung einer Zerlegung unter Berücksichtigung von wiederkehrenden Teilprozessen nach Gleichung 4.12

a Bewertung einer Zerlegung ohne Berücksichtigung wiederkehrender Teilprozesse nach Gleichung 4.11

Beispiel:

$$n_Z(Z_{E,s}) = \frac{a_Z(Z_{E,s})}{a(Z_{E,s})} = \frac{32}{48} = 0.\bar{6}$$

Ist der Wert $n_Z = 1$, dann ergibt sich keine Einsparung. Ist der Wert $n_Z < 1$, dann existieren wiederkehrende Teilprozesse, die zu einer Einsparung führen. Der Abstand von n_Z zu dem Wert 1 kann als Einsparpotential aufgefasst werden. Vergleicht man die normierte Bewertung n_Z unterschiedlicher Zerlegungen, ist eine Aussage darüber möglich, für welche Zerlegung sich die Berücksichtigung wiederkehrender Teilprozesse stärker auswirkt.

4.2.10 Bewertung eines Prozessmodells

Das Gebilde $(G_E, Z_{E,s})$ aus Ereignisgraph G_E und Zerlegung $Z_{E,s}$ beschreibt ein Prozessmodell ω . Als Bewertung eines Prozessmodells ω wird dessen Abbildung a_ω auf eine dimensionslose Ganzzahl eingeführt. Die Bewertung berechnet sich aus der Summe des Aufwands für die Zerlegung $a_Z(Z_{E,s})$ und der Anzahl der Kanten $|S_{E-E,s}|$. Die Bewertung a_ω ist

ein Maß für den Aufwand, unter Berücksichtigung wiederkehrender Teilprozesse und der zusätzlich zu spezifizierenden Kanten.

$$a_\omega : \Omega \rightarrow N$$

$$a_\omega(\omega) = a_Z(Z_{E,s}) + |S_{E-E,s}| \quad (4.14)$$

a_ω Bewertung eines Prozessmodells

Ω Menge an Prozessmodellen

ω ein Prozessmodell

N Menge der natürlichen Zahlen

$Z_{E,s}$ eine Zerlegung

$S_{E-E,s}$ Menge der entfernten Kanten:

$$S_{E-E,s} = R_{E-E} \setminus Z_{E-E,s}$$

R_{E-E} Menge der Kanten des Ereignisgraphen G_E

$Z_{E-E,s}$ Menge der Kanten, die zu einer Komponente gehören

Beispiel:

$$a_\omega(\omega) = 32 + 4 = 36$$

4.3 Optimale Zerlegung

Abbildung 4.3 zeigt einen Ereignisgraphen. Jeder Knoten und jede Kante hat einen Identifikator und eine Beschriftung. Knotenbeschriftungen sind als Kleinbuchstaben dargestellt und Kantenbeschriftungen sind als Großbuchstaben dargestellt. Aus Gründen der Übersichtlichkeit wird auf das Zusammensetzen der Kantenidentifikatoren und Kantenbeschriftungen nach Abschnitt 4.2.4 verzichtet. Dieser Graph dient als Beispielgraph, an dem Vorgehensweisen und Zusammenhänge erläutert werden. Der Graph hat keinen fachlichen Hintergrund. Er ist bewusst klein gehalten, um nicht unnötig die Übersicht zu erschweren.

4.3.1 Optimierungsproblem

Ein Ereignisgraph G_E lässt sich auf unterschiedliche Weise zerlegen. Die Menge der möglichen Zerlegungen sei in der Menge Ω zusammengefasst.

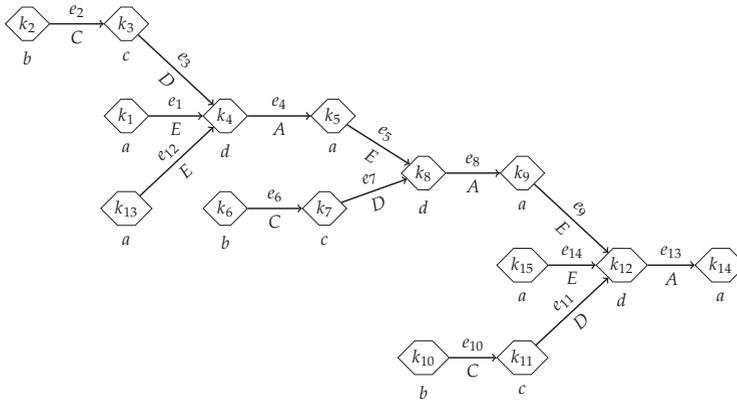


Abbildung 4.3: Beispielgraph für die Mustererkennung

Die Bestimmung der optimalen Zerlegungen lässt sich als Minimierungsproblem formulieren. Gegeben ist der Lösungsraum Ω und die Bewertungsfunktion $a_\omega : \Omega \rightarrow N$. Gesucht ist die Teilmenge $\Omega_{opt} \subseteq \Omega$, für die die Bewertung a_ω der Elemente $\omega \in \Omega$ minimal ist.

$$\begin{aligned} a_{\omega,opt} &= \min\{a_\omega(\omega) \mid \omega \in \Omega\} \\ \Omega_{opt} &= \{\omega_{opt} \in \Omega \mid a_\omega(\omega_{opt}) = a_{\omega,opt}\} \end{aligned} \quad (4.15)$$

- Ω_{opt} Menge an Prozessmodellen, für die die Bewertung $a_\omega(\omega)$ minimal ist
- $a_{\omega,opt}$ minimale Bewertung, nach der Auswertung aller Elemente von Ω
- Ω Menge an Prozessmodellen
- ω ein Prozessmodell
- ω_{opt} ein Prozessmodell mit minimaler Bewertung

Bei dem Minimierungsproblem handelt es sich um ein Suchproblem. Nicht der minimale Wert $a_{\omega,opt}$ der Abbildung $a_\omega(\omega)$ ist von Interesse, sondern das oder die Elemente $\omega \in \Omega$, die sich auf das Minimum $a_{\omega,opt}$ abbilden lassen.

Der Aufwand der Suche ist abhängig von der Anzahl der Prozessmodelle in Ω . Die maximale Anzahl der Prozessmodelle, die für einen Ereignisgraphen $G_E = (M_E; R_{E-E})$ existiert, ist abhängig von der Anzahl der

Kanten in R_{E-E} . Sei $f : N \rightarrow R_{E-E}$ mit $f(i) = r_i$ und $1 \leq i \leq n$ eine Abbildung, die die Kanten des Ereignisgraphen indiziert, dann lässt sich jedes Prozessmodell $\omega = (G_E, Z_{E,S})$ ausdrücken als Vektor \mathbf{w} .

$$\mathbf{w} = \begin{array}{c} r_1 \\ \vdots \\ r_i \\ \vdots \\ r_n \end{array} \begin{array}{|c|} \hline w_1 \\ \hline \vdots \\ \hline w_i \\ \hline \vdots \\ \hline w_n \\ \hline \end{array} \quad \begin{array}{l} r_i \in Z_{E-E,S} \Rightarrow w_i = 1 \\ r_i \notin Z_{E-E,S} \Rightarrow w_i = 0 \end{array} \quad (4.16)$$

- \mathbf{w} boolescher Vektor, der ein Prozessmodell beschreibt
- r_i indizierte Kante aus R_{E-E}
- w_i Zugehörigkeitskoeffizient

Die Anzahl unterschiedlicher Vektoren \mathbf{w} beschreibt den maximalen Aufwand für das Suchproblem. Er ergibt sich aus der Kombinatorik. Bei n Zugehörigkeitskoeffizienten und zwei Möglichkeiten $W = \{0, 1\}$ je Zugehörigkeitskoeffizient existieren genau 2^n unterschiedliche Vektoren \mathbf{w} .

Die Funktion $f(n) = 2^n$ gibt Aufschluss über die betrachtete Problemgröße. Sie stellt die obere Schranke für den Suchaufwand dar. Der Aufwand für das betrachtete Suchproblem ist exponentiell und wird mit $O(2^n)$ beschrieben. Algorithmen, deren Lösungsaufwand mit der Problemgröße exponentiell ansteigt, gehören zu der Klasse \mathcal{NP}^9 . Probleme der Klasse \mathcal{NP} sind nur bis zu einer bestimmten Größe mit vertretbarem Aufwand lösbar. Da die Schrankenfunktion $f(n) = 2^n$ den ungünstigsten Fall beschreibt, bleibt zu untersuchen, ob die Berücksichtigung von Nebenbedingungen und der Einsatz eines Branch-and-Bound-Verfahrens die exakte Bestimmung der gesuchten Lösungsmenge für realistische Problemgrößen erlaubt.

4.3.2 Branch-and-Bound

Branch-and-Bound ist eine Methode aus dem Bereich des Operation Research mit dem Ziel, für ein gegebenes Optimierungsproblem eine beste Lösung zu ermitteln. Es handelt sich nicht um ein festgelegtes Verfahren,

⁹ [Zimmermann, 2008, Komplexität von Algorithmen, Modellen und Problemen, S. 272 ff.]

sondern um eine Vorgehensweise, die der Problemstellung spezifisch angepasst wird. Das Prinzip besteht darin, ausgehend von einem beliebigen Element des Lösungsraums, das als aktuell beste Lösung deklariert wird (Initialisieren), den Lösungsraum schrittweise in Teilmengen zu zerlegen (Branching) und diejenigen Teilmengen aus der Betrachtung auszuschließen, die nachweislich keine bessere Lösung enthalten, als die bisher beste gefundene Lösung (Bounding). Durch rekursives Zerlegen entsteht sukzessiv eine Baumstruktur. Mit jeder Zerlegung wächst der Baum und die Teilmengen werden kleiner. Erreicht man eine Teilmenge, die nur noch ein Element enthält, wird es mit der bisher besten Lösung verglichen und gegebenenfalls ersetzt oder ergänzt. Die Suche ist abgeschlossen, wenn keine weitere Teilmenge mehr zu untersuchen ist. Im ungünstigsten Fall müssen alle Lösungen untersucht werden.

Einem Branch-and-Bound-Verfahren liegt ein Entscheidungsbaum zugrunde. Er dient der Strukturierung von Informationen, um eine systematische Suche zur Bestimmung der optimalen Lösungen zu unterstützen. Abhängig von der Art der Informationen, sind diese einem Knoten, einer Kante oder dem Baum als Ganzes zugeordnet.

Bei einem Suchproblem kann die optimale Lösung ein Element oder eine Teilmenge des Lösungsraums sein. Für den hier betrachteten Fall - der Suche nach Prozessmodellen - deren Abbildung a_ω minimal sein soll, stellt die optimale Lösung eine Menge von Prozessmodellen dar.

Die Beschreibung des Branch-and-Bound-Verfahrens erfordert Vorbetrachtungen. Der Abschnitt *Ansatz* geht auf den Grundgedanken ein, um den Aufwand der Suche zu reduzieren. Hierbei spielt das Konzept der Komponentenkombinationen eine entscheidende Rolle. Diese werden im Abschnitt *Komponentenkombinationen* eingeführt. Im Anschluss an die Vorbetrachtungen wird das *Prinzip des Entscheidungsbaums* und die *Struktur des Entscheidungsbaums* beschrieben. Abschließend wird auf die Schritte *Initialisieren*, *Verzweigen (Branching)* und *Terminieren (Bounding)* jeweils getrennt eingegangen.

Ansatz: Ausgangspunkt der Untersuchung ist der gesamte Lösungsraum Ω . Wie bereits in Abschnitt 4.3.1 *Optimierungsproblem* gezeigt, enthält Ω bei einer Anzahl von n Kanten 2^n Prozessmodelle. Nicht jedes Prozessmodell ω ist für die Suche von Bedeutung. Das Wissen darüber, dass Zerlegungen mit mehrfach auftretenden Mustern denjenigen überlegen sind, für die sich keine Wiederholungen ergeben, kann genutzt werden, um die Anzahl der zu betrachtenden Prozessmodelle deutlich herabzusetzen. Anstatt einzelne Kanten zu variieren, werden Komponentenkombinationen

variiert. Nur diejenigen Komponenten, die zu einer Einsparung beitragen können, werden betrachtet. Dies bedeutet, dass mit jedem Schritt nicht eine einzelne Kante, sondern eine Kantenmenge variiert wird. Diese Menge braucht nicht weiter verfeinert zu werden.

Komponentenkombinationen: Ein Ereignisgraph lässt sich in eine Vielzahl an Komponenten zerlegen. Ziel ist es, diejenigen Komponenten zu identifizieren, die durch Kombination zu einer Einsparung des Aufwands beitragen können. Dies ist für Komponenten der Fall, (1) die sich auf dasselbe Muster abbilden lassen, (2) die mindestens aus zwei Knoten und einer Kante bestehen und (3) die gleichzeitig und überschneidungsfrei als Untergraphen in G_E auftreten. Die Bedingungen (1) und (2) ergeben sich aus der Bewertung von Zerlegungen nach Gleichung 4.12. Die Bedingung (3) lässt sich aus dem Modellierungsansatz heraus erklären. Gesucht sind unabhängige Teilprozesse, die durch eine Menge an zusätzlichen Kanten den Graphen G_E ergeben. Es kommen daher nur Kombinationen von Komponenten in Betracht, die überschneidungsfrei sind.

Die Ermittlung der Komponenten, die die Bedingungen (1) und (2) erfüllen, wird in Abschnitt 4.3.4 *Entwickeln der Komponenten* beschrieben. Das Ergebnis ist eine Menge M_K an Komponenten, die sich entsprechend ihres Musters in Teilmengen M_K^p einteilen lassen. Der Indikator p bezieht sich auf die unterschiedlichen Muster.

M_K Menge der Komponenten, die die Bedingungen (1) und (2) erfüllen

M_K^p Menge an Komponenten, für die das Muster gleich ist, $M_K^p \subseteq M_K$

Die Teilmengen M_K^p repräsentieren Teilprozesse, die sich aus derselben Vorlage ableiten lassen. Nur wenn gleichzeitig aus einer Teilmenge M_K^p mindestens zwei Komponenten Teil eines Prozessmodells sind, ergibt sich eine Reduzierung des Aufwands. Jede Zerlegung, die nicht eine Einteilung der Kantenmenge in den Grenzen von wiederkehrenden Teilprozessen vornimmt, ist mit mehr Aufwand verbunden, als unter Berücksichtigung dieser Grenzen. Vor diesem Hintergrund lässt sich die Anzahl der zu untersuchenden Zerlegungen erheblich reduzieren. Bei dem Branch-and-Bound-Verfahren sollen nur solche Prozessmodelle untersucht werden, deren Zerlegung immer in den Grenzen von Komponenten liegt, die sich wiederholen. Die relevanten Komponenten sind die Elemente der Teilmengen M_K^p .

Nicht in jedem Fall sind die Knotenmengen der Komponenten, die zu derselben Teilmenge M_K^p gehören, paarweise disjunkt. Bestimmte Komponenten können somit nicht gleichzeitig Teil einer Zerlegung sein. Es ist daher erforderlich, diejenigen Komponentenkombinationen zu ermitteln, die überschneidungsfrei sind. In Abschnitt 4.3.5 *Überschneidungsfreie Komponentenkombinationen* wird gezeigt, wie aus den Teilmengen M_K^p überschneidungsfreie Komponentenkombinationen $C_{M_K^p,i}$ entwickelt werden. Alle überschneidungsfreien Komponentenkombinationen $C_{M_K^p,i}$ werden in der Menge C_K zusammengefasst.

- C_K Menge der überschneidungsfreien Komponentenkombinationen, die zu einer Einsparung führen
- $C_{M_K^p,i}$ eine überschneidungsfreie Komponentenkombination

Jedes Element der Menge C_K ist eine Komponentenkombinationen $C_{M_K^p,i}$. Eine Komponentenkombination $C_{M_K^p,i}$ ist eine Menge an überschneidungsfreien Komponenten gleichen Musters. Die Menge der Kanten, die sich aus einer Komponentenkombination ergibt, beschreibt diejenige Kantenmenge, die mit jedem Schritt im Branch-and-Bound-Verfahren variiert wird. Die Menge der Kanten wird implizit als Komponentenkombination $C_{M_K^p,i}$ erfasst. Diese Kantenmenge ist immer mit einer Einsparung verbunden.

Da sich die Komponenten und ihre Kombinationen zur Laufzeit der Suche nicht ändern, können sie vorab bestimmt werden. Die Kombination von Kantenmengen, denen Komponenten unterschiedlichen Musters zugrunde liegen, erfolgt im Branch-and-Bound-Verfahren. Jede Kante des Entscheidungsbaums ist mit einer Komponentenkombination verknüpft. Jeder Pfad repräsentiert eine Menge an Komponentenkombinationen. Für unterschiedliche Pfade sind diese Mengen unterschiedlich. Entlang eines Pfades ist das Muster, auf das sich die Komponenten einer Kombination abbilden lassen, paarweise verschieden. Durch dieses Vorgehen erfolgt jeder Schritt im Entscheidungsbaum (I.) in den Grenzen von Komponenten, (II.) in Verbindung mit einer Einsparung und (III.) ohne Wiederholungen.

Es folgt ein Beispiel, um das Konzept der Komponentenkombinationen zu veranschaulichen. Im Mittelpunkt steht die Fragestellung, was eine Komponentenkombination ist. Eine detaillierte Beschreibung der Einzelschritte erfolgt in den genannten Abschnitten. An dieser Stelle wird nur auf charakteristische Zwischenergebnisse eingegangen, die sich aus den

Bedingungen (1) bis (3) ergeben. Dem Beispiel liegt der Ereignisgraph aus Abbildung 4.3 zugrunde. Um die Zahl der Komponenten klein zu halten, wird nur der in Abbildung 4.4 dargestellte Ausschnitt betrachtet. Grundsätzlich muß immer der gesamte Graph berücksichtigt werden.

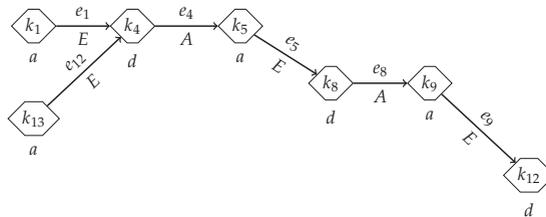


Abbildung 4.4: Ausschnitt eines Ereignisgraphen

Im ersten Schritt wird die Menge der Komponenten M_K bestimmt. Sie enthält die Komponenten, die zu einer Einsparung führen können. Diese sind für den betrachteten Ausschnitt in Abbildung 4.5 dargestellt. Die Komponenten sind nach Muster sortiert und zu Teilmengen M_K^p zusammengefasst. Es sind nur diejenigen Teilmengen dargestellt, die mehr als eine Komponente enthalten. Enthält eine Teilmenge M_K^p nicht mindestens zwei Elemente, lässt sich keine Einsparung erzielen.

Im nächsten Schritt werden die Komponenten der Teilmengen M_K^p auf Überschneidungen überprüft. Die Teilmenge M_K^1 aus Abbildung 4.5 enthält vier Komponenten. Die Komponenten $K_{E,1}$ und $K_{E,2}$ haben einen gemeinsamen Knoten k_4 (siehe Abbildung 4.4). Sie können nicht gleichzeitig Teil eines Prozessmodells sein. Um die Überschneidung aufzulösen, wird die Menge M_K^1 dupliziert. Aus M_K^1 wird $M_{K,1}^1$ und $M_{K,2}^1$ (siehe Abbildung 4.6). In den duplizierten Mengen wird jeweils eine der sich überschneidenden Komponenten entfernt. Das Duplizieren der Mengen wird rekursiv fortgesetzt, bis keine Überschneidungen mehr vorliegen. Der Bezeichner einer überschneidungsfreien Komponentenmenge $M_{K,i}^p$ entspricht dem Bezeichner der zugrunde liegenden Komponentenmenge M_K^p , erweitert um einen Index i . Nur diejenigen Mengen werden benannt, die sich unterscheiden und die keine Überschneidungen haben.

Abbildung 4.6 zeigt die aufgelösten Komponentenmengen $M_{K,i}^p$. Aus M_K^1 sind zwei Mengen $M_{K,1}^1$ und $M_{K,2}^1$ entstanden. Die Menge $M_{K,2}^1$ enthält

nur Komponenten, die sich nicht überschneiden. Ein Auflösen ist nicht erforderlich. Die Mengen M_K^2 und $M_{K,1}^2$ sind identisch. Die Komponenten in den Mengen M_K^3 und M_K^4 überschneiden sich paarweise. Das Auflö-

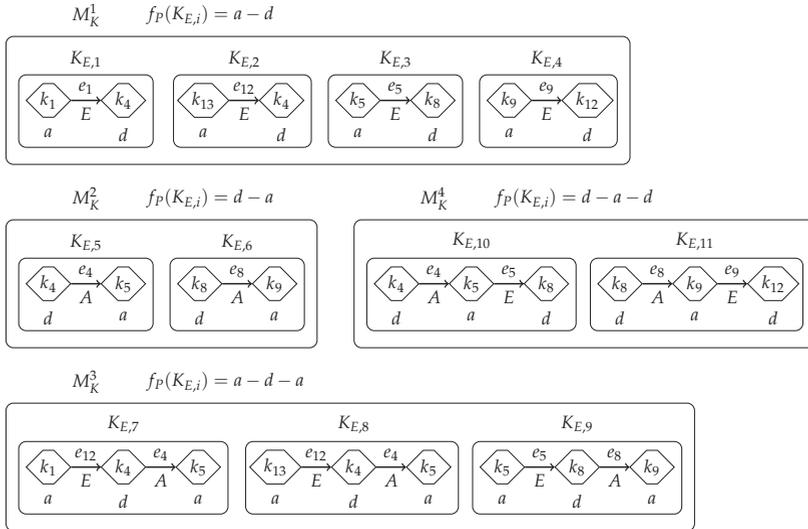


Abbildung 4.5: Komponenten, nach Muster sortiert

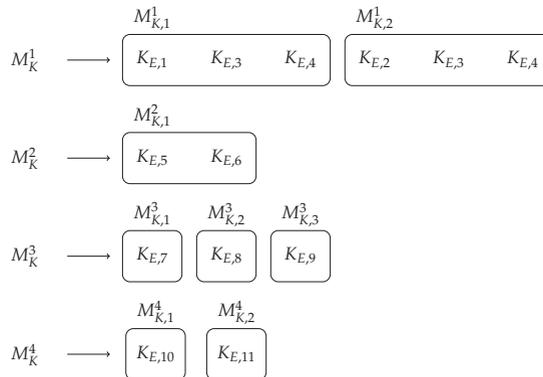


Abbildung 4.6: Auflösen von Überschneidungen

sen der Überschneidungen führt zu Mengen, in denen jeweils nur eine Komponente verbleibt. Sie können zu keiner Einsparung beitragen und werden verworfen. Für die weiteren Betrachtungen sind nur die Mengen $M_{K,1}^1$, $M_{K,2}^1$ und $M_{K,1}^2$ von Bedeutung.

Im letzten Schritt werden für jede Komponentenmenge $M_{K,i}^p$ die Kombinationen ihrer Elemente gebildet und in der Menge C_K zusammengefasst. Gesucht sind diejenigen Kombinationen, die mindestens zwei Komponenten enthalten. Eine Komponentenkombination wird mit $C_{M_{K,i}^p}$ bezeichnet. Sie ist immer eine Teilmenge von $M_{K,i}^p$. Der Bezeichner einer Komponentenkombination berücksichtigt den Bezeichner der zugrunde liegenden Komponentenmenge $M_{K,i}^p$. Da sich die Komponenten der Menge M_K^p unterschiedlich kombinieren lassen, erhält eine Komponentenkombination einen Zähler i . Der Zähler i einer Komponentenkombination $C_{M_{K,i}^p}$ hat keinen Zusammenhang mit dem Zähler i der Komponentenmenge $M_{K,i}^p$.

Die Mengen $M_{K,1}^1$ und $M_{K,2}^1$ enthalten drei Komponenten. Die Kombination ihrer Elemente liefert jeweils vier Komponentenkombinationen mit mehr als zwei Elementen. Da Elemente der beiden Mengen aber zum Teil gleich sind, ergeben sich in der Summe nur sieben unterschiedliche Komponentenkombinationen.

Die Teilmenge $M_{K,1}^2$ enthält zwei Komponenten. Entsprechend ist $C_{M_{K,1}^2} = M_{K,1}^2$. Die Mengen M_K^3 und M_K^4 liefern keinen Beitrag. Abbildung 4.7 zeigt die Menge der Komponentenkombinationen, die sich für den betrachteten Prozessausschnitt ergibt.

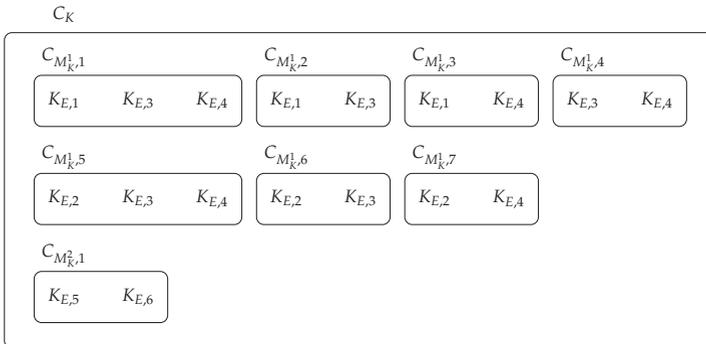


Abbildung 4.7: Komponentenkombinationen

Prinzip des Entscheidungsbaums: Das Prinzip des hier verwendeten Entscheidungsbaums besteht darin, zunächst alle Kanten des Graphen als variabel zu betrachten. Variabel bedeutet, dass für eine Kante noch keine Entscheidung darüber getroffen wurde, ob sie zu einem Teilprozess dazugehört, also ein Element von Z_{E-E} ist, oder ob sie zu der Menge der zusätzlich zu spezifizierenden Kanten S_{E-E} gehört. Mit jedem Schritt in den Baum hinein wird eine Teilmenge der Kanten fixiert, d.h. die Kanten der aktuellen Teilmenge werden einer der beiden Mengen zugeordnet. Mit einem Schritt zurück in Richtung der Wurzel, wird eine bereits fixierte Teilmenge wieder variabel.

Die Menge der Kanten, die mit einem Schritt verbunden ist, ergibt sich aus der Auswahl einer zulässigen Komponentenkombination. Zulässig ist eine Komponentenkombination, wenn die Menge ihrer Kanten eine Teilmenge der noch variablen Kanten ist. Existiert für einen Knoten keine weitere zulässige Komponentenkombination, dann werden alle verbleibenden variablen Kanten der Menge S_{E-E} übertragen. Damit ist die Kantenmenge vollständig fixiert. Erreicht man einen Knoten im Entscheidungsbaum, für den alle Kanten fixiert sind, dann beschreibt dieser ein Prozessmodell. Der Knoten kann nicht weiter verzweigt werden. Er stellt ein Blatt des Baums dar. Unterschiedliche Blätter beschreiben unterschiedliche Prozessmodelle. Gesucht sind diejenigen Prozessmodelle mit minimaler Bewertung.

Die Menge der Kanten lässt sich in drei disjunkte Kantenmengen einteilen. Dieses sind die Mengen $Z_{E-E,s}$, $S_{E-E,s}$ und $V_{E-E,s}$. Sie stehen in einem Zusammenhang. Mit jeder Bewegung im Entscheidungsbaum ergibt sich eine unterschiedliche Belegung. Abbildung 4.8 dient der Beschreibung dieser Mengen. Dargestellt ist eine Bewegung in den Baum hinein. Es handelt sich um einen Schritt von der Wurzel des Entscheidungsbaums zu einem Knoten in der Tiefe 1. Eine Bewegung erfordert die Auswahl einer Komponentenkombination. Stellvertretend wird die Komponentenkombination $C_{M_{k,1}^2}$ nach Abbildung 4.7 betrachtet. Vor der Auswahl waren alle Kanten variabel. Durch die Auswahl werden bestimmte Kanten fixiert. Dies betrifft zum einen die Kanten der Teilprozesse e_4 und e_8 und zum anderen die an die Teilprozesse angrenzenden Kanten e_1, e_3, e_5, e_7, e_9 und e_{12} . Die Kanten der Teilprozesse, die sich aus den Komponenten $K_{E,5}$ und $K_{E,6}$ der ausgewählten Komponentenkombination ergeben, werden in der Menge $Z_{E-E,1}$ zusammengefasst. Die an die Teilprozesse angrenzenden Kanten werden in der Menge $S_{E-E,1}$ zusammengefasst. Alle anderen Kanten gehören zu der Menge $V_{E-E,1}$. Die Kanten der Mengen $Z_{E-E,1}$ und $S_{E-E,1}$ werden als fixiert bezeichnet. Die Kanten der Menge $V_{E-E,1}$

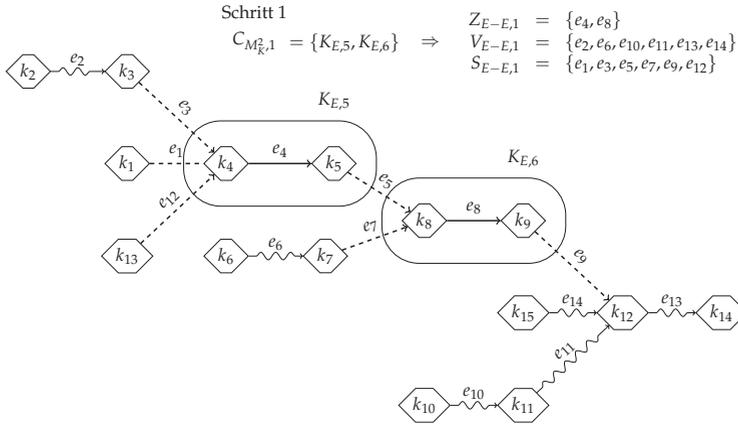


Abbildung 4.8: Selektion einer Komponentenkombination

gelten als variabel. Die Menge der fixierten Kanten steht in den nachfolgenden Schritten nicht mehr für die Auswahl geeigneter Komponenten kombinationen zur Verfügung. Jedem Schritt kann immer nur solch eine Komponenten kombination zugeordnet werden, so dass die Kanten der ausgewählten Komponenten kombination eine Teilmenge der verbleibenden variablen Kantenmenge ist. Das Fixieren der Kanten, die in $S_{E-E,s}$ enthalten sind, stellt sicher, dass sich die Teilprozesse, die zu unterschiedlichen Komponenten kombinationen gehören, nicht überschneiden.

Eine kompakte Beschreibung des Zusammenhangs zwischen den Mengen $Z_{E-E,s}$, $S_{E-E,s}$ und $V_{E-E,s}$ lässt sich auf der Grundlage des Vektors \mathbf{w} nach Gleichung 4.16 realisieren. Die Vereinigung der Mengen $Z_{E-E,s}$, $S_{E-E,s}$ und $V_{E-E,s}$ repräsentiert die Kantenmenge, die dem gegebenen Prozess zugrunde liegt. Für jede Kante dieser Menge enthält der Vektor \mathbf{w} einen Zugehörigkeitskoeffizienten. Koeffizienten, die sich auf Kanten der Menge $Z_{E-E,s}$ beziehen, werden mit einer 1 fixiert. Koeffizienten, die sich auf Kanten der Menge $S_{E-E,s}$ beziehen, werden mit einer 0 fixiert. Koeffizienten, die sich auf Kanten der Menge $V_{E-E,s}$ beziehen, werden nicht fixiert. Unterschiedliche Belegungen dieser Koeffizienten führen zu unterschiedlichen Prozessmodellen. Diejenige Teilmenge des Lösungsraums Ω , die sich auf der Grundlage der variablen Koeffizienten abbilden lässt, sei durch $\mathbf{w}_{\Omega_{x,y}}$ beschrieben.

Abbildung 4.9 zeigt einen Vektor $\mathbf{w}_{\Omega_{x,y}}$ und seine qualitative Entwick-

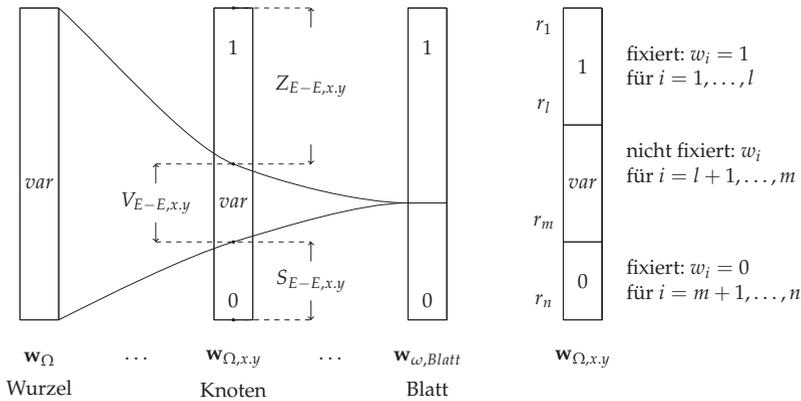
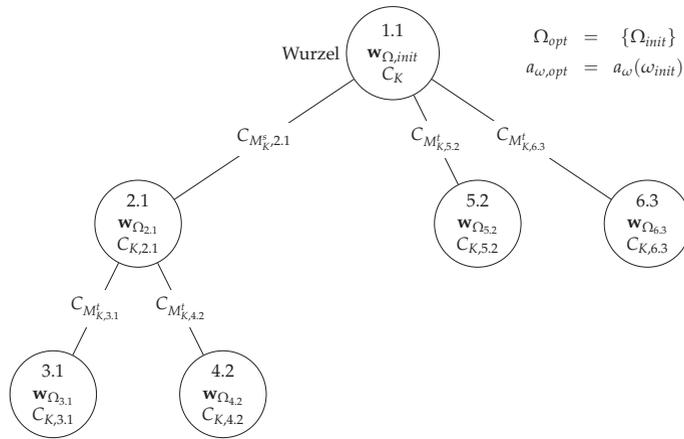


Abbildung 4.9: Prinzip des Entscheidungsbaums

lung auf einem Pfad von der Wurzel bis zu einem Blatt in dem Entscheidungsbaum. Sind alle Koeffizienten variabel, beschreibt der Vektor \mathbf{w} den gesamten Lösungsraum Ω und wird mit \mathbf{w}_Ω bezeichnet. Sind alle Koeffizienten fixiert, stellt der Vektor \mathbf{w} ein konkretes Prozessmodell ω dar. In diesem Fall entspricht der Vektor $\mathbf{w}_{\Omega,x,y}$ dem Vektor \mathbf{w} nach Gleichung 4.16 und wird mit $\mathbf{w}_{\omega,x,y}$ bezeichnet. Bezogen auf den Entscheidungsbaum stellt der Knoten ein Blatt dar.

Struktur des Entscheidungsbaums: Ein Entscheidungsbaum dient der Strukturierung von Informationen, um eine systematische Suche zur Bestimmung der optimalen Lösungen zu unterstützen. Abhängig von der Art der Informationen sind diese einem Knoten, einer Kante oder dem Baum als Ganzes zugeordnet. Abbildung 4.10 zeigt die Struktur des Entscheidungsbaums, der dem hier verwendeten Branch-and-Bound-Verfahren zugrunde liegt.

Jeder Knoten hat einen Identifikator. Der Ursprungsknoten wird mit *Wurzel* bezeichnet. Der Identifikator setzt sich aus zwei Zählern x,y zusammen. Der erste Zähler x ist ein Laufindex. Mit jedem Knoten, der neu entwickelt wird, zählt x hoch. Sein Wert beschreibt die Reihenfolge, in der der Baum durchlaufen wird. Der zweite Zähler y beschreibt die Position in der Reihenfolge von Kindknoten, die denselben Vater haben. Zur Identifikation reicht bereits der erste Zähler x . Der zweite Zähler dient der An-



$$\Omega_{opt} = \{\Omega_{init}\}$$

$$a_{\omega, opt} = a_{\omega}(\omega_{init})$$

Legende

Knoten

$x.y$	Identifikator
	x Reihenfolge, y wieviertes Kind
$\mathbf{w}_{\Omega_{x,y}}$	Teilmenge des Lösungsraums
$C_{K,x,y}$	Menge an verbleibenden Komponentenkombinationen

Kanten

$C_{M_{k,x,y}^p}$	Komponentenkombination
Baum	
Ω_{opt}	Optimale Lösungen
$a_{\omega, opt}(\dots)$	obere Schranke

Abbildung 4.10: Entscheidungsbaum

schauung und ist nur für die Beschreibung des Verfahrens von Bedeutung.

Weitere Elemente eines Knotens sind der Vektor $\mathbf{w}_{\Omega_{x,y}}$ und eine Menge an Komponentenkombinationen $C_{K,x,y}$. Der Vektor $\mathbf{w}_{\Omega_{x,y}}$ wurde bereits im letzten Abschnitt erläutert. Er dient als Grundlage, um Kantenmengen zu ermitteln. Die Menge $C_{K,x,y}$ enthält Komponentenkombinationen. Der Wurzelknoten enthält alle Komponentenkombinationen. Mit jedem Schritt in den Baum hinein ist diese Menge anzupassen. Für einen Knoten $x.y$ verbleiben diejenigen Komponentenkombinationen von C_K in $C_{K,x,y}$, deren Kanten vollständig in $V_{E-E,x,y}$ enthalten sind, die also noch nicht fixiert sind. Jede Komponentenkombination in $C_{K,x,y}$ ist ein Kandidat, aus dem für den Knoten $x.y$ ein Kind entstehen kann.

Eine Kante des Entscheidungsbaums verknüpft einen Vaterknoten mit einem seiner Kindknoten. Ein Vaterknoten kann maximal so viele Kinder haben, wie es Komponentenkombinationen in $C_{K,Vater}$ gibt. Jede Kante ist mit einer Komponentenkombination $C_{M_{k,x,y}^p}$ verknüpft. Da sie den Kind-

knoten definiert, wird eine Kante mit dem Bezeichner des Kindknotens verknüpft. Komponentenkombinationen können bewertet werden. Aus der normierten Bewertung nach Abschnitt 4.2.9 lässt sich eine Reihenfolge bestimmen. Diese Reihenfolge wird über den Zähler y ausgedrückt.

Entlang eines Pfads grenzen die Komponentenkombinationen den verbleibenden Lösungsraum immer weiter ein, bis ein Blatt erreicht ist, das ein Prozessmodell repräsentiert. Dieses kann bewertet werden. Ist seine Bewertung besser oder gleich der bisher besten gefundenen Lösung, ersetzt bzw. ergänzt das Prozessmodell die Elemente in der Menge Ω_{opt} . Der Wert $a_{\omega,opt}$ ist entsprechend anzupassen. Ω_{opt} und die Schranke $a_{\omega,opt}$ sind dem Baum als Ganzes zugeordnet.

Initialisieren: Das Initialisieren des Entscheidungsbaums umfasst das Anlegen der Wurzel und die Angabe einer ersten Schranke für das Minimierungsproblem. Diejenigen Schritte, die für eine Transformation der Problemstellung auf die Elemente des Baum erforderlich sind, gehören ebenfalls zur Initialisierung dazu. Für das hier beschriebene Verfahren betrifft dies die Ermittlung minimaler Komponenten und die Ermittlung der bereits eingeführten Komponentenkombinationen.

Minimale Komponenten dienen der Bestimmung eines initialen Prozessmodells ω_{init} . Mit der Identifikation von ω_{init} lässt sich bereits für die Wurzel eine obere Schranke angeben, die besser ist als $a_{\omega,opt} = +\infty$. A priori lässt sich die Kantenmenge eines gegebenen Prozesses einteilen in die drei Kantenmengen $Z_{E-E,init}$, $V_{E-E,init}$ und $S_{E-E,init}$. Diejenigen Kanten, die immer Teil einer Komponente sein werden, seien in der Menge $Z_{E-E,init}$ zusammengefasst. Der Untergraph von G_E , der nur aus dieser Kantenmenge und den entsprechenden Knoten besteht, lässt sich nach Gleichung 4.3 als Zerlegung $Z_{E,init} = (M_E ; Z_{E-E,init})$ beschreiben. Die Komponenten der Zerlegung $Z_{E,init}$ werden als minimale Komponenten bezeichnet. Ihre Ermittlung wird im Abschnitt 4.3.3 beschrieben. Ist $Z_{E-E,init}$ bekannt, ergibt sich eine korrespondierende Menge $S_{E-E,init}$ (siehe Abbildung 4.8). Diese Menge umfasst diejenigen Kanten, die aus dem Gesamtprozess herauszuschneiden wären, um die minimalen Komponenten zu isolieren. Darüber hinaus werden weitere Kanten in der Menge $S_{E-E,init}$ eingetragen. Kanten, deren Muster sich nicht wiederholt, können nicht zu einer Einsparung des Aufwands beitragen. Sie werden ebenfalls a priori als spezifizierende Kanten in der Menge $S_{E-E,init}$ registriert. Die verbleibenden Kanten, die nicht ein Element von $Z_{E-E,init}$ oder $S_{E-E,init}$ sind, gehören zu der Menge $V_{E-E,init}$. Die Mengen $Z_{E-E,init}$, $V_{E-E,init}$ und $S_{E-E,init}$ lassen sich als Vektor $\mathbf{w}_{\Omega,init}$ darstellen. Setzt man in $\mathbf{w}_{\Omega,init}$ alle

variablen Kanten auf 0, dann ergibt sich ein Vektor $\mathbf{w}_{\omega_{init}}$, der das Prozessmodell ω_{init} beschreibt. Das Prozessmodell ω_{init} wird als vorläufig beste Lösung zu der Menge Ω_{opt} hinzugefügt. Sein Aufwand wird als vorläufige obere Schranke $a_{\omega_{opt}} = a_{\omega}(\omega_{init})$ vermerkt. Das Prozessmodell ω_{init} enthält nur minimale Komponenten.

Die Berücksichtigung minimaler Komponenten zum Zeitpunkt der Initialisierung des Entscheidungsbaums führt an zwei Stellen zu einer Einsparung des Lösungsaufwands. Zum einen erfolgt das *Entwickeln der Komponenten* nach Abschnitt 4.3.4 auf der Grundlage der minimalen Komponenten und zum anderen ergibt sich eine Reduzierung des Lösungsraums Ω . Wie bereits in Abschnitt 4.3.1 *Optimierungsproblem* gezeigt, enthält Ω bei einer Anzahl von n Kanten 2^n Prozessmodelle. Vor dem Hintergrund, dass minimale Komponenten nicht weiter zerlegt werden müssen, ergibt sich eine Reduzierung des Lösungsraums von 2^n auf 2^q .

$$\begin{aligned} l &= |Z_{E-E,init}| \\ n - m &= |S_{E-E,init}| = p \\ n - l + p &= |V_{E-E,init}| = q \end{aligned} \quad (4.17)$$

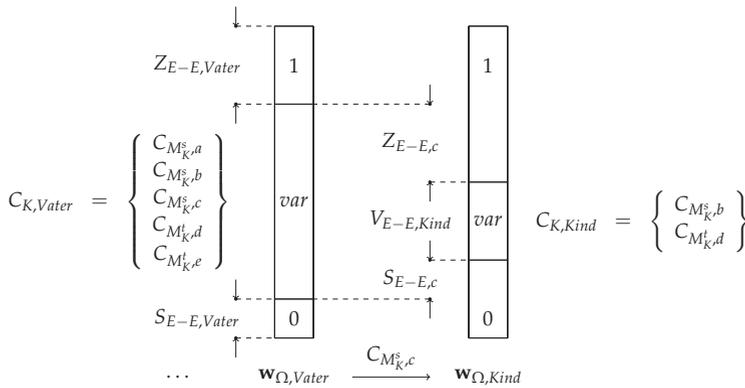
l, m und n nach Abbildung 4.9

- l Anzahl der Koeffizienten w_i fixiert auf 1
- p Anzahl der Koeffizienten w_i fixiert auf 0
- q Anzahl der variablen Koeffizienten w_i

Die Initialisierung ist abgeschlossen, wenn der Wurzelknoten angelegt ist.

Verzweigen (Branching): Das Verzweigen beschreibt sowohl das Anlegen von Kindknoten als auch die Auswahl eines Kindknotens. Das Anlegen von Kindknoten wird durch Verzweigungsregeln gesteuert. Diese Regeln legen fest, wie ein Knoten in Kindknoten zu zerlegen ist. Die Auswahl von Kindknoten wird durch Auswahlregeln beeinflusst. Diese Regeln legen fest, in welcher Reihenfolge die Kindknoten betrachtet werden.

Das Anlegen und die Auswahl eines Kindknotens sind eng miteinander verknüpft. Dies ist auf folgenden Umstand zurückzuführen: Anstatt den Baum sukzessiv zu entwickeln, d.h. die einmal traversierten Knoten als bekannt und abrufbar zu betrachten, wird hier wegen des begrenzt zur Verfügung stehenden Speicherplatzes der Ansatz verfolgt, einen terminierten Knoten zu verwerfen. Zur Laufzeit der Suche sind immer nur diejenigen Baumknoten bekannt, die auf dem Pfad von dem Wurzelknoten zu dem aktuellen Knoten liegen. Als aktueller Knoten wird der Knoten bezeichnet, der gerade untersucht wird.

Abbildung 4.11: Verzweigen von $\mathbf{w}_{\Omega,Vater}$

Bewegt man sich in den Baum hinein, ist ein Verzweigen des aktuellen Knotens erforderlich. Das Verzweigen eines Knotens wird an dem Beispiel aus Abbildung 4.11 erläutert. Gegeben seien die Knoten entlang eines Pfades bis zu dem Knoten *Vater*. Der Pfad ist nicht dargestellt. Der Knoten *Vater* sei durch den Vektor $\mathbf{w}_{\Omega,Vater}$ und die Komponentenkombinationen $C_{K,Vater}$ beschrieben. Die Menge $Z_{E-E,Vater}$ enthält alle Kanten, die zu den Komponenten gehören, die entlang des Pfades fixiert wurden. Die Menge $S_{E-E,Vater}$ enthält diejenigen Kanten, die die fixierten Komponenten isolieren, und diejenigen, die nur einfach auftreten. Die Menge $C_{K,Vater}$ umfasst diejenigen Komponentenkombinationen, deren Kantenmengen Teilmengen der variablen Kanten von *Vater* sind. Jedes Element von $C_{K,Vater}$ ist ein Kandidat für einen Kindknoten.

Die Auswahlregel: Komponentenkombinationen tragen in unterschiedlichem Maße zu einer Einsparung des Spezifikationsaufwands bei. Zur Erläuterung: Eine Komponentenkombination enthält eine Menge an Komponenten, die sich auf dasselbe Muster abbilden lassen. Ihnen liegt dieselbe Vorlage zugrunde. Der Spezifikationsaufwand ergibt sich aus der Spezifikation der Vorlage (Knoten und Kanten) und der n -fachen Anwendung der Vorlage bei n Komponenten. Grundsätzlich haben diejenigen Komponentenkombinationen mit einer großen Anzahl an Komponenten ein größeres Einsparpotential als solche, mit einer kleineren Anzahl an Komponenten. Dies folgt aus Gleichung 4.13. Wie sehr eine Komponentenkombination durch Wiederholungen profitiert, geht aus der *normier-*

ten Bewertung einer Zerlegung nach Abschnitt 4.2.9 hervor. Um möglichst schnell auf eine gute Lösung zu treffen, berücksichtigt die Auswahlregel das Einsparpotential, das sich umgekehrt proportional zur normierten Bewertung verhält. Je kleiner der Wert der normierten Bewertung, desto größer ist das Einsparpotential. Es wird immer nur für diejenige Komponentenkombination ein Kind angelegt, für die das Einsparpotential maximal ist bzw. die normierte Bewertung minimal ist.

$$n_{Z,min} = \min\{n_Z(C_{M_{K,i}^p}) \mid C_{M_{K,i}^p} \in C_{K,x,y}\} \quad (4.18)$$

- $n_{Z,min}$ minimale normierte Bewertung
 $C_{K,x,y}$ Menge der Komponentenkombinationen eines Knotens x,y
 $C_{M_{K,i}^p}$ eine Komponentenkombinationen aus $C_{K,x,y}$

Die Verzweigungsregel: Sei die Komponentenkombination $C_{M_{K,c}^s}$ das Element von $C_{K,Vater}$ mit der minimalen normierten Bewertung $n_{Z,min}$. Für die Komponentenkombination $C_{M_{K,c}^s}$ wird ein Kindknoten angelegt. Die Auswahl von $C_{M_{K,c}^s}$ führt dazu, dass das Element $C_{M_{K,c}^s}$ aus $C_{K,Vater}$ entfernt wird. Auf diese Weise wird sichergestellt, dass bei einer erneuten Auswahl eines Kindes für den Knoten *Vater* das nächste Element mit dem dann größten Einsparpotential aus $C_{K,Vater}$ herangezogen wird. Für einen Knoten kann solange eine Auswahl erfolgen, bis keine Elemente mehr in $C_{K,Vater}$ verbleiben. Da mit jeder erneuten Auswahl ein anderer Knoten behandelt wird, beschreibt dieses Vorgehen das Verzweigen.

Das Anlegen eines Knotens: Mit jedem neuen Knoten wird ein neuer Vektor $\mathbf{w}_{\Omega,Kind}$ angelegt. Dieser wird mit den Werten seines Vaters vorkonfiguriert. Durch die Auswahl einer Komponentenkombination, hier stellvertretend für das Element $C_{M_{K,c}^s}$ aus $C_{K,Vater}$, werden weitere Kanten fixiert. Die Menge $Z_{E-E,c}$ sei die Menge der Kanten, die zu $C_{M_{K,c}^s}$ gehört. Diese werden in $\mathbf{w}_{\Omega,Kind}$ mit 1 fixiert. Diejenigen Kanten, die zusätzlich erforderlich sind, um die Komponenten von $C_{M_{K,c}^s}$ zu isolieren, sind in $S_{E-E,c}$ enthalten. Diese werden in $\mathbf{w}_{\Omega,Kind}$ mit 0 fixiert. Das Fixieren der Kanten in $\mathbf{w}_{\Omega,Kind}$ reduziert die Menge der variablen Kanten. Für den Kindknoten sind diejenigen Komponentenkombinationen aus $C_{K,Vater}$ zu ermitteln, (1) für die die Kantenmenge eine Teilmenge der variablen Kanten $V_{E-E,Kind}$ des Kindes ist und (2) deren Muster sich von dem Muster der Komponentenkombination $C_{M_{K,c}^s}$ unterscheidet. Diese Komponentenkombinationen werden in die Menge $C_{K,Kind}$ übertragen. Jedes Element von $C_{K,Kind}$ stellt einen Kandidaten dar, aus dem in späteren Schritten weitere Kinder entstehen können.

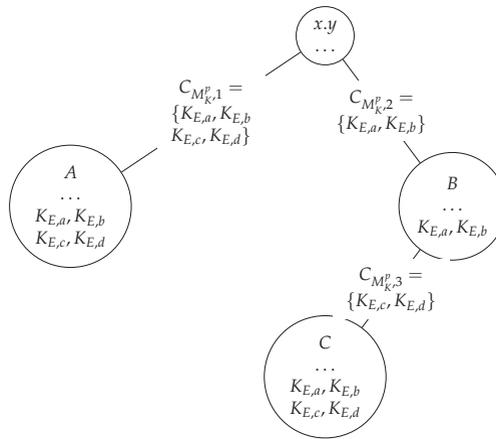


Abbildung 4.12: Wiederkehrende Kombinationen auf unterschiedlichen Pfaden

Die Bedingung (2) stellt sicher, dass sich die Menge der fixierten Komponenten auf unterschiedlichen Pfaden nicht wiederholt. Abbildung 4.12 zeigt ein Beispiel für den Fall, dass die Bedingung nicht berücksichtigt wird. Sei $M_K^p = \{K_{E,a}, K_{E,b}, K_{E,c}, K_{E,d}\}$ eine Menge an überschneidungsfreien Komponenten. Aus dieser Menge ergeben sich unter anderem die Komponentenkombinationen $C_{M_K^p,1} = \{K_{E,a}, K_{E,b}, K_{E,c}, K_{E,d}\}$, $C_{M_K^p,2} = \{K_{E,a}, K_{E,b}\}$ und $C_{M_K^p,3} = \{K_{E,c}, K_{E,d}\}$. Die Komponentenkombination $C_{M_K^p,1}$ führt zu dem Kind A. Die Komponentenkombination $C_{M_K^p,2}$ führt zu dem Kind B und über $C_{M_K^p,3}$ weiter nach C. Innerhalb der Knoten sind die fixierten Komponenten aus den jeweiligen Schritten dargestellt. Es ergibt sich, dass die Knoten A und C identisch sind. Durch die Bedingung (2) wird verhindert, dass Komponentenkombinationen, deren Komponenten sich auf dasselbe Muster abbilden lassen, in aufeinanderfolgenden Schritten auf unterschiedlichen Pfaden erneut kombiniert werden.

Terminieren (Bounding): Erreicht das Branch-and-Bound-Verfahren einen Knoten im Entscheidungsbaum, der nicht weiter verzweigt wird, dann markiert der aktuelle Knoten das Ende eines Pfades, an dem die Suche terminiert wird. Für das hier beschriebene Verfahren können zwei Ursa-

chen zum Terminieren der Suche an einem Knoten führen.

1. Alle Kanten des Vektors $\mathbf{w}_{\Omega,x,y}$ sind fixiert, der Knoten kann terminiert werden.
2. Nicht alle Kanten des Vektors $\mathbf{w}_{\Omega,x,y}$ sind fixiert, dennoch kann ausgeschlossen werden, dass innerhalb der verbleibenden Teilmenge $\Omega_{x,y}$ des Lösungsraums Ω eine Lösung ω existiert, für die die Bewertung $a_\omega(\omega) \leq a_{\omega,opt}$ ist.

Fall 1: Das Ende eines Pfads ist erreicht. Es existieren keine weiteren Elemente mehr in $C_{K,x,y}$, um den aktuellen Knoten weiter zu verzweigen. Der Vektor $\mathbf{w}_{\Omega,x,y}$ beschreibt ein Prozessmodell $\omega_{x,y}$. Seine Bewertung $a_\omega(\omega_{x,y})$ wird bestimmt und mit der Schranke $a_{\omega,opt}$ verglichen. Ist die Bewertung besser oder gleich der bisher besten gefundenen Lösung, ersetzt bzw. ergänzt das Prozessmodell $\omega_{x,y}$ die Elemente in der Menge Ω_{opt} . Der Wert $a_{\omega,opt}$ ist entsprechend anzupassen.

Fall 2: Der Vektor $\mathbf{w}_{\Omega,x,y}$ beschreibt eine Teilmenge des Lösungsraums $\Omega_{x,y}$. Ein Vergleich von Aufwandswerten erlaubt es, den verbleibenden Lösungsraum $\Omega_{x,y}$ abzuschätzen. Ist der verbleibende, verfügbare Aufwand kleiner als der verbleibende erforderliche Aufwand, dann kann die Suche auf dem aktuellen Pfad terminiert werden. Im Folgenden werden der verbleibende verfügbare Aufwand und der verbleibende erforderliche Aufwand eingeführt.

Verbleibender verfügbarer Aufwand: Zugrunde liegt die Teilmenge $\Omega_{x,y}$ des Lösungsraums Ω . Betrachtet man für einen Knoten x,y den Untergraphen Z_E von G_E , der nur die fixierten Komponenten enthält, so lässt sich dieser Graph nach Gleichung 4.12 bewerten. Die Bewertung berücksichtigt wiederkehrende Teilprozesse. Ergänzt man diese Bewertung um die Anzahl der Kanten in $S_{E-E,x,y}$, dann beschreibt die Summe den Aufwand, der sich auf den fixierten Anteil von x,y bezieht. Die Bewertung sei mit $a_{\mathbf{w}_\Omega}$ bezeichnet. Sie ist ein Maß für den ausgeschöpften Aufwand, um den aktuellen Stand zu erreichen.

$$\begin{aligned}
 a_{\mathbf{w}_\Omega}(\mathbf{w}_{\Omega,x,y}) &= a_Z(Z_E(\mathbf{w}_{\Omega,x,y})) + |S_{E-E}(\mathbf{w}_{\Omega,x,y})| \\
 &= a_{Z,Z_{E,x,y}} + |S_{E-E,x,y}| \\
 &= a_{\mathbf{w}_{\Omega,x,y}}
 \end{aligned} \tag{4.19}$$

$a_{\mathbf{w}_{\Omega,x,y}}$	ausgeschöpfter Aufwand am Knoten $x.y$ des Entscheidungsbaums
$Z_{E,x,y}$	Untergraph von G_E , der nur die fixierten Komponenten am Knoten $x.y$ enthält
$S_{E-E,x,y}$	isolierende Kanten am Knoten $x.y$
$Z_{E-E,x,y}$	fixierte Kanten am Knoten $x.y$
$a_{Z,Z_{E,x,y}}$	ausgeschöpfter Aufwand für die fixierten Komponenten am Knoten $x.y$

Der Begriff *ausgeschöpft* bezieht sich auf die obere Schranke $a_{\omega,opt}$. Lösungen, die diese Schranke nicht einhalten, brauchen nicht betrachtet zu werden. Mit jedem Schritt in den Baum hinein werden immer mehr Komponenten fixiert. Der ausgeschöpfte Aufwand steigt. Die Differenz aus $a_{\omega,opt}$ und $a_{\mathbf{w}_{\Omega,x,y}}$ liefert den verbleibenden nicht ausgeschöpften Aufwand. Der verbleibende nicht ausgeschöpfte Aufwand ist gleichzeitig der verbleibende verfügbare Aufwand. Er wird mit $a_{\mathbf{w}_{\Omega}}^{verf}$ bezeichnet.

$$a_{\mathbf{w}_{\Omega,x,y}}^{verf} = a_{\omega,opt} - a_{\mathbf{w}_{\Omega,x,y}} \quad (4.20)$$

$a_{\mathbf{w}_{\Omega,x,y}}^{verf}$ verbleibender nicht ausgeschöpfter Aufwand
bzw. verbleibender verfügbarer Aufwand am
Knoten $x.y$

Verbleibender erforderlicher Aufwand: Zugrunde liegt die Teilmenge $\Omega_{x,y}$ des Lösungsraums Ω . Betrachtet wird der Untergraphen V_E von G_E , der nur die variablen Kanten $V_{E-E,x,y}$ mit den entsprechenden Knoten enthält. Der Untergraphen V_E , beschreibt den Teil von G_E , der noch zu fixieren ist. Hierfür stehen am Knoten $x.y$ die in $C_{K,x,y}$ enthaltenen Komponentenkombinationen zur Verfügung. Das Minimum für den verbleibenden erforderlichen Aufwand stellt sich für den Fall ein, dass es eine Komponentenkombination in $C_{K,x,y}$ gibt, die den Untergraphen V_E vollständig ausfüllt und dessen normierte Bewertung n_Z minimal ist. In jedem anderen Fall ist der verbleibenden erforderlichen Aufwand größer, da der Mittelwert der normierten Bewertungen, der sich aus der Kombination von beliebigen, verbleibenden Komponentenkombinationen ergibt, nur ansteigen kann.

Der erforderlicher Aufwand $a_{\mathbf{w}_{\Omega}}^{erf}$ berechnet sich aus dem Produkt aus $a(V_E)$ - Anzahl der Knoten und Kanten von V_E nach Gleichung 4.11 - und der minimalen normierten Bewertung n_Z nach Gleichung 4.18. Er ist ein

Maß dafür, wieviel Aufwand mindestens verzehrt wird, um eine der verbleibenden Komponentenkombinationen aus $C_{K,x,y}$ in V_E einzusetzen.

$$\begin{aligned} a_{\mathbf{w}_\Omega}^{erf}(\mathbf{w}_{\Omega,x,y}) &= a(V_E(\mathbf{w}_{\Omega,x,y})) \cdot n_{Z,min}(\mathbf{w}_{\Omega,x,y}) \\ &= a_{V_{E,x,y}} \cdot n_{Z,min,x,y} \\ &= a_{\mathbf{w}_{\Omega,x,y}}^{erf} \end{aligned} \quad (4.21)$$

$a_{\mathbf{w}_{\Omega,x,y}}^{erf}$ verbleibender erforderlicher Aufwand am Knoten x,y

$V_{E,x,y}$ Untergraph von G_E , der nur die variablen Kanten am Knoten x,y enthält

$V_{E-E,x,y}$ Menge der variablen Kanten am Knoten x,y

$a_{V_{E,x,y}}$ Anzahl der variablen Knoten und Kanten am Knoten x,y

$n_{Z,min,x,y}$ Komponentenkombination aus C_K am Knoten x,y , für die die normierte Bewertung minimal ist

Ist der, aus der Abschätzung resultierende, erforderliche Aufwand $a_{\mathbf{w}_\Omega}^{erf}$ größer als der verbleibende verfügbare Aufwand $a_{\mathbf{w}_{\Omega,x,y}}^{verf}$, so kann ausgeschlossen werden, dass andere Komponentenkombinationen aus $C_{K,x,y}$ diese Grenze unterschreiten.

Nur wenn es eine Möglichkeit gibt, die verbleibenden Kanten zu fixieren, ohne dass der dafür erforderliche Aufwand den noch verfügbaren Aufwand bis zum Erreichen der Schranke übersteigt, muss weiter verzweigt werden. Die Bedingung für ein Terminieren der Suche am aktuellen Knoten x,y lässt sich wie folgt formal ausdrücken:

$$a_{\mathbf{w}_{\Omega,x,y}}^{verf} \leq a_{\mathbf{w}_{\Omega,x,y}}^{erf} \quad \text{Wenn der Ausdruck } wahr \text{ ist, dann ist der aktuelle Knoten zu terminieren.} \quad (4.22)$$

$a_{\mathbf{w}_{\Omega,x,y}}^{erf}$ verbleibender erforderlicher Aufwand aus $V_{E-E,x,y}$ und $C_{K,x,y}$

$a_{\mathbf{w}_{\Omega,x,y}}^{verf}$ verbleibender verfügbarer Aufwand aus $Z_{E-E,x,y}$

4.3.3 Minimale Komponenten

Eine minimale Komponente ist eine Komponente, die vor dem Hintergrund der Suche nach optimalen Zerlegungen nicht weiter zerlegt werden muss. Dies wird an dem Beispiel aus Abbildung 4.13 einleitend erläutert. Jeder Knoten ist mit seiner Beschriftung dargestellt. Betrachtet man die

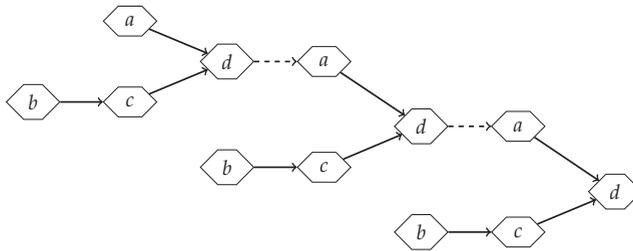


Abbildung 4.13: Einführung minimaler Komponenten

Knoten mit der Beschriftung b , so stellt man fest, dass diese immer mit einem Nachfolger c in Beziehung stehen. Die Knoten c haben immer b als Vorgänger und d als Nachfolger. Jeder Knoten a hat immer einen Nachfolger d aber nicht in jedem Fall einen Vorgänger d . Aufgrund dieser Unregelmäßigkeit werden alle Kanten (d, a) gelöscht. Unterschiede bei dem Vergleich von Kanten zu Vorgängern und Nachfolgern, die an Knoten mit gleicher Beschriftung anliegen, werden als Unregelmäßigkeit aufgefasst. Ist die Anzahl der vergleichbaren Kanten an den Knoten mit Beschriftung x unterschiedlich, dann werden alle Kanten mit gleicher Beschriftung an allen Knoten mit Beschriftung x entfernt. Die Suche nach Unregelmäßigkeiten wird solange fortgesetzt, bis es keine Unregelmäßigkeiten mehr gibt. Diejenigen Komponenten, die sich durch das Entfernen der unregelmäßigen Kanten ergeben, stellen die Menge der minimalen Komponenten dar.

Bei der Suche nach Unregelmäßigkeiten wird für jeden Knoten eine Kontextkomponente entwickelt. Diese enthält den Knoten, seine unmittelbaren Nachbarn und die verbindenden Kanten. Die Kontextkomponenten sind nach dem Bezeichner ihres zentralen Knotens in Teilmengen zusammenzufassen. Die Auswertung erfolgt für jede Teilmenge getrennt. Die Kontextkomponenten einer Teilmenge werden paarweise verglichen. Treten Kanten mit unterschiedlicher Häufigkeit auf, dann werden alle Kanten dieses Typs entfernt.

Die Ermittlung der minimalen Komponenten erfolgt mit dem Initialisieren des Entscheidungsbaums. Sie verändern sich zur Laufzeit der Suche nicht und können daher a priori bestimmt werden. Die nachfolgende Beschreibung zur Bestimmung der minimalen Komponenten orientiert sich am Beispiel des Ereignisgraphen aus Abbildung 4.3.

Kontextkomponenten: Der Kontext eines Knotens stellt eine Komponente dar. Die Komponente umfasst den Knoten $x \in M_E$ selbst, die zu dem Knoten adjazenten Knoten und die zu dem Knoten inzidenten Kanten. Der hochgestellte Wert in Klammern gibt an, bis zu welcher Entfernung von x die Nachbarschaft bzw. der Kontext betrachtet wird. Für eine Kontextkomponente ist nur die direkte Nachbarschaft von Bedeutung, daher wird die Entfernung 1 betrachtet. Die Vorgänger eines Knotens x sind in der Menge $t_{V,x}^{(1)}$ erfasst und die Nachfolger in der Menge $t_{N,x}^{(1)}$.

$$\begin{aligned}
 K_{E,x}^{(1)} &= (M_{E,x}^{(1)} ; R_{E-E,x}^{(1)}) \\
 M_{E,x}^{(1)} &= \{x\} \cup t_{V,x}^{(1)} \cup t_{N,x}^{(1)} \\
 R_{E-E,x}^{(1)} &= \{ (v,x) \in R_{E-E} \mid v \in t_{V,x}^{(1)} \ \wedge \\
 &\quad (x,n) \in R_{E-E} \mid n \in t_{N,x}^{(1)} \}
 \end{aligned} \tag{4.23}$$

$K_{E,x}^{(1)}$	Kontextkomponente um x
$M_{E,x}^{(1)}$	Menge der Knoten adjazent zu x incl. x
$R_{E-E,x}^{(1)}$	Menge der Kanten, inzident mit x
$t_{V,x}^{(1)}$	direkte Vorgänger von x
$t_{N,x}^{(1)}$	direkte Nachfolger von x

Kontextkomponenten, sortiert nach Beschriftung: Die Menge der Knoten M_E eines Ereignisgraphen G_E lässt sich auf der Grundlage ihrer Beschriftungen in Äquivalenzklassen einteilen. Jede Äquivalenzklasse stellt diejenige Teilmenge von M_E dar, für die die Knotenbeschriftung gleich ist.

$$[e] := \{x \in M_E \mid (e,x) \in \ddot{A}_E\} \tag{4.24}$$

	Beispiel:
$[e]$ Äquivalenzklasse	$[a] = \{k_1, k_5, k_9, k_{13}, k_{14}, k_{15}\}$
e ein Ereignis (Repräsentant)	$[b] = \{k_2, k_6, k_{10}\}$
M_E Menge der Ereignisse	$[c] = \{k_3, k_7, k_{11}\}$
\ddot{A}_E Äquivalenzrelation	$[d] = \{k_4, k_8, k_{12}\}$

Die Ermittlung der minimalen Komponenten erfolgt durch Auswerten der Kontextkomponenten, getrennt nach Äquivalenzklassen. Abbildung 4.14 zeigt jeden Knoten $x \in M_E$ mit seinem Kontext $K_{E,x}^{(1)}$ aus dem zugrunde liegenden Ereignisgraphen G_E , sortiert nach Äquivalenzklassen. Grau hinterlegt sind diejenigen Knoten, die zu der Einteilung in Äquivalenzklassen geführt haben.

Auswerten der Kontextkomponenten: Die Auswertung der Kontextkomponenten liefert Kanten, die immer Teil einer Komponente sein wer-

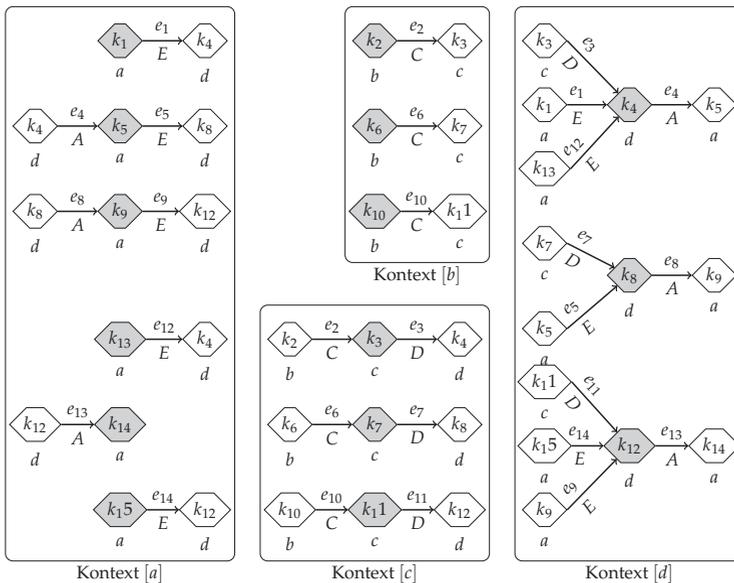


Abbildung 4.14: Kontext um Knoten mit gleicher Beschriftung

den. Diese seien in der Menge $Z_{E-E,init}$ zusammengefasst. Der Teilgraph von G_E , der alle Knoten und die Teilmenge $Z_{E-E,init}$ der Kantenmenge R_{E-E} enthält, lässt sich nach Gleichung 4.3 als Zerlegung $Z_{E,init} = (M_E ; Z_{E-E,init})$ beschreiben. Die Komponenten der Zerlegung $Z_{E,init}$ werden als minimale Komponenten bezeichnet. Eine minimale Komponente kann einen einzelnen Knoten darstellen oder einen Untergraphen von G_E .

Zu Beginn werden alle Kanten in $Z_{E-E,init}$ eingetragen. Enthält eine Äquivalenzklasse mehr als einen Knoten, dann sind ihre Kontextkomponenten miteinander zu vergleichen. Dies ist für alle dargestellten Äquivalenzklassen nach Abbildung 4.14 der Fall. Betrachtet man die Kontextkomponenten zu $[b]$ und $[c]$, so ist die Abbildung f_p für die Kontextkomponenten einer Klasse gleich. Die Auswertung führt zu keinen Veränderungen. Betrachtet man die Kontextkomponenten zu $[a]$ und $[d]$, so ist die Abbildung f_p für die Kontextkomponenten innerhalb einer Klasse unterschiedlich. Kanten mit gleicher Beschriftung und unterschiedlicher Häufigkeit werden aus der Menge $Z_{E-E,init}$ in die Menge $V_{E-E,init}$ übertragen.

Abbildung 4.15 zeigt das Ergebnis der Auswertung für jede Äquivalenzklasse. Aus der Darstellung sind diejenigen Kanten entfernt worden, für die die Auswertung zu Konsequenzen geführt hat. Jede Äquivalenzklasse wird getrennt betrachtet. Eine Kante, z.B. zwischen dem Knoten k_4 und k_5 ist aus der Darstellung von Kontext $[a]$ entfernt, nicht aber aus der Darstellung von Kontext $[d]$. Die Abbildung soll zum Ausdruck bringen, für welche Kanten die Auswertung zu Konsequenzen führt. Da die Auswertung nach Äquivalenzklassen getrennt durchgeführt wird, sind die Auswirkungen getrennt nach Äquivalenzklassen dargestellt.

Ist die Auswertung für alle Äquivalenzklassen abgeschlossen, dann ist die Menge $Z_{E-E,init}$ bekannt. Abbildung 4.16 zeigt die minimalen Komponenten. Die Kanten der Menge $Z_{E-E,init}$ sind als durchgezogene Pfeile dargestellt. Jeder Zusammenhang, der sich auf der Grundlage von $Z_{E-E,init}$ ergibt, stellt eine minimale Komponente dar. Ist ein Knoten nicht durch Kanten aus $Z_{E-E,init}$ mit anderen Knoten verbunden, so stellt der Knoten für sich eine minimale Komponente dar. Die Menge M_K^{min} fasst alle minimalen Komponenten zusammen. Nachfolgend sind die minimalen Komponenten in ihrer symbolischen Schreibweise durch Aufzählen der Knoten und der Kanten aufgelistet.

$$\begin{aligned} K_{E,x}^{min} &= \text{eine minimale Komponente} \\ M_K^{min} &= \text{Menge aller minimalen Komponenten} \end{aligned} \tag{4.25}$$

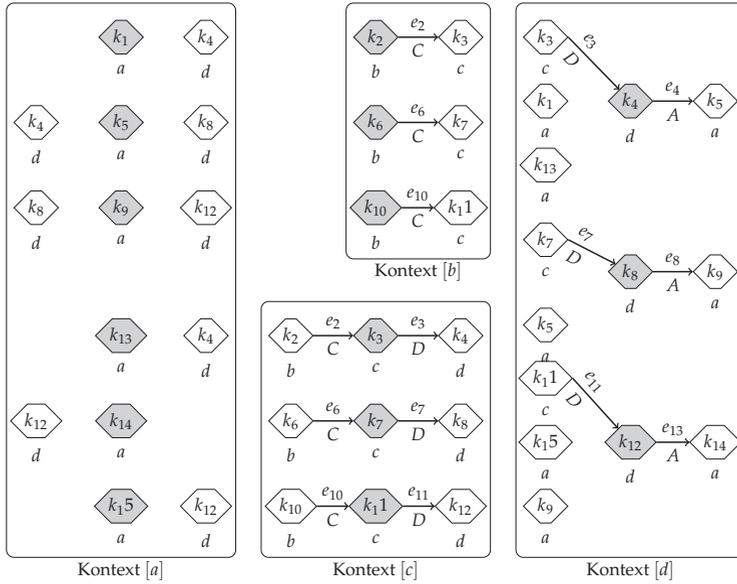


Abbildung 4.15: Bereinigte Kontextkomponenten

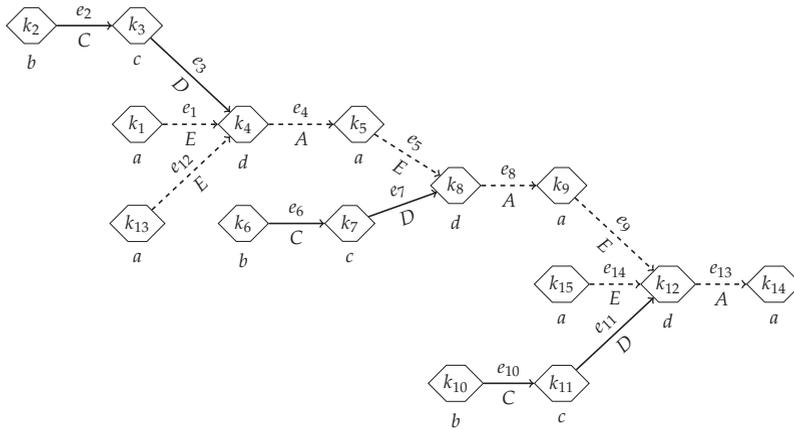


Abbildung 4.16: Minimale Komponenten

$$\begin{array}{ll}
K_{E,1}^{min} = (\{k_1\}; \{\}) & K_{E,2}^{min} = (\{k_2, k_3, k_4\}; \{e_2, e_3\}) \\
K_{E,5}^{min} = (\{k_5\}; \{\}) & K_{E,6}^{min} = (\{k_6, k_7, k_8\}; \{e_6, e_7\}) \\
K_{E,9}^{min} = (\{k_9\}; \{\}) & K_{E,10}^{min} = (\{k_{10}, k_{11}, k_{12}\}; \{e_{10}, e_{11}\}) \\
K_{E,13}^{min} = (\{k_{13}\}; \{\}) & K_{E,14}^{min} = (\{k_{14}\}; \{\}) \\
K_{E,15}^{min} = (\{k_{15}\}; \{\}) &
\end{array}$$

4.3.4 Entwickeln der Komponenten

Eine Komponente ist ein zusammenhängender Untergraph eines gegebenen Graphen. Die Abbildung f_P nach Gleichung 4.8 definiert eine Zeichenkette, die als Muster der Komponente bezeichnet wird. Das Konzept der Mustererkennung, das in dieser Arbeit beschrieben wird, basiert auf dem Vergleich von Mustern unterschiedlicher Komponenten. Um einen Ereignisgraphen G_E auf seine Muster hin zu untersuchen, ist es erforderlich, den Graphen in Komponenten zu zerlegen, diese auf ihre Muster abzubilden, um sie dann vergleichen zu können. Lassen sich unterschiedliche Komponenten auf dasselbe Muster abbilden, so handelt es sich nach dem hier zugrunde liegenden Verständnis um eine Mustererkennung.

Jede Zerlegung des Graphen lässt sich bewerten. Je mehr übereinstimmende Muster erkannt werden, desto besser ist die Bewertung. Die Vielzahl der möglichen Zerlegungen des Ereignisgraphen erfordert eine Systematik, um sicherzustellen, dass der gesamte Lösungsraum berücksichtigt wird. Im Rahmen dieser Arbeit erfolgt die Steuerung der Suche nach Zerlegungen auf der Grundlage eines Branch-and-Bound-Verfahrens. Hierbei werden nicht einzelne Kanten variiert, sondern mit jedem Schritt Mengen an Kanten. Die Kantenmengen ergeben sich aus der Betrachtung von Komponenten. Nur diejenigen Komponenten werden betrachtet, die zu einer Einsparung führen können. Dies ist für Komponenten der Fall, die (1) sich auf dasselbe Muster abbilden lassen, die (2) mindestens aus zwei Knoten und einer Kante bestehen und die (3) gleichzeitig und überschneidungsfrei als Untergraphen in G_E auftreten.

Dieser Abschnitt beschreibt, wie die Menge der relevanten Komponenten im Vorfeld bestimmt werden kann. Ziel ist es, den Aufwand bei der Bestimmung dieser Menge zu minimieren, dabei aber sicherzustellen, dass alle Lösungen ermittelt werden. Das Vorgehen berücksichtigt die Struktur des Ereignisgraphen.

Prinzip: Um die Vollständigkeit bei der Ermittlung der Menge der Komponenten sicherzustellen, wird knotenweise vorgegangen. Von einem Knoten ausgehend, werden all diejenigen Komponenten entwickelt, die diesen Knoten enthalten. Für jeden Zusammenhang, der sich auf der Grundlage der Nachbarschaft eines Knotens ergibt, wird eine Komponente angelegt und in einer Komponentenmenge vermerkt. Schrittweise wird der Grad der Nachbarschaft erweitert. Der Umfang an Knoten und Kanten der neu entstehenden Komponenten wächst stetig. Erreicht eine Komponente eine gewisse Größe, kann abgebrochen werden. Damit sich das Muster einer Komponente wiederholen kann, darf eine Komponente nur maximal halb so groß werden wie der Graph selbst.

Bei dem knotenweisen Vorgehen werden Komponenten entwickelt, die kleiner sind als die minimalen Komponenten. Weil ihr Einsparpotential bezogen auf die Bewertung a_ω aber kleiner ist als das der minimalen Komponenten, werden sie niemals Teil einer optimalen Lösung sein. Anstatt jeden Knoten einzeln zu betrachten, werden daher Zellen betrachtet. Jede minimale Komponente aus M_K^{min} nach Abschnitt 4.3.3 sei eine Zelle. Diese sind nicht teilbar. Betrachtet wird das Wachsen der Zellen. Jede Zelle kann entlang derjenigen Kanten wachsen, für die gilt: (1) Sie sind ein Element des Graphen G_E und (2) sie zeigen von außen in die Zelle hinein bzw. sie zeigen von innen aus der Zelle heraus. Der Anfangs- und der Endknoten einer Kante liegen immer in unterschiedlichen Zellen. Das Wachstum erfolgt schrittweise entlang genau einer Kante. Mit jedem Wachstumsschritt entsteht eine neue Zelle. Diese besteht aus den Knoten und Kanten der jeweiligen Zellen, die den Anfangs- bzw. den Endknoten einer Kante enthalten und aus der betrachteten Kante.

Das Wachstum lässt sich in einem Baum darstellen. Jeder Knoten des Baums steht für eine Zelle und jede Zelle beschreibt eine zusammenhängende Komponente. Die Wurzel eines Baums ist immer eine minimale Komponente. Mit jedem Schritt in den Baum hinein wird ein Knoten - Zelle - um eine minimale Komponente - Zelle - erweitert. Das Wachstum entlang der Kanten zu den Vorgängern und entlang der Kanten zu den Nachfolgern wird unabhängig voneinander in getrennten Bäumen untersucht. Für jede minimale Komponente wird ein Vorgängerbaum und ein Nachfolgerbaum entwickelt. Alle Komponenten, die sich auf der Grundlage des Vorgängerbaums ergeben, werden in der Menge M_K^V zusammengefasst und diejenigen, die sich auf der Grundlage des Nachfolgerbaums ergeben, werden in der Menge M_K^N zusammengefasst.

Der Vorgänger- und der Nachfolgerbaum, für die die Wurzel $k \in M_K^{min}$ identisch ist, bilden ein Paar. Jede Komponente aus $M_K^{V,k}$ und $M_K^{N,k}$ enthält

die Knoten und Kanten der gemeinsamen Zelle k . Um alle Komponenten zu ermitteln, die die Knoten und Kanten der gemeinsamen Zelle k enthalten, sind die Knoten des Vorgängerbaums mit den Knoten des Nachfolgerbaums zu kombinieren. Die obere Schranke für den Aufwand zur Bestimmung der Kombinationen ist $O(|M_K^{V,k}| \cdot |M_K^{N,k}|)$. Nicht alle Kombinationen müssen ermittelt werden. Nur diejenigen Kombinationen sind von Interesse, die nicht größer sind als die Hälfte des Graphen G_E . Das Ergebnis der Auswertung wird in $M_K^{VN,k}$ zusammengefasst. Die Vereinigungsmenge M_K^k umfasst alle Komponenten, die k enthalten.

$$M_K^k = M_K^{V,k} \cup M_K^{N,k} \cup M_K^{VN,k} \quad (4.26)$$

M_K^k	Menge an Komponenten, die k enthalten
k	minimale Komponente $k \in M_K^{min}$
$M_K^{V,k}$	Menge an Komponenten aus dem Vorgängerbaum
$M_K^{N,k}$	Menge an Komponenten aus dem Nachfolgerbaum
$M_K^{VN,k}$	Menge an Komponenten aus der Kombination der Elemente von $M_K^{V,k}$ und $M_K^{N,k}$

Die Auswertung ist für jede minimale Komponente des Graphen durchzuführen. Die Teilmengen M_K^k werden in der Menge M_K zusammengefasst. Nur diejenigen Komponenten werden übertragen, die mindestens zwei Knoten und eine Kante umfassen.

$$M_K = \bigcup_{k \in M_K^{min}} M_K^k \quad (4.27)$$

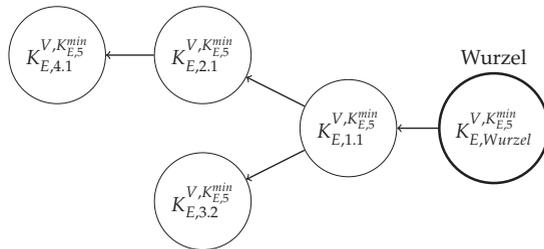
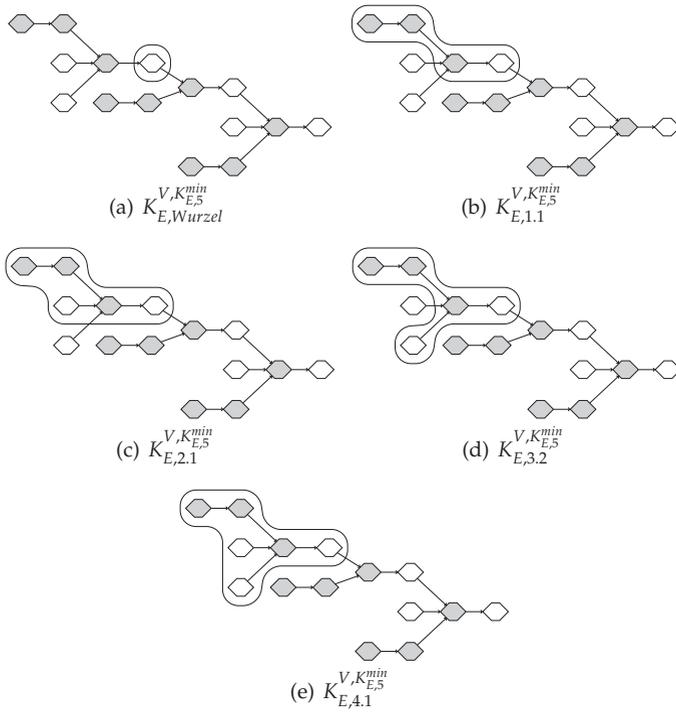
M_K	Menge aller Komponenten, die mindestens zwei Knoten enthalten und die maximal die Hälfte aller Knoten bzw. Kanten umfasst
M_K^k	Menge an Komponenten, die k enthalten
k	minimale Komponente $k \in M_K^{min}$

Vorgängerbaum: Abbildung 4.17 zeigt das Wachsen der Vorgängerkomponenten für die minimale Komponente $K_{E,5}^{min}$. Jeder Knoten des Baums hat einen Identifikator. Der Ursprungsknoten wird mit *Wurzel* bezeichnet.

Der Identifikator setzt sich aus zwei Zählern $x.y$ zusammen. Der erste Zähler x ist ein Laufindex. Mit jedem Knoten, der neu entwickelt wird, zählt x hoch. Sein Wert beschreibt die Reihenfolge, in der der Baum durchlaufen wird. Der zweite Zähler y beschreibt die Position in der Reihenfolge von Kindknoten, die denselben Vater haben. Zur Identifikation reicht bereits der erste Zähler x . Der zweite Zähler dient der Anschauung und ist nur für die Beschreibung des Verfahrens von Bedeutung. Die grau hinterlegten Knoten stellen nicht teilbare zusammenhängende minimale Komponenten dar. Die nicht gefärbten Knoten stellen minimale Komponenten dar, die nur aus einem Knoten bestehen.

Der Knoten k_5 stellt den Wurzelknoten $K_{E,Wurzel}^{V,K_{E,5}^{min}}$ des Vorgängerbaums dar (4.17(a)). Eine Kante zeigt von außen in die Zelle hinein. Entlang dieser Kante erfolgt die erste Verschmelzung. Es entsteht die neue Zelle $K_{E,1.1}^{V,K_{E,5}^{min}}$ (4.17(b)). Da die Vorgängerzelle aus einer minimalen Komponente mit drei zusammenhängenden Knoten besteht, werden diese gleichzeitig in die neue Zelle übertragen. Die Zelle $K_{E,1.1}^{V,K_{E,5}^{min}}$ wird zur aktuellen Zelle. Für den nächsten Schritt existieren zwei Kanten, die von außen in die aktuelle Zelle hineinzeigen. Jede Kante wird einzeln betrachtet. Die Reihenfolge hat keine Bedeutung. Es entstehen die Zellen $K_{E,2.1}^{V,K_{E,5}^{min}}$ (4.17(c)) und $K_{E,3.2}^{V,K_{E,5}^{min}}$ (4.17(d)). Sind alle Kinder eines Vaterknotens erzeugt, wird das erste Kind als aktuelles Kind gesetzt und weiter verschmolzen. Es entsteht die Zelle $K_{E,4.1}^{V,K_{E,5}^{min}}$ (4.17(e)). Für diese Zelle existieren keine weiteren eingehenden Kanten mehr. Das Wachstum ist auf diesem Pfad abgeschlossen. Der aktuelle Knoten wird terminiert. Die Aufmerksamkeit wird an den Vater zurückgegeben. Von dem Vater aus wird die Aufmerksamkeit an das nächste Kind übertragen, sofern es noch nicht terminierte Kinder gibt. In diesem Fall wird die Zelle $K_{E,3.2}^{V,K_{E,5}^{min}}$ als aktuelle Zelle gesetzt. Es existiert eine eingehende Kante. Die Verschmelzung entlang dieser Kante führt aber zu einer Zelle, die bereits existiert. Bevor eine neue Zelle angelegt wird, wird überprüft, ob bereits eine gleichartige Zelle existiert. Ist dies der Fall, wird kein Kind angelegt. Alle Komponenten, die sich auf der Grundlage des Vorgängerbaums ergeben, werden in der Menge M_K^V zusammengefasst.

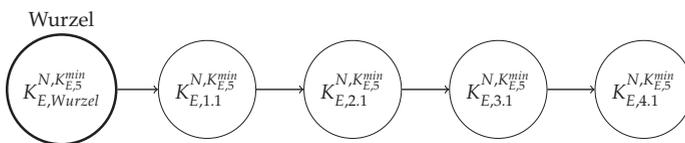
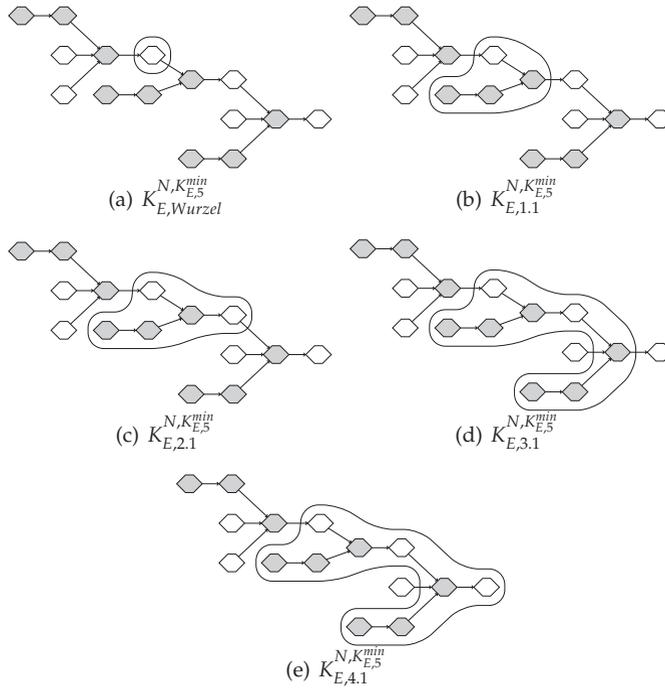
Nachfolgerbaum: Analog zu der Vorgehensweise des Vorgängerbaums wird ein Nachfolgerbaum entwickelt. Die einzelnen Schritte sind in Abbildung 4.18 dargestellt. Alle Komponenten, die sich auf der Grundlage des



$$M_K^{V,K_{E,5}^{min}} = \{K_{E,Wurzel}^{V,K_{E,5}^{min}}, K_{E,1,1}^{V,K_{E,5}^{min}}, K_{E,2,1}^{V,K_{E,5}^{min}}, K_{E,3,2}^{V,K_{E,5}^{min}}, K_{E,4,1}^{V,K_{E,5}^{min}}\}$$

(f) Vorgängerbaum

Abbildung 4.17: Entwickeln der Vorgängerkomponenten



$$M_K^{N,K_{E,5}^{min}} = \{K_{E,Wurzel}^{N,K_{E,5}^{min}}, K_{E,1,1}^{N,K_{E,5}^{min}}, K_{E,2,1}^{N,K_{E,5}^{min}}, K_{E,3,1}^{N,K_{E,5}^{min}}, K_{E,4,1}^{N,K_{E,5}^{min}}\}$$

(f) Nachfolgerbaum

Abbildung 4.18: Entwickeln der Nachfolgerkomponenten

Nachfolgerbaums ergeben, werden in der Menge M_K^N zusammengefasst.

Menge der relevanten Komponenten: Der Vorgänger- und der Nachfolgerbaum, für die die Wurzel identisch ist, bilden ein Paar. Jeder Knoten dieser beiden Bäume repräsentiert eine Komponente, die die Knoten und Kanten der gemeinsamen Zelle enthält. Es existieren noch weitere Komponenten im Graphen mit dieser Eigenschaft. Um alle Komponenten zu ermitteln, die die Knoten und Kanten dieser Zelle enthalten, sind die Knoten des Vorgängerbaums mit den Knoten des Nachfolgerbaums zu kombinieren. Nur diejenigen Kombinationen sind von Interesse, die nicht größer sind als die Hälfte des Graphen G_E .

Für die Komponentenmengen aus dem Vorgänger- und dem Nachfolgerbaum von $K_{E,5}^{min}$ und ihre Kombination ergeben sich folgende Mengen:

$$\begin{aligned} M_K^{V,K_{E,5}^{min}} &= \{K_{E,Wurzel}^{V,K_{E,5}^{min}}, K_{E,1.1}^{V,K_{E,5}^{min}}, K_{E,2.1}^{V,K_{E,5}^{min}}, K_{E,3.2}^{V,K_{E,5}^{min}}, K_{E,4.1}^{V,K_{E,5}^{min}}\} \\ M_K^{N,K_{E,5}^{min}} &= \{K_{E,Wurzel}^{N,K_{E,5}^{min}}, K_{E,1.1}^{N,K_{E,5}^{min}}, K_{E,2.1}^{N,K_{E,5}^{min}}, K_{E,3.1}^{N,K_{E,5}^{min}}, K_{E,4.1}^{N,K_{E,5}^{min}}\} \\ M_K^{VN,K_{E,5}^{min}} &= \{\} \end{aligned}$$

Die Vereinigung dieser Mengen unter Berücksichtigung der Größenkriterien führt zu der Menge der relevanten Komponenten, die $K_{E,5}^{min}$ enthalten.

$$\begin{aligned} M_K^{K_{E,5}^{min}} &= \{ K_{E,Wurzel}^{V,K_{E,5}^{min}}, K_{E,1.1}^{V,K_{E,5}^{min}}, K_{E,2.1}^{V,K_{E,5}^{min}}, K_{E,3.2}^{V,K_{E,5}^{min}}, \\ & K_{E,4.1}^{V,K_{E,5}^{min}}, K_{E,1.1}^{N,K_{E,5}^{min}}, K_{E,2.1}^{N,K_{E,5}^{min}} \} \end{aligned}$$

Diese Auswertung ist für jede minimale Komponente vorzunehmen. Abbildung 4.19 zeigt das Ergebnis der Auswertung. Insgesamt lassen sich 20 Komponenten identifizieren, die sich auf sechs verschiedene Muster abbilden lassen. Jedes Muster ist als Graph dargestellt. Komponenten, die sich auf ein bestimmtes Muster abbilden lassen, sind in ihrer symbolischen Schreibweise durch Aufzählen der Knoten und der Kanten unterhalb des jeweiligen Graphen aufgelistet. Die Komponenten erfüllen die Anforderungen, die aus den Bedingungen (1) und (2) resultieren, die am Anfang dieses Abschnitts genannt wurden. Für dieses Beispiel gibt es keine Komponenten, die kleiner sind als die minimalen Komponenten. Jede Komponente besteht aus mindestens zwei Knoten und einer Kante. Keine der

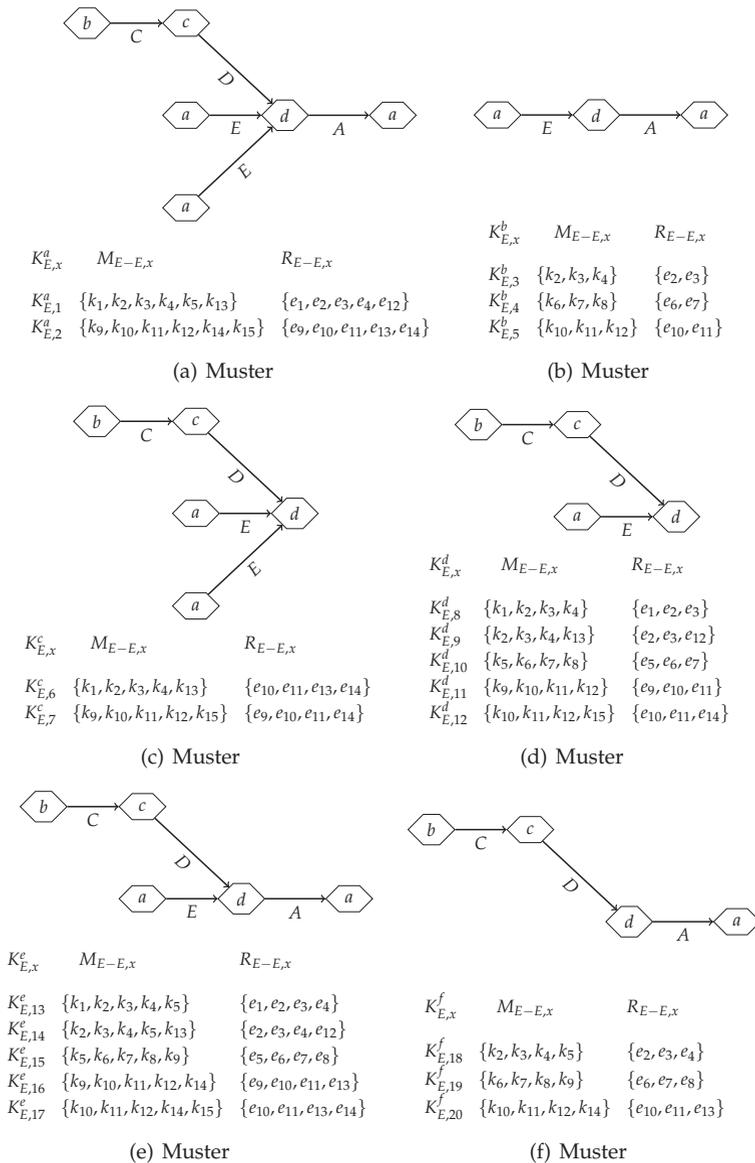


Abbildung 4.19: Komponenten nach Muster

Komponenten enthält mehr als die Hälfte der Knoten bzw. Kanten insgesamt. Jede Menge an Komponenten, die sich auf dasselbe Muster abbilden lässt, ist größer als eins.

Die optimale Lösung bzw. die optimalen Lösungen sind Zerlegungen von G_E , die sich aus den Komponenten zusammensetzen, die in Abbildung 4.19 dargestellt sind. Nur diese Komponenten können zu einer günstigen Bewertung a_ω führen, da sie das Potential haben, durch geeignete Kombination Einsparungen aufgrund wiederkehrender Teilprozesse zu bewirken.

Eine optimale Lösung ergibt sich aus einer Kombination dieser Komponenten. Damit sich eine Einsparung ergibt, müssen gleichzeitig mindestens zwei Komponenten mit demselben Muster zum Einsatz kommen. In dem nächsten Abschnitt geht es um die Ermittlung überschneidungsfreier Komponentenkombinationen.

4.3.5 Überschneidungsfreie Komponentenkombinationen

Abbildung 4.19 zeigt eine Einteilung der relevanten Komponenten für das Beispiel aus Abbildung 4.3 entsprechend ihres Musters in Äquivalenzklassen. Da sich die Komponenten derselben Klasse überschneiden können, sich überschneidende Komponenten aber nicht gleichzeitig unabhängige Teilprozesse eines Prozesses sein können, werden für jede Äquivalenzklasse getrennt diejenigen Teilmengen an Komponenten bestimmt, die überschneidungsfrei sind.

Die Ermittlung von Komponentenkombinationen, die sich aus der Menge der Komponenten einer Äquivalenzklasse bestimmen lassen, erfolgt auf der Grundlage einer Baumstruktur. Abbildung 4.20 zeigt einen Baum für die Äquivalenzklasse d nach Teilbild 4.19(d). Die hier beschriebene Vorgehensweise fasst die in Abschnitt 4.3.2 Absatz *Komponentenkombinationen* getrennt behandelten Schritte *Auflösen der Überschneidungen* und *Kombinieren der überschneidungsfreien Komponenten* in einer Struktur zusammen. Für diese Struktur lässt sich nachvollziehbar die obere Schranke des Aufwands angeben.

Prinzip: Betrachtet wird die Komponentenmenge einer Äquivalenzklasse. Jeder Kante des Baums ist eine Komponente zugeordnet und jedem Knoten eine Menge an Komponenten. Diese ergibt sich aus der Vereinigung der Komponenten entlang des Pfads. Solange die Menge der Komponenten eines Knotens überschneidungsfrei ist, wird dieser weiter verzweigt.

Verzweigen: Die Komponenten einer Äquivalenzklasse seien indiziert. Kinder werden für diejenigen Komponenten angelegt, deren Index i größer ist als der Index des Vaters. Dieses Vorgehen stellt sicher, dass die Menge der Kombinationen ohne Wiederholungen vollständig ermittelt wird. Die obere Schranke für den Aufwand ist $O(n!)$, wenn n die Anzahl der Komponenten der betrachteten Äquivalenzklasse beschreibt. Dieser Aufwand wird nur erreicht, wenn alle Komponenten überschneidungsfrei sind.

Terminieren: Sind die Komponenten, die zu einem Knoten gehören, nicht überschneidungsfrei, wird seine Komponentenmenge durch die leere Menge ersetzt und der Knoten terminiert. Dieses Vorgehen reduziert den Aufwand. Je mehr Überschneidungen es gibt, desto größer ist die Einsparung.

Nachdem der Baum entwickelt ist, stehen die Komponentenkombinationen fest. Diejenigen, mit mehr als zwei Elementen, können zu einer Einsparung führen. Für das Beispiel sind diese in der Abbildung 4.20 durch ihre Umrandung hervorgehoben und durch die Bezeichner $C_{M_K^d}^d$ benannt. Die obere Schranke beträgt $5! = 120$. Der realisierte Aufwand dagegen umfasst nur 25 Knoten.

Für jede Äquivalenzklasse ist getrennt ein eigener Baum zu entwickeln. Die Menge aller Komponentenkombinationen ergibt sich aus der Vereinigung.

$$C_K = \bigcup_{\ddot{a} \in M_K / \ddot{A}_K} C_{M_K^{\ddot{a}}} \quad (4.28)$$

C_K	Menge aller überschneidungsfreien Komponentenkombinationen
M_K / \ddot{A}_K	Quotientenmenge: Menge aller Äquivalenzklassen
\ddot{A}_K	Äquivalenzrelation
$C_{M_K^{\ddot{a}}}$	überschneidungsfreie Komponenten- kombination einer Äquivalenzklasse

Abbildung 4.21 zeigt alle überschneidungsfreien Komponentenkombinationen für das betrachtete Beispiel.

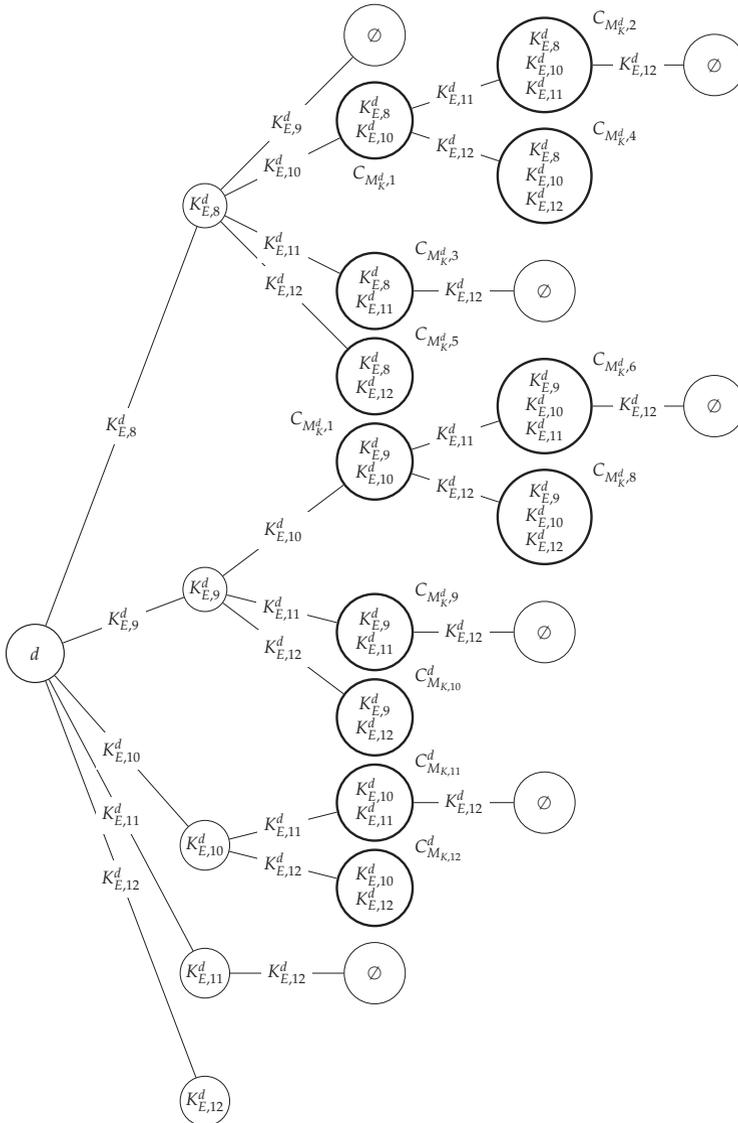
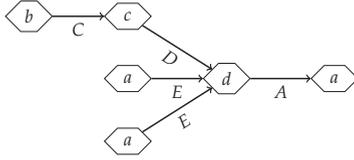


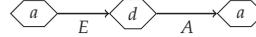
Abbildung 4.20: Komponentenkombinationen für Muster d



$$M_K^a = \{K_{E,11}^a, K_{E,2}^a\}$$

$$C_{M_K^a,1} = \{K_{E,1}^a, K_{E,2}^a\}$$

(a) Muster



$$M_K^b = \{K_{E,3}^b, K_{E,A}^b, K_{E,5}^b\}$$

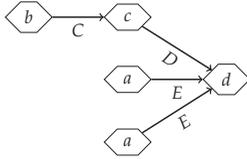
$$C_{M_K^b,1} = \{K_{E,3}^b, K_{E,A}^b\}$$

$$C_{M_K^b,2} = \{K_{E,3}^b, K_{E,5}^b\}$$

$$C_{M_K^b,3} = \{K_{E,A}^b, K_{E,5}^b\}$$

$$C_{M_K^b,A} = \{K_{E,3}^b, K_{E,A}^b, K_{E,5}^b\}$$

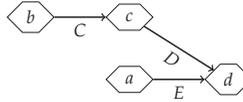
(b) Muster



$$M_K^c = \{K_{E,6}^c, K_{E,6}^c\}$$

$$C_{M_K^c,1} = \{K_{E,6}^c, K_{E,6}^c\}$$

(c) Muster



$$M_K^d = \{K_{E,8}^d, K_{E,9}^d, K_{E,10}^d, K_{E,11}^d, K_{E,12}^d\}$$

$$C_{M_K^d,1} = \{K_{E,8}^d, K_{E,10}^d\}$$

$$C_{M_K^d,2} = \{K_{E,8}^d, K_{E,10}^d, K_{E,11}^d\}$$

$$C_{M_K^d,3} = \{K_{E,8}^d, K_{E,11}^d\}$$

$$C_{M_K^d,4} = \{K_{E,8}^d, K_{E,10}^d, K_{E,12}^d\}$$

$$C_{M_K^d,5} = \{K_{E,8}^d, K_{E,12}^d\}$$

$$C_{M_K^d,6} = \{K_{E,9}^d, K_{E,10}^d, K_{E,11}^d\}$$

$$C_{M_K^d,7} = \{K_{E,9}^d, K_{E,10}^d\}$$

$$C_{M_K^d,8} = \{K_{E,9}^d, K_{E,10}^d, K_{E,12}^d\}$$

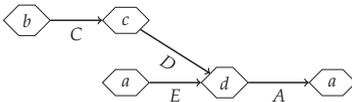
$$C_{M_K^d,9} = \{K_{E,9}^d, K_{E,11}^d\}$$

$$C_{M_K^d,10} = \{K_{E,9}^d, K_{E,12}^d\}$$

$$C_{M_K^d,11} = \{K_{E,10}^d, K_{E,11}^d\}$$

$$C_{M_K^d,12} = \{K_{E,10}^d, K_{E,12}^d\}$$

(d) Muster



$$M_K^e = \{K_{E,13}^e, K_{E,14}^e, K_{E,15}^e, K_{E,16}^e, K_{E,17}^e\}$$

$$C_{M_K^e,1} = \{K_{E,13}^e, K_{E,16}^e\}$$

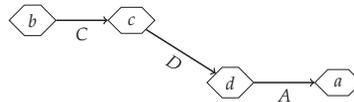
$$C_{M_K^e,2} = \{K_{E,13}^e, K_{E,17}^e\}$$

$$C_{M_K^e,3} = \{K_{E,14}^e, K_{E,16}^e\}$$

$$C_{M_K^e,4} = \{K_{E,14}^e, K_{E,17}^e\}$$

$$C_{M_K^e,5} = \{K_{E,15}^e, K_{E,17}^e\}$$

(e) Muster



$$M_K^f = \{K_{E,18}^f, K_{E,19}^f, K_{E,20}^f\}$$

$$C_{M_K^f,1} = \{K_{E,18}^f, K_{E,19}^f\}$$

$$C_{M_K^f,2} = \{K_{E,18}^f, K_{E,20}^f\}$$

$$C_{M_K^f,3} = \{K_{E,19}^f, K_{E,20}^f\}$$

$$C_{M_K^f,4} = \{K_{E,18}^f, K_{E,19}^f, K_{E,20}^f\}$$

(f) Muster

Abbildung 4.21: Komponentenkombinationen nach Muster

4.3.6 Bewertung von Komponentenkombinationen

Im Rahmen des Branch-and-Bound-Verfahrens variiert mit jedem Schritt die Menge der Komponenten. Welche Komponenten in einem Schritt hinzukommen bzw. wegfallen, ist abhängig von den überschneidungsfreien Komponentenkombinationen. Die Reihenfolge, in der die Komponentenkombinationen bei der Variation berücksichtigt werden, hat einen Einfluss auf den Lösungsaufwand des Verfahrens. In diesem Abschnitt geht es um die Bewertung von Komponentenkombinationen. Sie erlaubt es, unterschiedliche Komponentenkombinationen miteinander zu vergleichen und sie zu ordnen.

Ohne die optimale Lösung zu kennen, lässt sich keine optimale Reihenfolge in der Menge aller Komponentenkombinationen bestimmen. Hierin besteht daher nicht die Motivation für die Bewertung. Ziel der Bewertung ist es, ein effizientes Abschätzen an den Knoten zu ermöglichen, um den Lösungsaufwand durch das Eingrenzen des Lösungsraums herabzusetzen. Die Abschätzung an einem Knoten prüft, ob für die Spezifikation der verbleibenden Menge an Knoten und Kanten der noch verfügbare Aufwand ausreicht oder nicht. Reicht er nicht aus, kann auf eine Verzweigung des aktuellen Knotens verzichtet werden. Gelingt es, die Kinder eines Knotens so zu ordnen, dass das Einsparpotential der Komponentenkombinationen mit jedem Kind kleiner wird, dann betrifft die Abschätzung eines Kindes zwangsläufig auch alle nachfolgenden Kinder.

Ein Kriterium, das diese Anforderungen erfüllt, folgt aus der normierten Bewertung $n_Z(Z_E)$ nach Gleichung 4.13. Sie kann als Maß für das Einsparpotential einer Komponentenkombination interpretiert werden, das sich durch die Berücksichtigung wiederkehrender Teilprozesse ergibt. An jedem Knoten des Entscheidungsbaums, der verzweigt wird, existiert eine Menge an Komponentenkombinationen. Für jede Komponentenkombination wird ein Kind angelegt. Auf der Grundlage der normierten Bewertung wird eine Reihenfolge für die Kindknoten festgelegt. Die Reihenfolge ordnet die Kinder nach ihrem Einsparpotential. Reicht das Einsparpotential einer Komponentenkombination nicht mehr aus, um mit dem verbleibenden Aufwand auszukommen, können der aktuelle Knoten und auch die nachfolgenden Kindknoten terminiert werden.

Eine Komponentenkombination wird in diesem Zusammenhang als ein Graph bzw. eine Zerlegung aufgefasst, die nach der folgenden Gleichung ausgewertet werden kann:

$$n_Z(Z_E) = \frac{a_Z(Z_E)}{a(Z_E)} \quad \text{siehe Gleichung 4.13}$$

Die Terme berechnen sich auf der Grundlage der Gleichungen:

$$a(G_E) = |M_E| + |R_{E-E}| \quad \text{siehe Gleichung 4.11}$$

$$a_Z(Z_E) = |M_K| + \sum_{x \in M_R} a(x) \quad \text{siehe Gleichung 4.12}$$

Für Z_E und G_E werden die Komponentenkombinationen eingesetzt. Die Auswertung $a_Z(Z_E)$ liefert den Aufwand für eine Komponentenkombination unter Berücksichtigung der wiederkehrenden Teilprozesse. Dieser Aufwand wird auf die Anzahl der Knoten und Kanten bezogen, aus denen die Komponentenkombination besteht. Die Auswertung von $a(G_E)$ liefert genau diese Anzahl. Der Quotient aus $a_Z(Z_E)$ und $a(G_E)$ stellt eine Normierung dar. Die normierte Bewertung einer Komponentenkombination liefert einen Wert, der größer ist als 0 und immer kleiner ist als 1. Der Abstand des Werts zu dem Wert 1 beschreibt das Einsparpotential.

In der nachfolgenden Berechnung wird stellvertretend, am Beispiel der Komponentenkombinationen $C_{M_K^f,2}$ und $C_{M_K^f,4}$ nach Abbildung 4.22, die

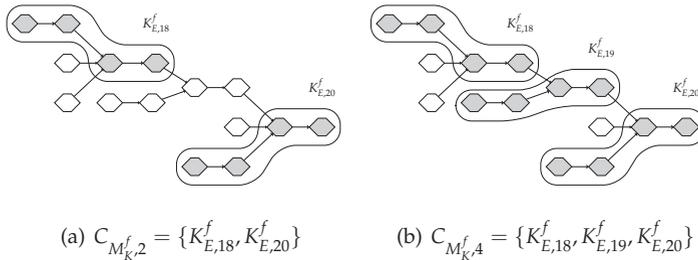


Abbildung 4.22: Komponentenkombinationen

normierte Bewertung ermittelt. Da sich alle Komponenten auf dasselbe Muster abbilden lassen, wird für beide willkürlich die Komponente $K_{E,18}^f$ als Repräsentant ausgewählt.

Z_E bzw. G_E	$C_{M_K^f,2}$	$C_{M_K^f,4}$
Menge der Komponenten M_K	$\{K_{E,18}^f, K_{E,20}^f\}$	$\{K_{E,18}^f, K_{E,19}^f, K_{E,20}^f\}$
Repräsentantensystem M_R	$\{K_{E,18}^f\}$	$\{K_{E,18}^f\}$

$$\begin{aligned}
a_Z(C_{M_k^f,2}) &= |\{K_{E,18}^f, K_{E,20}^f\}| + a(K_{E,18}^f) &= 2 + 7 &= 9 \\
a_Z(C_{M_k^f,4}) &= |\{K_{E,18}^f, K_{E,19}^f, K_{E,20}^f\}| + a(K_{E,18}^f) &= 3 + 7 &= 10 \\
a(C_{M_k^f,2}) &= a(K_{E,18}^f) + a(K_{E,20}^f) &= 7 + 7 &= 14 \\
a(C_{M_k^f,4}) &= a(K_{E,18}^f) + a(K_{E,19}^f) + a(K_{E,20}^f) &= 7 + 7 + 7 &= 21 \\
n_Z(C_{M_k^f,2}) &= \frac{a_Z(C_{M_k^f,2})}{a(C_{M_k^f,2})} = \frac{9}{14} = 0.64 \\
n_Z(C_{M_k^f,4}) &= \frac{a_Z(C_{M_k^f,4})}{a(C_{M_k^f,4})} = \frac{10}{21} = 0.48
\end{aligned}$$

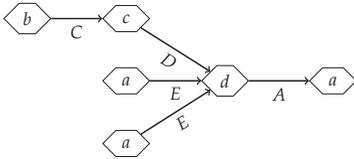
Abbildung 4.23 zeigt die normierten Bewertungen aller Komponenten-kombinationen. Entsprechend dieser Bewertungen werden die Kinder eines Knotens aufsteigend geordnet. Komponentenkombinationen mit der kleinsten normierten Bewertung haben das größte Einsparpotential.

Ein Knoten des Entscheidungsbaums, der verzweigt wird, ist verknüpft mit einer Menge an Knoten und Kanten, die noch nicht fixiert sind. Ihre Anzahl sei die Summe aus $anz_{Knoten} + anz_{Kanten}$. Die normierte Bewertung $n_Z(C_{M_k^p,x})$ erlaubt es, ohne die betroffenen Knoten und Kanten zu analysieren, einen verbleibenden, mindestens erforderlichen Aufwand zu ermitteln. Bildet man das Produkt aus $anz_{Knoten} + anz_{Kanten}$ und der normierten Bewertung $n_Z(C_{M_k^p,x})$, ergibt sich eine Abschätzung für den erforderlichen Aufwand¹⁰ am aktuellen Kindknoten. Ist der erforderliche Aufwand größer als der verbleibende verfügbare Aufwand, so kann der aktuelle Kindknoten terminiert werden. Dies trifft auch für alle nachfolgenden Kindknoten zu, weil diese nach der normierten Bewertung aufsteigend geordnet sind. Das Kriterium erlaubt es, alle nachfolgenden Kinder, die denselben Vater haben wie der aktuelle Kindknoten, und ihre Teilbäume von der Suche auszuschließen, ohne dabei Gefahr zu laufen, eine optimale Lösung zu verpassen.

4.3.7 Algorithmus

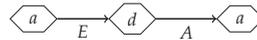
In diesem Abschnitt wird der Algorithmus für die Suche nach optimalen Zerlegungen behandelt. Seine Betrachtung bringt die beschriebenen

¹⁰ siehe Abschnitt 4.3.2 Terminieren (Bounding)



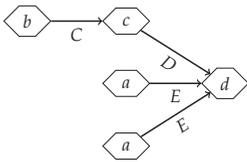
$$n_Z(C_{M_k^c,1}) = 0.59$$

(a) Muster



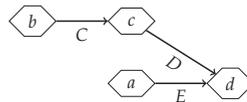
$$\begin{aligned} n_Z(C_{M_k^d,A}) &= 0.59 \\ n_Z(C_{M_k^d,x}) &= 0.70 \quad x = 1, 2, 3 \end{aligned}$$

(b) Muster



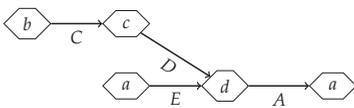
$$n_Z(C_{M_k^c,1}) = 0.61$$

(c) Muster



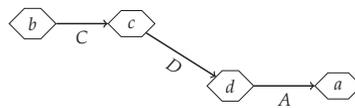
$$\begin{aligned} n_Z(C_{M_k^d,x}) &= 0.48 \quad x = 2, 4, 6, 8 \\ n_Z(C_{M_k^d,x}) &= 0.64 \quad x = 1, 3, 5, 7, 9, \dots, 12 \end{aligned}$$

(d) Muster



$$n_Z(C_{M_k^c,x}) = 0.61 \quad x = 1, \dots, 5$$

(e) Muster



$$\begin{aligned} n_Z(C_{M_k^d,x}) &= 0.48 \quad x = 4 \\ n_Z(C_{M_k^d,x}) &= 0.64 \quad x = 1, 2, 3 \end{aligned}$$

(f) Muster

Abbildung 4.23: Normierte Bewertung der Komponentenkombinationen

Aspekte in einen Zusammenhang. Der Algorithmus 14 beschreibt prinzipiell, wie die Entwicklung des Entscheidungsbaums abläuft, nachdem dieser initialisiert ist. Er umfasst das Verzweigen und das Terminieren der Knoten. Jeder Schritt wird zeilenweise erläutert.

Algorithmus 1: Suche nach optimalen Zerlegungen

Algorithmus : verzweige

Input : \mathbf{w}_Ω, C_K

```

1  $a_{\mathbf{w}_\Omega} \leftarrow \mathbf{w}_\Omega.\text{holeAufwand}();$ 
2 if  $a_{\mathbf{w}_\Omega} > a_{\omega, \text{opt}}$  then
3   return;
4  $C_{M_K^p} \leftarrow C_K.\text{holeMinimaleKombination}();$ 
5 while  $C_{M_K^p} \neq \text{null}$  do
6    $a_{\mathbf{w}_\Omega}^{\text{erf}} \leftarrow \mathbf{w}_\Omega.\text{holeAbschaetzung}(C_{M_K^p});$ 
7   if  $a_{\omega, \text{opt}} - a_{\mathbf{w}_\Omega} < a_{\mathbf{w}_\Omega}^{\text{erf}}$  then
8     break;
9    $\mathbf{w}_{\Omega, \text{Kind}} \leftarrow \mathbf{w}_\Omega.\text{erzeugeKind}(C_{M_K^p});$ 
10   $C_{K, \text{Kind}} \leftarrow \mathbf{w}_{\Omega, \text{Kind}}.\text{holeVerbleibendeKombinationen}(C_K);$ 
11  verzweige( $\mathbf{w}_{\Omega, \text{Kind}}, C_{K, \text{Kind}}$ );
12   $C_K.\text{entferne}(C_{M_K^p});$ 
13   $C_{M_K^p} \leftarrow C_K.\text{holeMinimaleKombination}();$ 
14 registriere( $\mathbf{w}_\Omega$ );
```

1. Der Vektor \mathbf{w}_Ω enthält variable und fixierte Kanten. Für den fixierten Anteil lässt sich ein Aufwand bestimmen. Der Aufwand $a_{\mathbf{w}_\Omega}$ beschreibt den bereits verzehrten Aufwand zur Spezifikation der Kanten aus den Mengen $Z_{E-E, \Omega}$ und $S_{E-E, \Omega}$.
2. Ist der verzehrte Aufwand $a_{\mathbf{w}_\Omega}$ größer als der mit der aktuell besten Lösung verbundene Aufwand $a_{\omega, \text{opt}}$, dann kann dieser Knoten terminiert werden.

3. Der aktuelle Knoten wird terminiert. Die Suche wird für den Vater fortgesetzt.
4. Aus der Menge der verbleibenden Komponentenkombinationen C_K wird diejenige ermittelt, für die die normierte Bewertung minimal ist. Diese Komponentenkombination wird in $C_{M_K^p}$ gespeichert.
5. Geordnet nach Einsparpotential wird in einer Schleife über die Elemente von C_K nacheinander jede Komponentenkombination $C_{M_K^p}$ betrachtet. Diese Schleife sorgt für das Verzweigen eines Knotens.
6. Für die Menge der nicht fixierten Kanten $V_{E-E,\Omega}$ wird der Aufwand $a_{\mathbf{w}_\Omega}^{erf}$ ermittelt, der mindestens erforderlich ist, um diese zu fixieren. Der Aufwand berechnet sich auf der Grundlage des Vektors \mathbf{w}_Ω und der normierten Bewertung von $C_{M_K^p}$.
7. Ist der verfügbare Aufwand $a_{\omega,opt} - a_{\mathbf{w}_\Omega}$ kleiner als der erforderliche Aufwand $a_{\mathbf{w}_\Omega}^{erf}$, dann wird die Schleife abgebrochen.
8. Die Verzweigung des aktuellen Knotens wird abgebrochen. Die verbleibenden Komponentenkombinationen in C_K werden nicht mehr berücksichtigt. Weil die Komponentenkombinationen in C_K nach Einsparpotential geordnet entnommen werden, kann für die verbleibende Menge an Kindknoten ausgeschlossen werden, dass sie zu einer optimalen Lösung führen.
9. Der Vektor $\mathbf{w}_{\Omega,Kind}$ wird angelegt. Die Kantenmengen Z_{E-E} , V_{E-E} und S_{E-E} von $\mathbf{w}_{\Omega,Kind}$ berücksichtigen gegenüber denen von \mathbf{w}_Ω die zusätzliche Komponentenkombination $C_{M_K^p}$.
10. Für den Vektor $\mathbf{w}_{\Omega,Kind}$ ist die Menge der Komponentenkombinationen zu bestimmen. Die Komponentenkombinationen aus $C_{M_K^p}$, die für $\mathbf{w}_{\Omega,Kind}$ von Bedeutung sind, dürfen nur aus Kanten bestehen, die in $V_{E-E,Kind}$ enthalten sind. Sie werden in der Menge $C_{K,Kind}$ zusammengefasst.
11. Für den Vektor \mathbf{w}_Ω und die Komponentenkombinationen $C_{K,Kind}$ wird der Algorithmus *verzweige* rekursiv aufgerufen (Tiefensuche).
12. Die letzte Komponentenkombination $C_{M_K^p}$ wird aus C_K entfernt. Die Menge C_K wird mit jedem betrachteten Kind kleiner.

13. Durch den erneuten Aufruf von *holeMinKomb* wird das nächste Element aus der verbleibenden Menge an Komponentenkombinationen C_K ermittelt, für das die normierte Bewertung minimal ist. Die Verzweigung setzt sich fort, bis die Menge C_K leer ist oder ein terminierendes Kriterium greift.
14. Bevor die Auswertung an einem Knoten abgeschlossen ist, wird das aktuelle Prozessmodell ω registriert. Seine Bewertung $a_\omega(\omega)$ wird mit der Schranke $a_{\omega,opt}$ verglichen. Ist die Bewertung besser oder gleich der bisher besten gefundenen Lösung, ersetzt bzw. ergänzt das Prozessmodell die Elemente in der Menge Ω_{opt} . Der Wert $a_{\omega,opt}$ ist entsprechend anzupassen.

4.3.8 Verlauf des Algorithmus

Abbildung 4.24 stellt den Verlauf des Algorithmus 14 dar. Entlang der Abszisse sind die Bewegungen im Baum angetragen. Jeder Schritt von einem Knoten des Entscheidungsbaums zu einem anderen wird als Bewegung aufgefasst. Es werden sowohl die Bewegungen in den Baum hinein als auch die Bewegungen wieder heraus gezählt.

Das obere Diagramm stellt zwei Größen dar, eine als abgestuften Verlauf und die andere schrittweise als Impulsstrich. Der abgestufte Verlauf zeigt die Bewertung $a_{\omega,opt}$. Die Impulsstriche geben an, wie viele Knoten und Kanten in einem Schritt durch Komponentenkombinationen fixiert sind. Sowohl der Verlauf als auch die Impulsstriche sind auf die Anzahl der Knoten und Kanten des zugrunde liegenden Ereignisgraphen normiert.

Das untere Diagramm stellt ebenfalls zwei Größen dar. In der Richtung der positiven Ordinaten ist der verfügbare Aufwand $a_{\mathbf{w}\Omega,i}^{verf}$ in einem Schritt angetragen und in der negativen Richtung der erforderliche Aufwand $a_{\mathbf{w}\Omega,i}^{erf}$. Beide sind auf die Anzahl der Knoten und Kanten des zugrunde liegenden Ereignisgraphen normiert. Ihre Differenz führt zu dem abgestuften Verlauf.

Für die Interpretation der Diagramme ist eine Leserichtung zu beachten. Sie ergibt sich aus der Buchstabenfolge.

- A. Mit der Initialisierung des Entscheidungsbaums wird ein initiales Prozessmodell ω_{init} ermittelt. Dieses stellt eine Zerlegung des betrachteten Graphen in seine minimalen Komponenten dar. Die Bewertung von $\mathbf{w}_{\omega,init}$ führt zu einer ersten oberen Schranke $a_{\omega,opt}$.

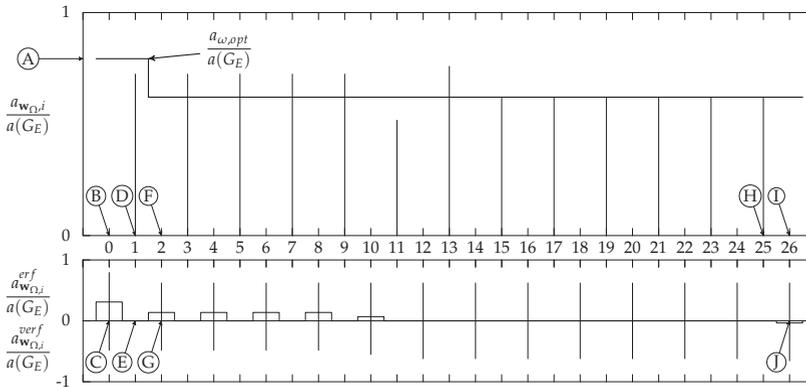


Abbildung 4.24: Verlauf des Algorithmus

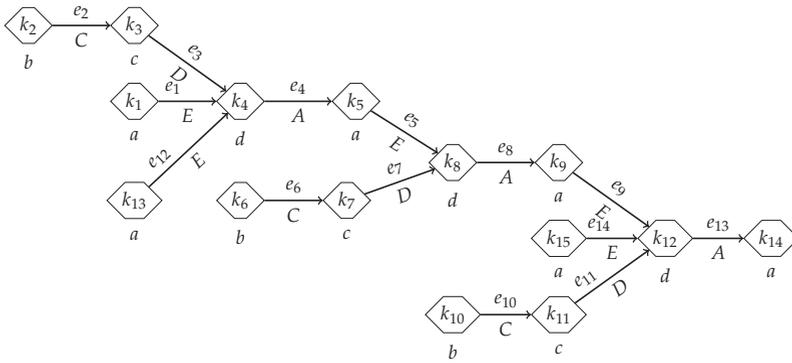
Normiert auf die Anzahl der Knoten und Kanten des zugrunde liegenden Ereignisgraphen ergibt sich ein Wert, der kleiner ist als eins.

- B. Zu diesem Zeitpunkt hat noch keine Bewegung stattgefunden. Im Schritt null ist der Anteil der fixierten Knoten und Kanten null. Der aktuelle Knoten ist der Wurzelknoten.
- C. Für den Wurzelknoten wird aus der Menge der Komponentencombinationen diejenige mit minimaler normierter Bewertung ermittelt. Ist für die ausgewählte Komponentencombination die Differenz aus dem verfügbaren Aufwand $a_{w_{\Omega,0}}^{verf}$ und dem erforderlichen Aufwand $a_{w_{\Omega,0}}^{erf}$ größer als null, dann ist ein Kind anzulegen. Das Anlegen eines Kindes ist eine Bewegung in den Baum hinein. Der aktuelle Schritt wird hochgezählt.
- D. Durch das Einsetzen der ersten Komponentencombination springt der Anteil der fixierten Knoten und Kanten hoch. Für den aktuellen Knoten wird der Vektor $w_{\omega,1}$ bestimmt. Die Menge der Komponentencombinationen wird auf $V_{E-E,1}$ angepasst.
- E. Für den aktuellen Knoten lässt sich keine weitere Komponentencombination ermitteln. Die verbleibende Menge an variablen Kanten $V_{E-E,1}$ reicht nicht aus, um mindestens zwei weitere Komponenten, deren Muster gleich ist, sich aber von denen der bereits eingesetzten Komponenten unterscheidet, einsetzen zu können. Der aktu-

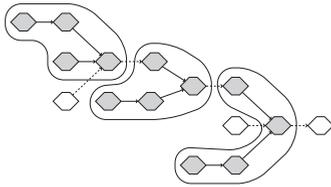
elle Knoten wird nicht weiter verzweigt. Abschließend wird für das Prozessmodell ω_1 geprüft, ob es eine bessere Lösung repräsentiert, als die bisher beste gefundene Lösung. Da der Verlauf von $a_{\omega,opt}$ zwischen dem Schritt eins und Schritt zwei fällt, ist dies der Fall. Seine Bewertung führt zu einem neuen Optimum. Das Prozessmodell ω_1 wird als neue beste Lösung registriert. Es folgt eine Bewegung zurück zu dem Wurzelknoten.

- F. Die Komponentenkombination, die zu dem ersten Kind geführt hat, wird verworfen.
- G. Für den Wurzelknoten wird die nächste Komponentenkombination mit minimaler normierter Bewertung ermittelt. Die Differenz aus dem verfügbaren Aufwand $a_{w\Omega,2}^{verf}$ und dem erforderlichen Aufwand $a_{w\Omega,2}^{erf}$ ist größer als null. Das nächste Kind wird angelegt.
- Dieses Vorgehen setzt sich kontinuierlich fort. In diesem Beispiel ist bereits nach dem ersten Schritt eine beste Lösung ermittelt. Bessere werden nicht gefunden. Zu welchem Zeitpunkt weitere, gleichwertige Lösungen gefunden werden, geht nicht aus dem Diagramm hervor. An dem abgestuften Verlauf der Differenz aus dem verfügbaren Aufwand $a_{w\Omega,i}^{verf}$ und dem erforderlichen Aufwand $a_{w\Omega,i}^{erf}$ ist zu erkennen, dass die Komponentenkombinationen in der Reihenfolge ihres Einsparpotentials eingesetzt werden. Die Differenz wird kontinuierlich kleiner. Zu keinem Zeitpunkt wird mehr als eine Komponentenkombination eingesetzt. Dies ist daran zu erkennen, dass der Anteil der fixierten Knoten und Kanten in aufeinander folgenden Schritten nicht ansteigt. Jede Bewegung in den Baum hinein wird durch eine Bewegung aus dem Baum heraus unterbrochen. Jeder Impulsstrich steht somit für ein Kind des Wurzelknotens. Insgesamt werden 13 Kinder entwickelt.
- H. In Schritt 25 ist das 13. Kinder der aktuelle Knoten. Es existiert keine weitere Komponentenkombination. Der aktuelle Knoten wird terminiert.
- I. Die Suche fällt zurück auf den Wurzelknoten.
- J. Für den Wurzelknoten wird die nächste Komponentenkombination mit minimaler normierter Bewertung ermittelt. Die Differenz aus dem verfügbaren Aufwand $a_{w\Omega,26}^{verf}$ und dem erforderlichen Aufwand

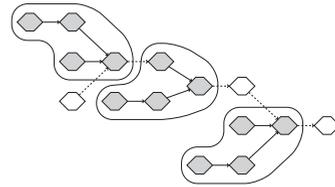
$a_{\mathbf{w}_{\Omega,26}}^{erf}$ ist kleiner als null. Für die ausgewählte Komponentenkombination muss kein Kind angelegt werden. Alle verbleibenden, noch nicht eingesetzten Komponentenkombinationen - 14 von 27 - würden ebenfalls einen Wert liefern, der kleiner ist als null. Entsprechend braucht auch für diese kein Kindknoten angelegt zu werden. Der aktuelle Knoten kann terminiert werden. Weil der aktuelle Knoten der Wurzelknoten ist, ist die Suche abgeschlossen. Abbildung 4.25 zeigt die gefundenen Zerlegungen.



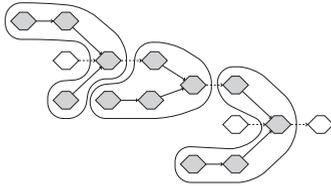
(a) Beispielprozess



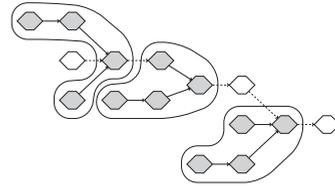
(b) Prozessmodell $\omega_{opt,1}$



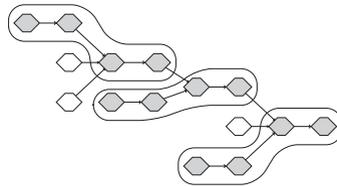
(c) Prozessmodell $\omega_{opt,2}$



(d) Prozessmodell $\omega_{opt,3}$



(e) Prozessmodell $\omega_{opt,4}$



(f) Prozessmodell $\omega_{opt,5}$

Abbildung 4.25: Optimale Lösungen

Kapitel 5

Auswertung exemplarischer Beispiele

Für die Modellierung von Bauabläufen auf der Grundlage des in Kapitel 3 beschriebenen Ansatzes entscheidet die Zerlegung des Projekts in Leistungsgegenstände über den entstehenden Spezifikationsaufwand. Jedem Leistungsgegenstand ist ein Leistungsprozess zuzuordnen. Fasst man die Zustandsgraphen der einzelnen Leistungsprozesse in einem Graphen zusammen, so entsteht ein Gebilde aus unabhängigen Komponenten - eine für jeden Leistungsgegenstand. Durch die Spezifikation von Voraussetzungen für Vorgänge nach Abschnitt 3.4.3 lassen sich diejenigen Abhängigkeiten ermitteln, die die unabhängigen Komponenten miteinander in Beziehung setzen. Spezifikationsaufwand ergibt sich aus dem Anlegen von Vorlagen für Leistungsgegenstände und Leistungsprozesse, dem Anlegen von projektspezifischen Ausprägungen und der Spezifikation von Voraussetzungen. Je nach Einteilung in Leistungsgegenstände kann der resultierende Spezifikationsaufwand variieren.

Durch das Erkennen von Mustern in Zustandsgraphen lassen sich für ein Projekt rückwirkend Zerlegungen in Leistungsprozesse ermitteln, deren Muster als Leistungsprozessvorlagen interpretiert werden können. Zerlegungen, für die der ermittelte Spezifikationsaufwand minimal ist, werden als optimal betrachtet. Die im Rahmen einer optimalen Zerlegung ermittelten Muster bilden eine Basis an optimalen Leistungsprozessvorlagen für ein konkretes Projekt.

In diesem Kapitel wird zum einen an exemplarischen Beispielen auf das Verfahren der Mustererkennung eingegangen und zum anderen eine Gegenüberstellung der Leistungsprozessvorlagen vorgenommen. Auf der einen Seite stehen die Leistungsprozessvorlagen, die für die Modellierung des zugrunde liegenden Prozessmodells eingesetzt wurden und auf der

anderen Seite stehen die Leistungsprozessvorlagen, die als Ergebnis der Mustererkennung berechnet werden.

In Abschnitt 5.1 *Beispiel 1: Herstellung von Wänden* wird ein Beispielprozess eingeführt, der der Gegenüberstellung zugrunde liegt. Die Betrachtungen erfolgen auf der Grundlage von Ereignisgraphen. Es wird auf die folgenden Schritte eingegangen: (1) Zerlegen des Projekts in Leistungsgegenstände, (2) Zuordnen von Leistungsprozessvorlagen und (3) Spezifikation von Voraussetzung.

5.1 Beispiel 1: Herstellung von Wänden

Betrachtet wird die Herstellung von vier Mauerwerkswänden, an denen unterschiedliche Arbeiten verrichtet werden. Abhängig von den Räumen, die diese Wände begrenzen, seien zusätzlich Elektro- und/oder Sanitärinstallationen vorzunehmen. Für die vier Wände existieren acht Wandoberflächen, an denen zum Teil unterschiedliche Maßnahmen vorgenommen werden. Alle Wandoberflächen seien verputzt. Es fallen somit Tätigkeiten an, die das Mauern, Verputzen, Verlegen von Leitungen, Anschließen von Apparaten, Anschließen von Armaturen u.s.w. betreffen.

Zwei Wände seien gleichartig. Auf beiden Oberflächen werden sowohl Elektro- als auch Sanitärinstallationen vorgenommen. Die dritte Wand erhalte auf einer ihrer Oberflächen eine Elektroinstallation und die vierte Wand erhalte auf einer ihrer Oberflächen eine Sanitärinstallation. Das betrachtete Projekt wird in die folgenden 10 Leistungsgegenstände eingeteilt:

Seite A	Wand	Seite B
$Seite_{A,1}$	– $Wand_1$ –	$Seite_{B,1}$
$Seite_{A,2}$	– $Wand_2$ –	$Seite_{B,2}$
$Seite_{A,3}$	– $Wand_3$	
	$Wand_4$ –	$Seite_{B,4}$

Die unterschiedlichen Seiten einer Wand werden mit A und B bezeichnet. Um einen Bezug zwischen den Leistungsgegenständen $Wand$ und $Seite$ herzustellen, erhält eine $Seite$ zusätzlich den Index der zugehörigen $Wand$.

Die Einteilung geht davon aus, dass das Mauern und das Verputzen einer Wand unabhängig von den Installationen an ihrer Oberfläche betrachtet wird. Die Installationen auf den Wandoberflächen werden als eigen-

ständige Leistungsgegenstände betrachtet. Für jede Wand wird ein Leistungsgegenstand $Wand_i$ angelegt. Von den acht Wandoberflächen brauchen zwei nicht weiter berücksichtigt zu werden. Es fallen keine weiteren Maßnahmen auf ihren Oberflächen an. Für diejenigen Wandoberflächen, auf denen eine Installation vorgenommen wird, wird ein Leistungsgegenstand $Seite_{x,i}$ angelegt.

Jedem Leistungsgegenstand $Wand_i$ wird ein Leistungsprozess zugeordnet, der aus der Leistungsprozessvorlage *verputzte Mauerwerkswand* abgeleitet ist. Hierbei handelt es sich um eine Prozessvorlage, die das Mauern der Wand beschreibt und das Verputzen auf beiden Seiten (siehe Abbildung 5.1 *verputzte Mauerwerkswand*). Von den verbleibenden sechs Wandoberflächen unterscheiden sich drei in ihrer Herstellungsweise. Für die sich unterscheidenden Herstellungsweisen werden entsprechende Leistungsprozessvorlagen entwickelt. Abbildung 5.1 zeigt die Leistungsprozessvorlagen als Ereignisgraphen. *Elektroinstallation* beschreibt das

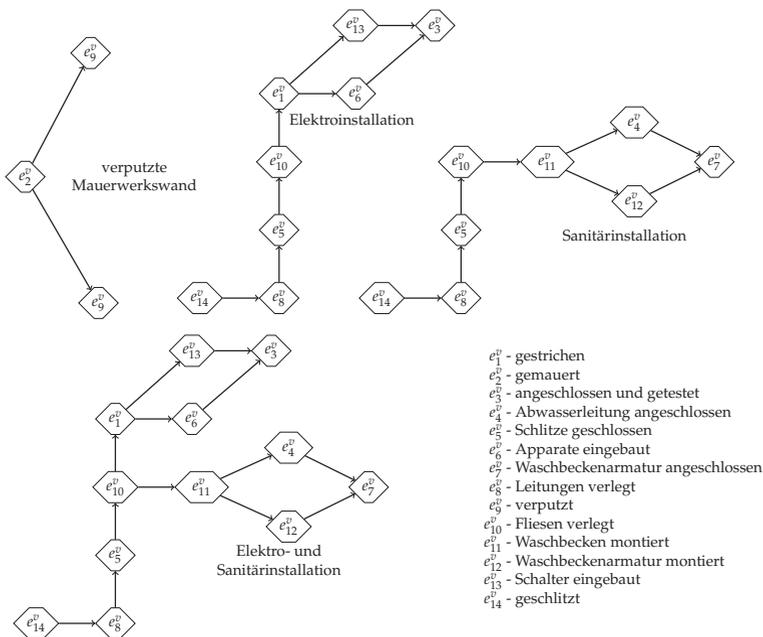


Abbildung 5.1: Leistungsprozessvorlagen

Verlegen von Leitungen und Tätigkeiten im Rahmen einer Elektroinstallation. *Sanitärinstallation* beschreibt das Verlegen von Leitungen und Tätigkeiten im Rahmen einer Sanitärinstallation. Die dritte Leistungsvorlage *Elektro- und Sanitärinstallation* ist eine Kombination aus den beiden vorherigen. Folgende Zuordnung wird für die Leistungsgegenstände vorgenommen:

Leistungsgegenstand	–	Leistungsprozessvorlage
<i>Wand₁</i>	–	<i>verputzte Mauerwerkswand</i>
<i>Wand₂</i>	–	<i>verputzte Mauerwerkswand</i>
<i>Wand₃</i>	–	<i>verputzte Mauerwerkswand</i>
<i>Wand₄</i>	–	<i>verputzte Mauerwerkswand</i>
<i>Seite_{A,1}</i>	–	<i>Elektroinstallation</i>
<i>Seite_{B,1}</i>	–	<i>Elektro – und Sanitärinstallation</i>
<i>Seite_{A,2}</i>	–	<i>Elektroinstallation</i>
<i>Seite_{B,2}</i>	–	<i>Elektro – und Sanitärinstallation</i>
<i>Seite_{A,3}</i>	–	<i>Elektroinstallation</i>
<i>Seite_{B,4}</i>	–	<i>Sanitärinstallation</i>

Auf der Grundlage der Zuordnung der Leistungsprozessvorlagen zu den Leistungsgegenständen wird ein Prozessmodell generiert. Abbildung 5.2 zeigt das Resultat. Für jeden Leistungsgegenstand existiert ein Leistungsprozess. Diese stellen unabhängige Komponenten dar.

Alle dargestellten Ereignisse sind das Ergebnis eines Vorgangs. Die Vorgänge sind nicht mit dargestellt. Die Spezifikation von Voraussetzungen für Vorgänge nach Abschnitt 3.4.3 liefert diejenigen Kanten, die die unterschiedlichen Komponenten miteinander in Beziehung setzen. Bei der gegebenen Einteilung in Leistungsgegenstände sind für folgende Vorgänge Voraussetzungen anzugeben: *schlitzen*, *verputzen*, *Fliesen verlegen* und *streichen*. Das Schlitzen einer Wand, um Leitungen verlegen zu können, setzt ihre Existenz voraus. Sie muß den Zustand gemauert erreicht haben. Bevor die Wand verputzt werden kann, sind die Schlitze zu schließen. Daraufhin folgt das Fliesen bzw. Streichen der Wand. Eine detaillierte Aufstellung der zusätzlich zu spezifizierenden Voraussetzungen ist in Abbildung 5.3 dargestellt. Die sich aus den Voraussetzungen ergebenden Kanten sind als gestrichelte Kanten in den Gesamtprozess in Abbildung

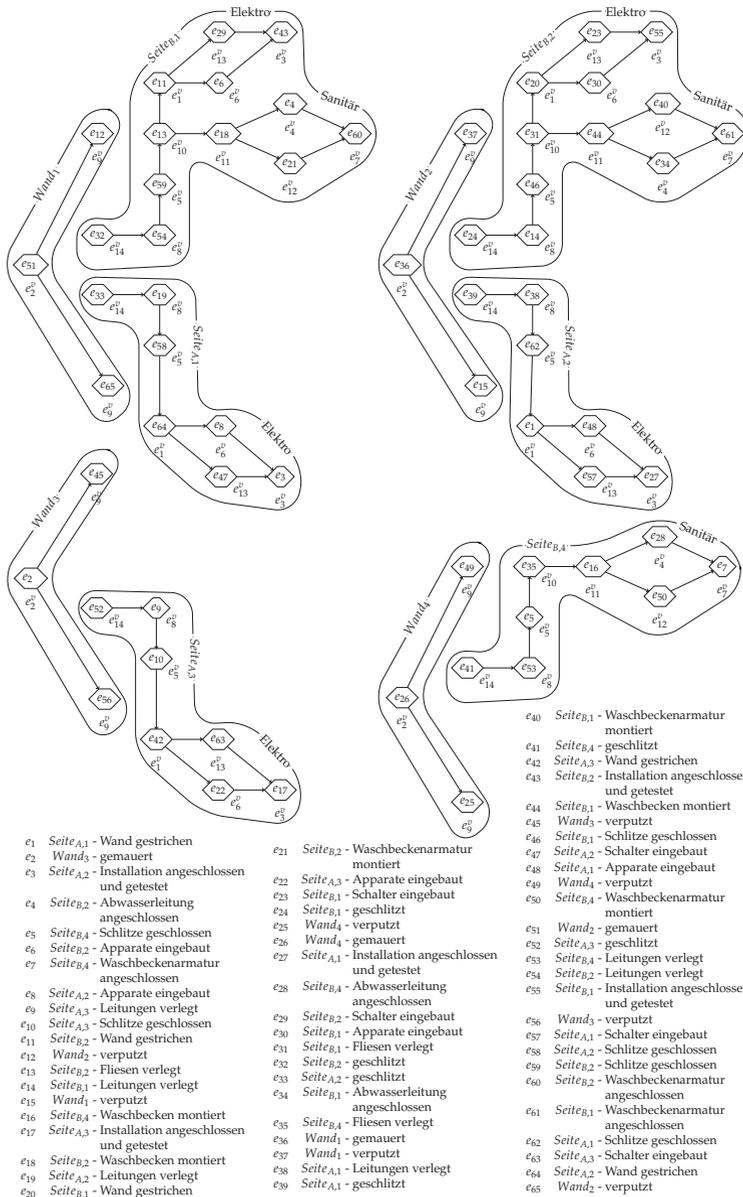


Abbildung 5.2: Prozessmodell - Generierter Gesamtprozess ohne Voraussetzungen

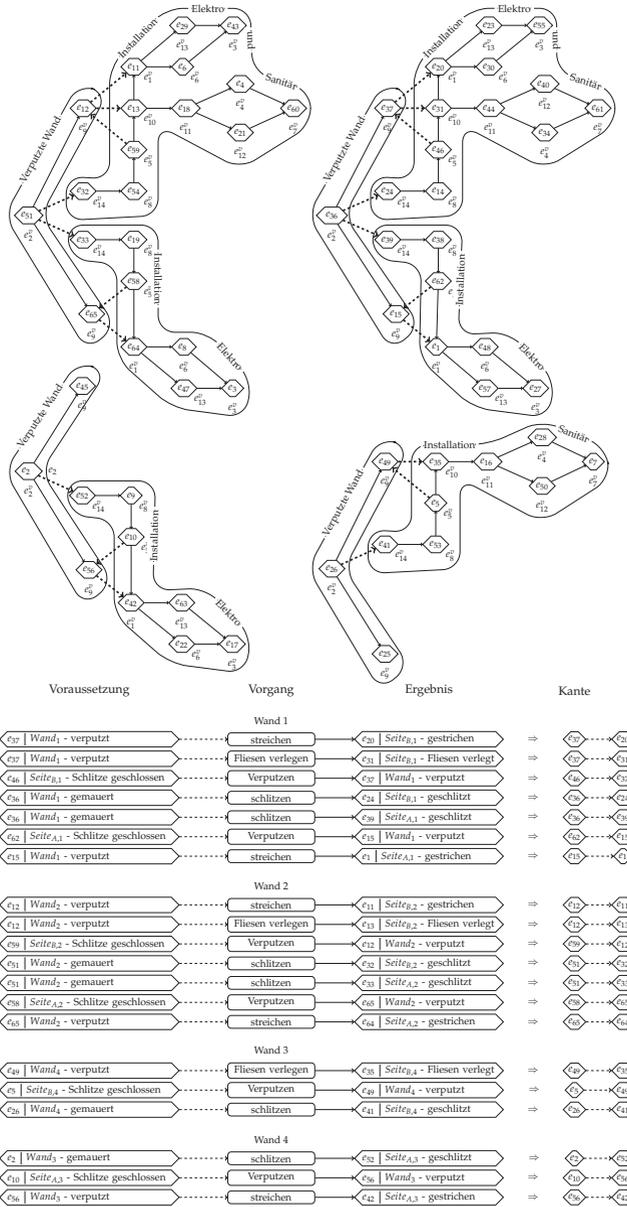


Abbildung 5.3: Prozessmodell - Spezifizierter Gesamtprozess

5.3 eingetragen. Der dargestellte Ereignisgraph bildet den Ausgangsgraphen für die Mustererkennung.

Der Graph besteht aus 65 Knoten und 83 Kanten. Ohne Berücksichtigung der wiederkehrenden Teilprozesse ist die Bewertung für den Aufwand des Graphen 148.

$$\begin{aligned} \text{Gleichung 4.11} \quad a(G_E) &= |M_E| + |R_{E-E}| \\ &= 65 + 83 = 148 \end{aligned} \quad (5.1)$$

Unter Berücksichtigung der wiederkehrenden Teilprozesse berechnet sich der Aufwand für dieses Modell aus dem Aufwand der einzelnen Leistungsprozessvorlagen, erweitert um die Anzahl der Komponenten, plus der Anzahl der zusätzlichen Kanten. Der Aufwand für die Leistungsprozessvorlagen berechnet sich nach Gleichung 4.11:

Leistungsprozess- vorlage	Knoten $ M_E $	Kanten $ R_{E-E} $	Bewertung $a(G_E)$
<i>verputzte Mauerwerkswand</i>	3	2	5
<i>Elektro- und Sanitärinstallation</i>	7	7	14
<i>Elektroinstallation</i>	8	8	16
<i>Sanitärinstallation</i>	12	13	25
$\sum_{x \in M_R} a(x)$			60

Für den generierten Gesamtprozess ohne Voraussetzungen nach Abbildung 5.2 berechnet sich der Aufwand nach Gleichung 4.12:

$$\begin{aligned} \text{Gleichung 4.12} \quad a_Z(Z_E) &= |M_K| + \sum_{x \in M_R} a(x) \\ &= 10 + 60 = 70 \end{aligned} \quad (5.2)$$

Unter Berücksichtigung der zu spezifizierenden Kanten ergibt sich nach Gleichung 4.14 der Gesamtaufwand für das Prozessmodell aus Abbildung 5.3.

$$\begin{aligned} \text{Gleichung 4.14} \quad a_\omega(\omega) &= a_Z(Z_E) + |S_{E-E}| \\ &= 70 + 20 = 90 \end{aligned} \quad (5.3)$$

Der Gesamtaufwand für die Spezifikation des Prozessmodells unter Berücksichtigung der wiederkehrenden Teilprozesse wird mit 90 bewertet. Ziel der Mustererkennung ist es, diejenige Basis oder Basen an Leistungsvorlagen zu ermitteln, auf deren Grundlage derselbe Prozess darstellbar ist, dessen oder deren Aufwandsbewertung minimal ist.

5.2 Mustererkennung für das Beispiel

Die Mustererkennung erfolgt auf der Grundlage des Branch-and-Bound-Verfahrens nach Abschnitt 4.3.2. In den folgenden Abschnitten wird auf das Initialisieren des Entscheidungsbaums eingegangen, auf den Verlauf der Suche und auf das Ergebnis der Suche.

5.2.1 Initialisieren

Das Initialisieren des Verfahrens umfasst die Schritte: (1) Zerlegen des Ereignisgraphen in seine minimalen Komponenten, (2) Ermitteln der Komponentenkombinationen und (3) Anlegen des Entscheidungsbaums.

Zu Punkt 1: Ausgehend von dem Gesamtprozess werden diejenigen Komponenten bestimmt, die nicht weiter zerlegt werden müssen. Abbildung 5.4 zeigt die Zerlegung des Gesamtprozesses in seine minimalen Komponenten. Die Komponenten sind entsprechend ihres Musters in

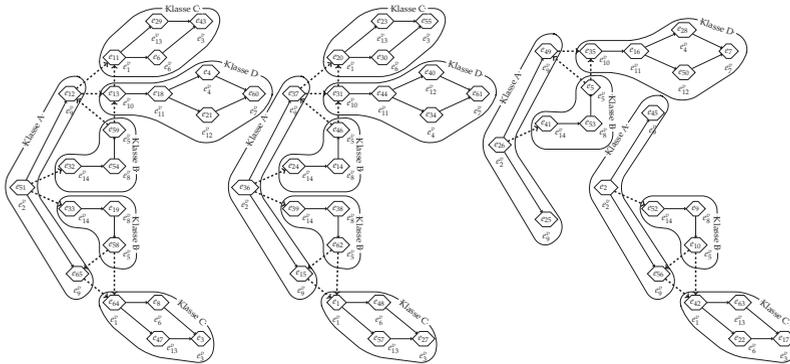


Abbildung 5.4: Prozessmodell - Minimale Komponenten

Äquivalenzklassen eingeteilt. Es ergeben sich vier verschiedene Muster:

Klasse A, Klasse B, Klasse C und *Klasse D*. Die Zugehörigkeit einer Komponente zu einer Äquivalenzklasse ist durch die Beschriftung in ihrer Umrandung angegeben. Die minimalen Komponenten stellen eine Zerlegung des Prozesses in 18 Komponenten dar.

Äquivalenzklasse	Bewertung $a(G_E)$	Anzahl der Komponenten
<i>Klasse A</i>	5	4
<i>Klasse B</i>	5	6
<i>Klasse C</i>	8	5
<i>Klasse D</i>	10	3
Σ	28	18

Aus der Ermittlung der minimalen Komponenten ergibt sich, dass von den 83 Kanten insgesamt nur 28 zu variieren sind. Im ungünstigsten Fall sind somit $2^{28} = 268.435.456$ Knoten im Entscheidungsbaum zu entwickeln. Dieser Aufwand lässt sich durch die Betrachtung der Komponentenkombinationen reduzieren.

Zu Punkt 2: Der Bestimmung der Komponentenkombinationen liegen die 18 minimalen Komponenten und die 28 zu variierenden Kanten zugrunde. Durch das Entwickeln von Vorgänger- und Nachfolgerbäumen für jede der 18 Komponenten nach Abschnitt 4.3.4 lassen sich insgesamt 1.368 verschiedene Komponenten ermitteln, die als Kandidaten für die Komponentenkombinationen in Frage kommen. Diese Komponenten lassen sich anhand ihres Musters in 652 verschiedene Äquivalenzklassen einteilen. Erzeugt man für jede Äquivalenzklasse die Menge der Kombinationen, die mindestens zwei Komponenten enthalten, so erhält man für dieses Beispiel 901 unterschiedliche Komponentenkombinationen.

Anzahl der minimalen Komponenten:	18
Anzahl der zu variierenden Kanten:	28
Anzahl der relevanten Komponenten:	1368
Anzahl der Äquivalenzklassen:	652
Anzahl der relevanten Komponentenkombinationen:	901

Anstatt im Entscheidungsbaum die oben genannten 28 Kanten einzeln zu variieren, werden mit jedem Schritt Teilmengen dieser Kanten variiert. Welche Teilmengen dies mit jedem Schritt sind, ergibt sich implizit aus der Menge an Komponenten, die insgesamt an einem Knoten des Entscheidungsbaums infolge der eingesetzten Komponentenkombinationen

fixiert sind. Sind einzelne Kanten in einer fixierten Komponente enthalten, gehören sie zu der Teilmenge dazu. Grenzen sie an eine Komponente an, dann sind sie definitiv nicht in der Teilmenge enthalten. Gehören sie zu der Menge der variablen Kanten, dann erfolgt die Festlegung erst mit dem Erreichen eines Baumknotens, der in größerer Tiefe liegt.

Zu Punkt 3: Als erste beste Lösung wird das Prozessmodell, bestehend aus den minimalen Komponenten (siehe Abbildung 5.4), in Ω_{opt} registriert. Der zugehörige Aufwand $a_\omega(\omega) = 74$ wird als obere Schranke vermerkt.

$$\begin{aligned}
 a_Z(Z_E) &= |M_K| + \sum_{x \in M_R} a(x) \\
 &= 18 + 28 = 46 \\
 a_\omega(\omega) &= a_Z(Z_E) + |S_{E-E}| \\
 &= 46 + 28 = 74
 \end{aligned} \tag{5.4}$$

5.2.2 Verlauf des Verfahrens

Nach dem Initialisieren des Entscheidungsbaums erfolgt die Suche. Der Verlauf des Verfahrens soll anhand des Diagramms in Abbildung 5.5 erläutert werden. Die Ordinaten beziehen sich auf normierte Aufwände und die Abszissen beziehen sich auf die Bewegungen im Baum. Alle Aufwände sind auf die Anzahl der Knoten und Kanten des zugrunde liegenden Ereignisgraphen bezogen (Gleichung 5.1). Als Bewegung gilt das Verzweigen - Bewegung in den Baum hinein - und das Terminieren - Bewegung aus dem Baum heraus. Bei insgesamt 1.041 dargestellten Bewegungen sind 521 Knoten im Entscheidungsbaum traversiert worden.

Das obere Diagramm stellt den Anteil der fixierten Knoten und Kanten für den aktuellen Schritt dar. Infolge einer Verzweigung ergibt sich ein Anstieg für die Ausschläge der Impulsstriche. Infolge des Terminierens ergibt sich ein Verkürzen der Impulsstriche. Zusätzlich ist der Verlauf der oberen Schranke angegeben. Zu Beginn liegt der Wert bei $74/148 = 0,5$. Der Wert 74 ergibt sich aus Gleichung 5.4 und 148 aus Gleichung 5.1.

Das untere Diagramm stellt den verfügbaren Aufwand¹ für eine weitere Verzweigung dem erforderlichen Aufwand² gegenüber. Ist die Differenz positiv, dann kann weiter verzweigt werden. Ist die Differenz negativ,

¹ Betrag in Richtung der positiven Ordinatenachse

² Betrag in Richtung der negativen Ordinatenachse

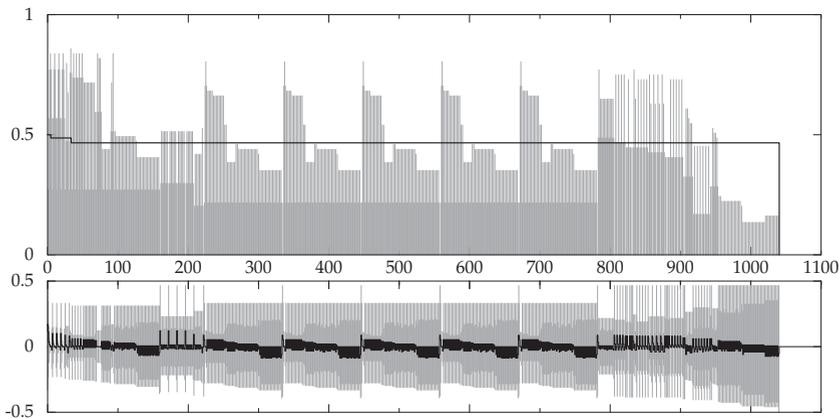


Abbildung 5.5: Verlauf der Suche

dann wird der aktuelle Knoten terminiert. Eine Erläuterung zur Lesart des Diagramms liegt in Abschnitt 4.3.8 vor.

Eine schrittweise Interpretation des Verlaufs ist aufgrund der Dichte der Informationen nicht zweckmäßig. Stattdessen werden Bereiche betrachtet, an denen das Verhalten des Verfahrens erläutert wird. Markante Intervallgrenzen ergeben sich an den Stellen, an denen die Ordinaten null sind. Dies ist immer dann der Fall, wenn der Wurzelknoten des Entscheidungsbaums passiert wird. Mit anderen Worten, ein Kind der Wurzel wurde vollständig abgearbeitet und terminiert, und die Suche wird für das nächste Kind fortgesetzt.

Intervall 0 - 160: Die Schritte 0 - 160 beziehen sich auf den Zweig des Entscheidungsbaums, der durch das erste Kind des Wurzelknotens definiert ist. Seine Traversierung erfordert 160 Bewegungen. Hierbei werden 80 unterschiedliche Baumknoten untersucht. Allen Pfaden, die zu den unterschiedlichen Knoten führen, ist die erste Kante gemein. Diese Kante steht für die Komponentenkombination, die zu dem ersten Kindknoten des Wurzelknotens führt. Der Anteil an Knoten und Kanten, der durch diese Komponentenkombination im Ereignisgraphen fixiert wird, ist für alle 160 Schritte gleich. Daraus folgt die konstante Länge, die jeder Impulsstrich in diesem Bereich mindestens hat. Erst in Schritt 161 wird dieser Verlauf unterbrochen. Die Ordinate fällt auf null.

Zu Beginn des Bereichs ist ein Verzweigen bis in die Tiefe vier zu

sehen. Mit zunehmender Anzahl der Schritte sind zwei Trends erkennbar: (1. oberes Diagramm) die Anzahl der Verzweigungen nimmt ab und (2. unteres Diagramm) der Mittelwert der Differenz aus verfügbarem Aufwand und erforderlichem Aufwand wird kontinuierlich kleiner. Dieses stellt sich anschaulich im unteren Diagramm durch die dunkle Fläche dar, deren Schwerpunkt immer weiter nach unten wandert.

Diese Trends werden erst durch die Bewegung zurück zur Wurzel unterbrochen. Für die weiteren Kinder der Wurzel wiederholt sich das beschriebene Szenario, am Anfang das Verzweigen, dann das Auslaufen. Der Begriff *Auslaufen* wird hier benutzt, um den Teil der Untersuchung eines Knotens k zu beschreiben, bei dem weiterhin Kinder von k entwickelt werden, für die aber keine Verzweigung mehr erfolgt. Das Auslaufen setzt sich solange fort, bis es keine weitere Komponentenkombination mehr für k gibt, deren erforderlicher Aufwand kleiner ist als der verfügbare Aufwand.

Das beobachtete Verhalten lässt sich wie folgt erklären: Zu Beginn stehen einem Baumknoten k alle ihm zugeordneten Komponentenkombinationen zur Verfügung. Nachdem das erste Kind von k abgearbeitet ist, wird die Komponentenkombination, die zu dem Kind geführt hat, für alle nachfolgenden Verzweigungen, die von k ausgehen, nicht mehr betrachtet. Je mehr Kinder von k terminiert sind, desto weniger Komponentenkombinationen verbleiben als Kandidaten für Nachfolger von k . Die Zahl der Verzweigungen nimmt ab, weil einerseits immer weniger Komponentenkombinationen zur Verfügung stehen und andererseits die verbleibenden Kandidaten ein immer geringeres Einsparpotential haben. Das untere Diagramm zeigt, dass immer in den Auslaufphasen der erforderliche Aufwand vom Trend ansteigt, bis der verfügbare Aufwand nicht mehr ausreicht, um eine der verbleibenden Komponentenkombinationen einsetzen zu können. Ist dieser Punkt erreicht, dann wird der aktuelle Knoten k terminiert. Alle bis zu diesem Zeitpunkt noch nicht untersuchten Komponentenkombinationen brauchen für den aktuellen Knoten nicht weiter untersucht zu werden.

Intervall 161 - 222: In diesem Intervall werden fünf weitere Kinder des Wurzelknotens nach optimalen Lösungen durchsucht. Das hierfür nur so wenig Bewegungen erforderlich sind, lässt sich darauf zurückführen, dass die eingesetzten Komponentenkombinationen entlang der Pfade zu den Kindknoten keinen Raum mehr lassen, um weitere Komponentenkombinationen überschneidungsfrei einzusetzen.

Vergleicht man den Verlauf des ersten Kindes der Wurzel (Intervall 1 - 160) mit dem Verlauf des zweiten Kindes (Intervall 162 - 173), so fällt auf, dass der Anteil an fixierten Knoten und Kanten aus der ersten Kompo-

tenkombination angestiegen ist. Dies ist nicht ungewöhnlich. Fest steht, dass die Komponentenkombination, die zu dem ersten Kind der Wurzel geführt hat, ein größeres Einsparpotential hat als die Komponentenkombination, die zu dem zweiten Kind geführt hat. Aus dieser Ordnung lässt sich aber nicht auf den Umfang der betroffenen Knoten und Kanten schließen. In diesem Fall ist die Summe der Knoten und Kanten der ersten Komponentenkombination kleiner als die Summe der Knoten und Kanten der zweiten Komponentenkombination. In den meisten Fällen nimmt der Anteil der fixierten Knoten und Kanten mit abnehmendem Einsparpotential ab. Es lassen sich aber auch Beispiele konstruieren, für die dies nicht der Fall ist.

Intervall 223 - 782: Dieses Intervall zeigt fünf sehr ähnliche Verläufe. Die Phase des Verzweigens zu Beginn ist sehr kurz und die Phase des Auslaufens sehr ausgeprägt. Es betrifft das 7. bis 11. Kind der Wurzel. Diesen Kindern liegen Komponentenkombinationen zugrunde, deren Komponenten der Klasse *C* nach Abbildung 5.4 angehören. Insgesamt umfasst die Klasse *C* fünf Komponenten. Diese Komponenten seien mit 1, 2, 3, 4 und 5 indiziert. Bildet man für diese Äquivalenzklasse die Menge der Komponentenkombinationen, dann entstehen genau fünf Kombinationen, die vier Komponenten enthalten: $\{1, 2, 3, 4\}$, $\{1, 2, 3, 5\}$, $\{1, 2, 4, 5\}$, $\{1, 3, 4, 5\}$ und $\{2, 3, 4, 5\}$. Diese Komponentenkombinationen sind den Kanten zugeordnet, die zu den fünf betrachteten Kindern der Wurzel führen.

Wird ein Kind entwickelt, wird für dieses Kind die Menge der vom Vater übertragenen Komponentenkombinationen angepasst. Die Menge der Komponentenkombinationen eines Kindes ist immer eine echte Teilmenge der Menge der Komponentenkombinationen seines Vaters. In der Menge der Komponentenkombinationen des Kindes verbleiben diejenigen, die sich überschneidungsfrei in den verbleibenden, nicht fixierten Graphen einsetzen lassen und für die das Muster der Komponenten von denen der bereits eingesetzten Komponentenkombinationen unterschiedlich ist.

Für die betrachteten fünf Kinder der Wurzel stellt sich heraus, dass die Menge ihrer Komponentenkombinationen identisch ist. Weil diese Menge identisch ist, ist auch der Verlauf der Suche, der sich aus dieser Menge ergibt, gleich. Aus dem Verlauf der Suche für eines der Kinder könnte auf den Verlauf der Suche der anderen Kinder geschlossen werden. Dieser Aspekt ist zur Zeit nicht Teil des Verfahrens. Hierin liegt weiteres Potential, um die Komplexität weiter zu reduzieren. Für das Beispiel würde sich allein für diese fünf Knoten eine Einsparung von mindestens $782 - 335 = 447$ Schritten ergeben.

Intervall 783 - 1041: Inzwischen sind 11 Kinder der Wurzel vollständig untersucht. Für die weitere Suche brauchen die ihnen zugeordneten Kom-

ponentenkombinationen nicht mehr berücksichtigt zu werden. Mit jedem untersuchten Kindknoten nimmt die Anzahl der zu berücksichtigenden Komponentenkombinationen ab. Diejenigen mit hohem Einsparpotential werden zuerst abgearbeitet. Hieraus folgt, dass die Anzahl der Verzweigungen mit dem Fortschritt der Suche abnimmt. Im Intervall von 0 - 783 werden elf Kinder der Wurzel untersucht. In dem Intervall von 784 - 1041, das nur 257 Schritte umfasst, werden 83 Kinder untersucht. Ab dem Schritt 955 läuft die Suche aus.

Fazit: In der Summe sind für die Wurzel 94 Kinder von 901 Kandidaten entwickelt worden. Die Suche ergab genau ein optimales Prozessmodell. Dieses wurde bereits in Schritt 34 identifiziert. An dieser Stelle liegt der letzte Sprung im Verlauf des optimalen Aufwands. Die weiteren Bewegungen waren erforderlich, weil auf der Grundlage der formalen Kriterien nicht schon früher ausgeschlossen werden konnte, dass der verbleibende Lösungsraum nicht doch noch eine bessere Lösung enthält. Dieses Kriterium wurde erst in Schritt 1041 erreicht.

Mit jedem Schritt in den Entscheidungsbaum hinein nimmt die Anzahl der Komponentenkombinationen, die als Kandidaten für das weitere Verzweigen zur Verfügung stehen, stark ab. Es entsteht ein Baum, der sich stark in der Breite ausdehnt, auf jedem Pfad aber nur eine geringe Tiefe erreicht. Als stärkster Mechanismus, um den Suchaufwand zu begrenzen, stellt sich die Ordnung der Komponentenkombinationen nach ihrem Einsparpotential heraus. Sie erlaubt es, ganze Mengen an Kindkandidaten von der Suche auszuschließen, ohne diese individuell abschätzen oder entwickeln zu müssen.

5.2.3 Ergebnis der Suche

Das Ergebnis der Suche ist die Menge der Prozessmodelle Ω_{opt} . Für das untersuchte Beispiel enthält diese Menge genau ein Element. Im Folgenden wird dieses Element mit ω_{opt} bezeichnet. Für das Prozessmodell ω_{opt} gilt: Es existiert kein anderes Prozessmodell $\omega \in \Omega$, für das die Bewertung $a_\omega(\omega)$ nach Gleichung 4.14 kleiner oder gleich $a_\omega(\omega_{opt})$ ist. Die Bewertung $a_\omega(\omega)$ ist ein Maß für den Spezifikationsaufwand. Das Prozessmodell ω_{opt} stellt somit genau die Zerlegung dar, für die der Spezifikationsaufwand minimal ist.

Abbildung 5.6 stellt drei Prozessmodelle dar. Teilbild 5.6(a) zeigt das Prozessmodell ω_{spez} , das auf der Grundlage der in Kapitel 3 beschriebenen Methodik die Herstellung von vier Mauerwerkswänden beschreibt. Das Teilbild 5.6(a) entspricht der Abbildung 5.3. Dem Prozessmodell lie-

gen Leistungsprozessvorlagen zugrunde, die aus der Anschauung heraus entwickelt wurden. Teilbild 5.6(b) zeigt das Prozessmodell ω_{min} . Das Teilbild ist mit der Abbildung 5.4 identisch. Ihm liegt die Zerlegung des Prozesses, der aus ω_{spez} folgt, in seine minimalen Komponenten zugrunde. Das Prozessmodell ω_{min} markiert den Ausgangspunkt der Suche im Entscheidungsbaum. Das Ergebnis der Suche liefert das Prozessmodell ω_{opt} . Dieses ist in Teilbild 5.6(c) dargestellt. Alle drei Prozessmodelle sind in einer Abbildung zusammengestellt, um sie besser vergleichen zu können.

Das Prozessmodell ω_{opt} wurde im Schritt 34 in der Tiefe *drei* des Entscheidungsbaums identifiziert. Aus der Tiefe geht die Anzahl der eingesetzten Komponentenkombinationen hervor und ebenso die Anzahl der unterschiedlichen Muster. Durch das Einsetzen der Komponentenkombinationen werden Knoten und Kanten fixiert. Durch das Einsetzen der drei Komponentenkombinationen, die zu ω_{opt} geführt haben, sind nicht alle Knoten und Kanten fixiert. In Teilbild 5.6(c) sind diejenigen Knoten und Kanten, die nicht fixiert sind, gestrichelt umrandet. Sie werden als Rest bezeichnet. Das Prozessmodell ω_{opt} teilt den Prozess in 11 Komponenten ein und in den Rest. Ihnen liegen drei verschiedene Muster zugrunde. Der Rest wird hierbei nicht berücksichtigt.

Äquivalenzklasse	Bewertung $a(G_E)$	Anzahl der Komponenten
<i>Verputzte Wand mit Sanitär</i>	24	3
<i>Leitungen</i>	5	3
<i>Elektro</i>	8	5
Σ	37	11

Die Namen der Klassen sind aus der Anschauung heraus gewählt. Es liegt kein Formalismus vor, der aus der Qualität der enthaltenen Ereignisse einen Bezeichner generiert.

In der nachfolgenden Übersicht sind die Aufwände der drei Prozess-

modelle gegenübergestellt.

		ω_{spez}	ω_{min}	ω_{opt}
Aufwand für die Leistungsprozessvorlagen	$\sum_{x \in M_R} a(x)$	60	28	37
Anzahl der Komponenten	$ M_K $	10	18	11
Aufwand für die Zerlegung	$a_Z(Z_E)$	70	46	48
Rest	$a(G_E)$	0	0	5
Anzahl der zusätzlichen Kanten	$ S_{E-E} $	20	28	16
Gesamtaufwand	$a_\omega(\omega)$	90	74	69

Der Aufwand für den Rest ergibt sich aus der Anzahl der Knoten und Kanten. Die Anzahl der Komponenten, die sich aus dem Rest ergibt, hat keinen Einfluss. Der Spezifikationsaufwand für das Prozessmodell ω_{spez} wird mit 90 bewertet. Für das Prozessmodell ω_{min} ergibt sich ein Aufwand von 74. Als Optimum stellt sich für das Prozessmodell $\omega_{opt} = 69$ heraus. Insgesamt sind 1041 Bewegungen im Baum erforderlich. Bereits nach 34 Schritten ist das Prozessmodell ω_{opt} erkannt.

5.3 Interpretation der Muster

Das Ergebnis des Branch-and-Bound-Verfahrens liefert Zerlegungen von Ereignisgraphen. Jede Zerlegung stellt eine Menge von Komponenten dar, die sich in Äquivalenzklassen einteilen lässt. Die Einteilung erfolgt auf der Grundlage der Muster der Komponenten.

Das Muster eines Ereignisgraphen ist eine Zeichenkette, die sich aus den Knotenbeschriftungen abbilden lässt. Nach Abschnitt 4.2.5 werden als Knotenbeschriftungen die Identifikatoren der Ereignisvorlagen verwendet. Jedes Ereignis kennt seine Ereignisvorlage. Ein Muster beschreibt somit den topologischen Zusammenhang von Ereignisvorlagen als Zeichenkette. Komponenten, die zu derselben Äquivalenzklasse gehören, haben denselben topologischen Zusammenhang in der Menge ihrer Ereignisvorlagen.

Eine Menge an Ereignisvorlagen und ihr topologischer Zusammenhang stellt nach Abschnitt 3.3.4 eine Leistungsprozessvorlage dar. Das Muster einer Komponente lässt eine Aussage darüber zu, ob unterschiedliche Komponenten von derselben Leistungsprozessvorlage abstammen können. Welche konkrete Leistungsprozessvorlage dies ist, geht aus einem

Muster nicht hervor. Im Allgemeinen existiert noch nicht einmal eine mustergleiche Leistungsprozessvorlage in der Menge der modellierten Leistungsprozessvorlagen. Bei der Berechnung optimaler Zerlegungen entstehen Komponenten, deren Gestalt sich nicht allein durch das Verknüpfen von existierenden Leistungsprozessvorlagen ergibt, bzw. durch das Zerlegen einer existierenden Leistungsprozessvorlage in mehrere kleinere. Abschnitt 5.3.1 geht auf die Gestalt der Komponenten ein, die sich aus berechneten Zerlegungen ergeben.

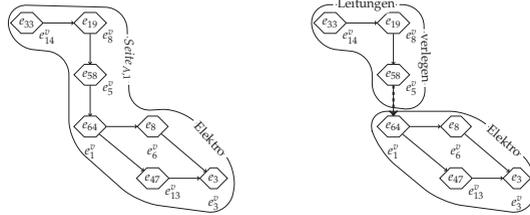
Da das Branch-and-Bound-Verfahren einen Prozess im Allgemeinen auf eine Art und Weise in Komponenten zerlegt, für die es keine bereits modellierten Leistungsprozessvorlagen gibt, ist es erforderlich, auf der Grundlage einer Komponente eine Leistungsprozessvorlage ableiten zu können. In Abschnitt 5.3.2 wird auf die Zusammenhänge eingegangen, um aus einer Komponente eine Leistungsprozessvorlage zu bestimmen.

5.3.1 Gestalt der Komponenten einer Zerlegung

Im Rahmen der Modellierung ist ein Projekt in Leistungsgegenstände einzuteilen. Ihnen sind Leistungsprozesse zuzuordnen. Aus der Zuordnung ergeben sich Teilprozesse - einer für jeden Leistungsgegenstand. Jeder Teilprozess stellt eine Komponente dar. Die Spezifikation von zusätzlichen Voraussetzungen führt zu einer Vernetzung der Teilprozesse. Das Ergebnis ist ein Gesamtprozess. Für diesen Gesamtprozess lässt sich im Nachhinein die Menge seiner optimalen Zerlegungen bestimmen. Vergleicht man die Gestalt der Komponenten, die sich aus der Berechnung ergeben, mit denen, die für die Modellierung verwendet wurden, so lassen sich vier Fälle unterscheiden: (1) Eine spezifizierte Komponente entspricht einer berechneten Komponente. Dieser Fall liefert keine neue Erkenntnis und wird daher nicht weiter verfolgt. (2) Eine spezifizierte Komponente zerfällt in kleinere Einheiten, (3) eine optimale Komponente setzt sich aus mehreren spezifizierten Komponenten zusammen und (4) die Grenzen der spezifizierten Komponenten haben keinen Zusammenhang mit den Grenzen der berechneten Komponenten. Abbildung 5.7 zeigt jeweils ein Beispiel für die relevanten Fälle.

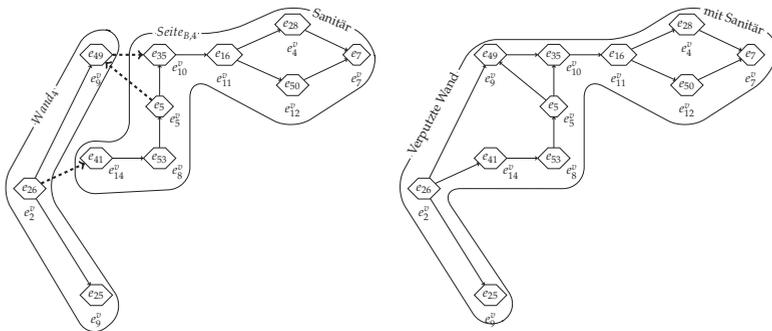
Durch die Berechnung optimaler Zerlegungen ergeben sich Komponenten, für die es keine modellierten Leistungsgegenstandsvorlagen gibt. Ein formaler Zusammenhang zwischen den spezifizierten und den berechneten Leistungsgegenstandsvorlagen lässt sich nicht aufstellen. Weicht das Muster einer berechneten Komponente von den Mustern der modellierten Komponenten ab, ist eine neue Leistungsgegenstandsvorla-

spezifizierte Komponente $\xrightarrow{1:n}$ optimale Komponente



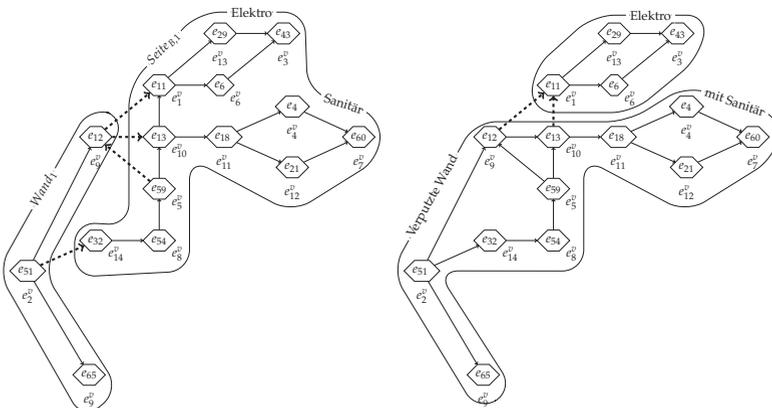
(a) Spezifizierte Komponenten zerfallen

spezifizierte Komponente $\xrightarrow{n:1}$ optimale Komponente



(b) Spezifizierte Komponenten verschmelzen

spezifizierte Komponente $\xrightarrow{n:m}$ optimale Komponente



(c) Die Grenzen der Komponenten haben keinen Zusammenhang

Abbildung 5.7: Vergleich von Komponenten

ge anzulegen.

Der Umfang generierter Leistungsgegenstandsvorlagen folgt nicht einer aus der Anschauung festgelegten Einteilung, die sich aus einer gewerkeorientierten, lageorientierten oder sonstigen festgelegten Sichtweise ergibt. Es stellt sich die Frage, ob sich aus optimalen Zerlegungen Strukturierungskriterien ergeben, die die in der Praxis üblichen in Frage stellen. Hierfür ist die Auswertung einer Vielzahl an Prozessmodellen erforderlich. Auswertungen dieser Art haben zu diesem Zeitpunkt noch nicht in ausreichendem Umfang stattgefunden, um diesbezüglich Ergebnisse vorstellen zu können.

5.3.2 Von der Komponente zur Leistungsprozessvorlage

In Abschnitt 4.2.1 wurde beschrieben, wie sich ein in ARIS modellierter Prozess auf einen Ereignisgraphen abbilden lässt. Ausgangspunkt sind die projektspezifischen Informationen, die als Instanzen von Entitäten modelliert vorliegen. Ihr Zusammenhang wird in Entity-Relationship-Modellen in der Datensicht beschrieben. Entitäten, die projektspezifische Informationen beschreiben, basieren auf Vorlagen, in denen die allgemeinen projektunabhängigen Informationen modelliert sind. Vorlagen sind ebenfalls Instanzen von Entitäten, deren Zusammenhang in Entity-Relationship-Modellen beschrieben wird.

Um ablaufbezogene Aspekte zu modellieren, werden die projektspezifischen Informationen auf Instanzen vom Typ Funktion und Ereignis der Steuerungssicht abgebildet. Im Ergebnis liegt der betrachtete Prozess als ereignisgesteuerte Prozesskette vor. Der Modellierungsansatz, Vorgänge durch Voraussetzungen und Ergebnisse zu beschreiben, führt zu ereignisgesteuerten Prozessketten, auf deren Pfaden Funktionen und Ereignisse immer im Wechsel auftreten. Das Ergebnis ist ein bipartiter Graph. Da auf einen Vorgang immer nur ein Ergebnis folgt, kann der bipartite Graph strukturerhaltend auf einen schlichten Graphen abgebildet werden, dessen Knotenmenge ausschließlich aus der Menge der Ereignisse besteht. Dieser wird als Ereignisgraph bezeichnet.

Dieses Vorgehen lässt sich umkehren. Die Knoten eines Ereignisgraphen sind Ereignisse. Ereignisse sind Voraussetzungen und Ergebnisse von Vorgängen. Für einen Ereignisgraphen kann unter Berücksichtigung der in den Entity-Relationship-Modellen definierten Verknüpfungen auf die Menge der Vorgänge, der Voraussetzungen und der Ergebnisse geschlossen werden. Elemente dieser Mengen sind mit ihren Vorlagen verknüpft. Aus den Vorlageelementen lässt sich, konsistent zu den Ereig-

nissen des Ereignisgraphen, eine bisher nicht modellierte Leistungsprozessvorlage ableiten. Leistungsgegenstandsvorlagen und Leistungsgegenstände, ergeben sich aus Vorgangspositionen. Vorgangspositionen verknüpfen projektspezifische Informationen und projektunabhängige Informationen. Bei der formalen Bestimmung von Leistungsgegenstandsvorlagen und Leistungsgegenständen werden die Informationen der Vorgangspositionen entsprechend ihrer Qualität den jeweiligen Instanzen zugeordnet.

Die Bestimmung von Leistungsgegenstandsvorlagen und Leistungsprozessvorlagen aus Ereignisgraphen ist erforderlich, da es für die Komponenten der berechneten Zerlegungen, die das Ergebnis des Branch-and-Bound-Verfahrens sind, keine Vorlagen gibt. Für die Bestimmung von projektunabhängigen Prozessbeschreibungen stehen nur die Komponenten der Zerlegungen zur Verfügung. Aus diesen sind die Vorlagen zu ermitteln. Alle Vorlagen, die für eine Zerlegung erforderlich sind, stellen ein Repräsentantensystem dar, die die Komponenten einer Zerlegung in Äquivalenzklassen einteilt.

5.4 Beispiel 2: Herstellung von Wänden

In Anlehnung an das bereits behandelte Beispiel wird in diesem Abschnitt ein Beispiel betrachtet, das sich nur geringfügig von dem ersten unterscheidet. Im ersten Beispiel sind die Wände *Wand 3* und *Wand 4* nur jeweils einseitig mit Installationen ausgestattet. In dem hier betrachteten Beispiel sind auf den freien Oberflächen weitere Installationen vorgesehen. *Wand 3* erhält auf der Seite *Seite B* eine Sanitärinstallation und *Wand 4* erhält auf der Seite *Seite A* eine Elektroinstallation. Das betrachtete Projekt wird in die folgenden 12 Leistungsgegenstände eingeteilt:

Seite A	Wand	Seite B
<i>Seite_{A,1}</i>	– <i>Wand₁</i>	– <i>Seite_{B,1}</i>
<i>Seite_{A,2}</i>	– <i>Wand₂</i>	– <i>Seite_{B,2}</i>
<i>Seite_{A,3}</i>	– <i>Wand₃</i>	– <i>Seite_{B,3}</i>
<i>Seite_{A,4}</i>	– <i>Wand₄</i>	– <i>Seite_{B,4}</i>

Die Leistungsgegenstandsvorlagen und die Leistungsprozessvorlagen bleiben gleich. Folgende Zuordnung wird für die Leistungsgegenstände

vorgenommen:

Leistungsgegenstand	Leistungsprozessvorlage
<i>Wand₁</i>	<i>verputzte Mauerwerkswand</i>
<i>Wand₂</i>	<i>verputzte Mauerwerkswand</i>
<i>Wand₃</i>	<i>verputzte Mauerwerkswand</i>
<i>Wand₄</i>	<i>verputzte Mauerwerkswand</i>

Leistungsgegenstand	Leistungsprozessvorlage
<i>Seite_{A,1}</i>	<i>Elektroinstallation</i>
<i>Seite_{B,1}</i>	<i>Elektro – und Sanitärinstallation</i>
<i>Seite_{A,2}</i>	<i>Elektroinstallation</i>
<i>Seite_{B,2}</i>	<i>Elektro – und Sanitärinstallation</i>
<i>Seite_{A,3}</i>	<i>Elektroinstallation</i>
<i>Seite_{B,3}</i>	<i>Sanitärinstallation</i>
<i>Seite_{A,4}</i>	<i>Elektroinstallation</i>
<i>Seite_{B,4}</i>	<i>Sanitärinstallation</i>

Abbildung 5.8 zeigt den spezifizierten Gesamtprozess. Der Graph besteht aus 80 Knoten und 98 Kanten. Ohne Berücksichtigung der wiederkehrenden Teilprozesse ist die Bewertung für den Aufwand des Graphen 178. Unter Berücksichtigung der zu spezifizierenden Kanten ergibt sich ein Gesamtaufwand für das Prozessmodell ω_{spez} von 98. Abbildung 5.9 zeigt die Zerlegung des spezifizierten Prozesses in seine minimalen Komponenten. Aufgrund der Ähnlichkeit der Wandaufbauten sinkt die Anzahl der minimalen Komponenten auf 10. Insgesamt sind 10 Kanten zu variieren. Das Prozessmodell ω_{min} wird mit einem Aufwand von 70 bewertet. Es ergeben sich 505 Komponentenkombinationen, deren Komponenten sich auf 129 verschiedene Muster abbilden lassen. Auf der Grundlage dieser Ausgangssituation beginnt die Suche. Abbildung 5.10 zeigt die berechnete optimale Zerlegung und den Verlauf des Verfahrens. Der Lösungsaufwand für das erste Beispiel umfasst 1.024 Bewegungen im Entscheidungsbaum. Obwohl der Graph aus dem zweiten Beispiel größer ist als der Graph aus dem ersten Beispiel, sind nur 75 Bewegungen erforderlich. Dies sind nur 7% des Aufwands verglichen mit dem ersten Beispiel.

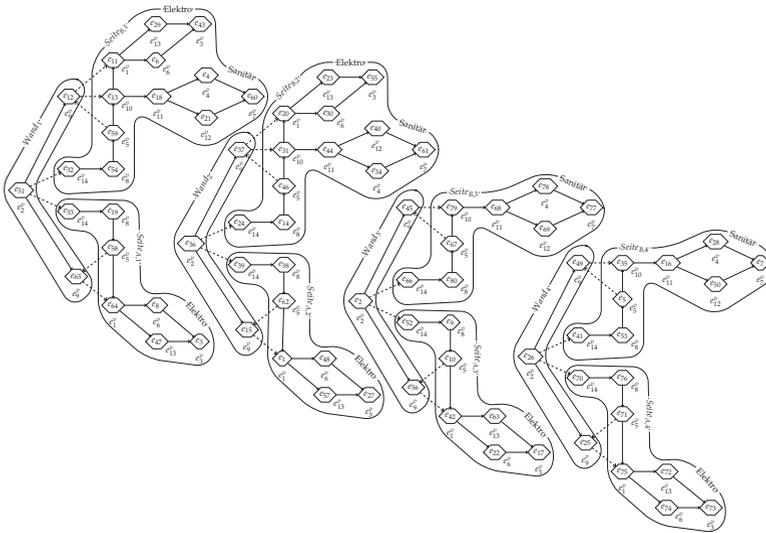


Abbildung 5.8: Beispiel 2 - Spezifiziertes Prozessmodell ω_{spez}

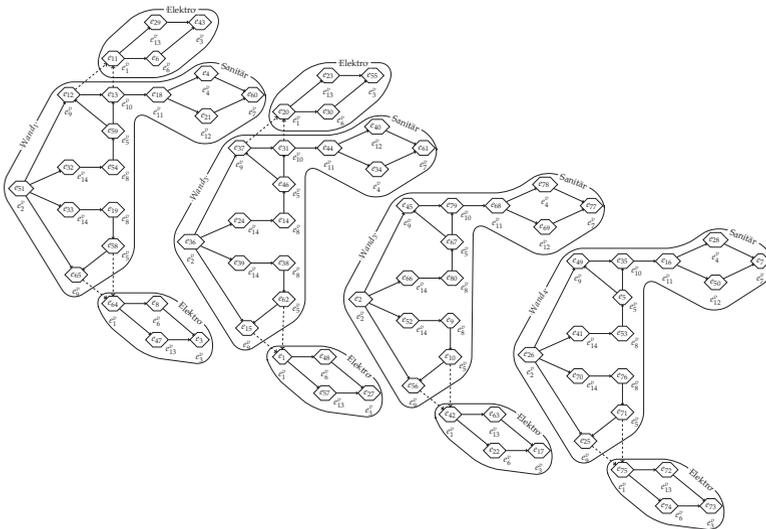
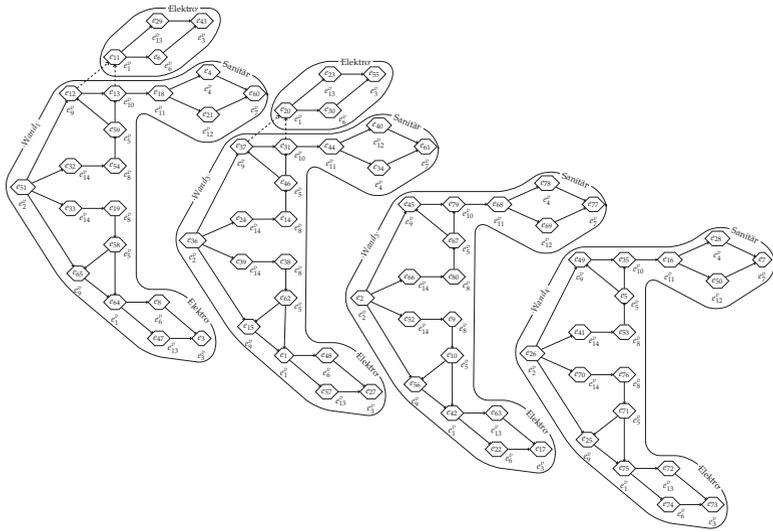
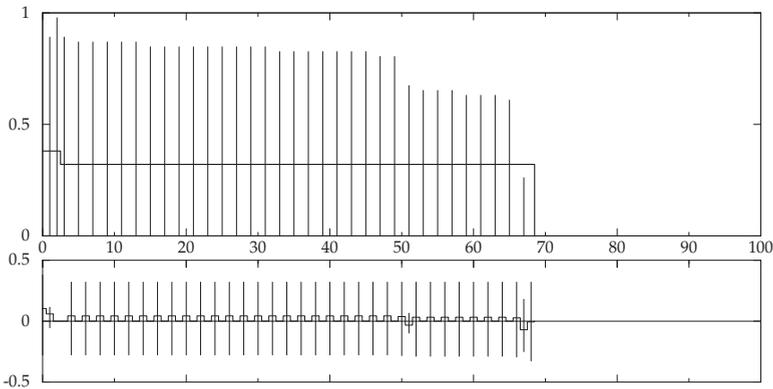


Abbildung 5.9: Beispiel 2 - Minimale Komponenten ω_{min}



(a) Optimales Prozessmodell ω_{opt}



(b) Verlauf der Berechnung

Abbildung 5.10: Beispiel 2 - Optimale Zerlegung und Verlauf des Verfahrens

In Abbildung 5.11 sind die charakteristischen Größen zusammengefasst und für die beiden Beispiele gegenübergestellt.

5.5 Aufwandsbetrachtungen

Das Branch-and-Bound-Verfahren zur Bestimmung optimaler Zerlegungen gliedert sich in drei Schritte: (1) Zunächst werden die Minimalen Komponenten ermittelt. Diese stellen Ausschnitte des Graphen dar, die nicht weiter zerlegt werden müssen. (2) Auf der Grundlage der minimalen Komponenten werden Komponentenkombinationen entwickelt. Eine Komponentenkombination dient der Festlegung von Kantenmengen, die im Rahmen des Branch-and-Bound-Verfahrens mit jeder Bewegung im Entscheidungsbaum zu variieren ist. (3) Im dritten Schritt erfolgt die Suche nach der Menge der optimalen Zerlegungen. Der Suche liegt der Entscheidungsbaum zugrunde. Der erste und der zweite Schritt stellen das Initialisieren des Entscheidungsbaums dar. Der dritte Schritt das systematische Durchsuchen des Lösungsraums.

Zu (1): Durch den ersten Schritt wird die Anzahl der Kanten, die effektiv zu betrachten sind, herabgesetzt. Ohne großen Aufwand lässt sich bereits eine Kantenmenge fixieren, von der man sicher weiß, dass ihr Verbleiben im Graphen zu überschneidungsfreien, sich wiederholenden Komponenten führt. Das Entfernen einer Kante, die zu einer minimalen Komponente gehört, resultiert zwangsläufig in einem Anstieg des Aufwands, ihr Verbleiben entsprechend in einer Einsparung. Darüber hinaus tragen die minimalen Komponenten bei der Bestimmung der Komponenten für die Komponentenkombinationen zu einer Reduzierung des Aufwands bei. Je mehr Kanten durch die minimalen Komponenten fixiert sind, desto kleiner ist die Anzahl der Knoten in den Vorgänger- und Nachfolgerbäumen. Die abgeminderte Menge an Knoten reduziert die Anzahl der resultierenden Komponenten. Weniger Komponenten haben entsprechend weniger Komponentenkombinationen zur Folge.

Zu (2): Das Ermitteln der Komponenten erfolgt auf der Grundlage der Vorgänger- und Nachfolgerbäume. Ausgehend von einer minimalen Komponente werden zusammenhängende Zellen entwickelt, auf deren Größe durch Abbruchkriterien Einfluß genommen werden kann. Das Entwickeln der Zellen wird von jeder minimalen Komponente aus angestoßen. Auf diese Weise ist sichergestellt, dass alle zusammenhängenden Komponenten erfasst werden. Vergleicht man dieses Vorgehen, das die topologischen Informationen berücksichtigt, mit einem kombinatorischen

ω_{spez}	Beispiel 1	Beispiel 2
Anzahl der Knoten	65	80
Anzahl der Kanten	83	104
Anzahl der Komponenten	10	12
Anzahl der Muster	4	4
Anzahl der spezifizierten Kanten	20	26
Aufwand	$a_{\omega}(\omega_{spez})$	90
98		
ω_{min}		
Anzahl der Komponenten	18	10
Anzahl der Muster	4	2
Anzahl der zu variierenden Kanten	20	12
Aufwand	$a_{\omega}(\omega_{min})$	74
70		
Verfahren		
Anzahl der relevanten Komponenten	1.368	326
Anzahl der Muster	652	129
Anzahl der relevanten Komponentenkombinationen	901	505
Anzahl der Bewegungen	1.041	75
	100,00%	7,20%
Ω_{opt}		
Anzahl der Lösungen	1	1
Anzahl der Komponenten	12	6
Anzahl der Muster	5	2
Anzahl der zu spezifizierenden Kanten	16	4
Rest	5	0
Aufwand	$a_{\omega}(\omega_{opt})$	69
59		

Abbildung 5.11: Gegenüberstellung von Beispiel 1 und Beispiel 2

Ansatz, der bei der Ermittlung von Kantenmengen diese Informationen nicht berücksichtigt, so besteht ein großer Unterschied im Aufwand. Unter Berücksichtigung von Nachbarschaften ist eine Kante nur mit ihren unmittelbar anknüpfenden Kanten zu kombinieren. Ohne die Berücksichtigung der Nachbarschaften ist sie mit allen verbleibenden Kanten zu kombinieren. Dies führt zu einer großen Anzahl an Kantenkombinationen, die untersucht werden müssen, von denen aber nur wenige relevant sind.

Sind die Komponentenmengen bestimmt, dann werden diese nach ihrem Muster in Äquivalenzklassen eingeteilt. Für jede Äquivalenzklasse sind die überschneidungsfreien Komponentenkombinationen zu ermitteln. Das Initialisieren des Entscheidungsbaums ist abgeschlossen.

Zu (3): Bevor die Suche im Entscheidungsbaum beginnt, werden die überschneidungsfreien Komponentenkombinationen entsprechend der normierten Bewertung aufsteigend geordnet. Diese Bewertung ändert sich im Verlauf der Suche nicht mehr. In der Reihenfolge, die sich aus der Tiefensuche ergibt, wird der Entscheidungsbaum traversiert. Für den Aufwand der Suche ist entscheidend, wie schnell der Wert der oberen Schranke $a_{\omega,opt}$ durch das Finden von vorläufig optimalen Lösungen absinkt, und wie die Zusammensetzung der überschneidungsfreien Komponentenkombinationen ist. In Abbildung 5.5 im Intervall 223 – 782 stellt sich eine Situation dar, die sich als besonders ungünstig für das Verfahren erweist. In diesem Intervall wiederholt sich dasselbe Szenario fünf mal, ohne ein Ergebnis zu liefern. Dies kann darauf zurückgeführt werden, dass es Komponentenkombinationen aus unterschiedlichen Äquivalenzklassen gibt, die voneinander unabhängige Bereiche des Graphen betreffen. Hierzu ein Beispiel: Gegeben seien Komponentenkombinationen der Äquivalenzklasse p , die einen Bereich im Graphen betreffen, der von sonst keiner anderen Komponente berührt werde. Existieren nun an einem Knoten mehrere Kinder, denen eine Komponentenkombination der Äquivalenzklasse p zugrunde liegt, dann wird der Verlauf der Suche für diese Kinder identisch sein. Dies liegt daran, dass die Menge der verbleibenden Komponentenkombinationen für diese Kinder gleich ist. Stellt man fest, dass die Suche nicht zu einem Erfolg geführt hat, dann kann aus diesem Ergebnis auf das der verbleibenden Kinder geschlossen werden, denen dieselbe normierte Bewertung zugeordnet ist.

Vor dem Hintergrund der Komplexität der Problemstellung ist die Größe der Graphen, für die die Menge der optimalen Prozessmodelle berechnet werden kann, begrenzt. Hierbei lassen sich die Grenzen des Verfahrens nicht zweckmäßig durch die Größe der Graphen angeben. Ihre Struktur hat einen entscheidenderen Einfluss auf das Lösungsverhalten. Eine allgemeine Formel, um den Aufwand für diese Vorgehensweise ab-

156

zuschätzen, lässt sich nicht angeben.

Kapitel 6

Zusammenfassung

Voraussetzung für einen erfolgreichen Projektablauf, bei stetig steigenden Anforderungen, ist eine detaillierte Kenntnis über die projektspezifischen Abläufe von der Planung über die Ausführung bis hin in den Betrieb. Kurze Realisierungsphasen können nur erreicht werden, wenn die Planung und die Ausführung sich so ineinander verschachteln, dass sie über weite Teile parallel erfolgen können. Die Komplexität, die Bauabläufe auszeichnet, kann nicht mehr allein aus der Anschauung heraus bewältigt werden.

Das übergeordnete Ziel dieser Arbeit ist es, das Gestalten komplexer Bauabläufe durch formale Methoden zu unterstützen, um konsistente und korrekte Ablaufmodelle für die Netzplantechnik zur Verfügung zu stellen. Im Fokus steht die Reduzierung des Spezifikationsaufwands und die Bewältigung der Komplexität von Bauabläufen.

Die Herangehensweise gliedert sich in zwei Schwerpunkte: (1) der Beschreibung einer formalen Modellierungsmethodik zum Gestalten von Bauabläufen und (2) einem Verfahren zur Mustererkennung in Bauabläufen mit dem Ziel, die Eingangsgrößen für die Modellierungsmethodik kontinuierlich zu verbessern.

6.1 Modellierungsmethodik

Die Ermittlung der Abhängigkeiten zwischen Vorgängen in komplexen Bauabläufen stellt eine Herausforderung dar. In der Praxis erfolgt diese Tätigkeit aus der Anschauung heraus. Auch mit großem Aufwand lässt sich nicht sicherstellen, ob alle Abhängigkeiten für einen reibungslosen Ablauf erfasst sind.

Aufbauend auf den Ansätzen von Huhnt und Lawrence [2004], Huhnt [2005] und Huhnt und Enge [2006] wird in dieser Arbeit eine Modellie-

runzungsmethodik zum Erstellen von Bauabläufen beschrieben, die es erlaubt, auf der Grundlage von formalen Algorithmen die Abfolge von Vorgängen zu berechnen.

Hierbei wird ein Projekt in Leistungsgegenstände zerlegt und jedem Leistungsgegenstand wird eine Leistungsprozessvorlage zugeordnet. Aus dem Tupel, bestehend aus projektspezifischem Leistungsgegenstand und projektunabhängiger Leistungsprozessvorlage, lässt sich ein projektspezifischer Leistungsprozess ableiten. Leitet man für jeden Leistungsgegenstand eines Projekts auf der Grundlage der ihm zugeordneten Leistungsprozessvorlage einen Leistungsprozess ab, so entsteht ein Ablaufmodell, das aus unabhängigen Teilabläufen besteht. Dieses Vorgehen erlaubt die Trennung von Produktinformationen und Ablaufinformationen.

Zusätzliche Verknüpfungen sind erforderlich, um die Teilabläufe miteinander in Beziehung zu setzen. Das Ergebnis ist ein Graph, der die technologisch zwingenden Abhängigkeiten zwischen Vorgängen vollständig enthält. Damit steht ein Rohling für die Netzplantechnik zur Verfügung, der um weitere Angaben, wie beispielsweise Ressourcenbegrenzungen, Terminvorgaben etc., ergänzt werden kann. Aufgrund der formalen Herangehensweise sind die Eigenschaften des resultierenden Prozessmodells mathematisch begründet und die Vollständigkeit und Korrektheit kann konsistent zur Eingabe gewährleistet werden.

Der Spezifikationsaufwand für ein Ablaufmodell ergibt sich aus dem Anlegen der Leistungsprozessvorlagen, der Zerlegung des Projekts in Leistungsgegenstände und der Spezifikation der zusätzlichen Abhängigkeiten. Nur das Rückrechnen von optimalen Zerlegungen erlaubt eine Aussage darüber, wie effizient - bezogen auf den Spezifikationsaufwand - eine Menge an eingesetzten Leistungsprozessvorlagen für einen gegebenen Prozess war. Die Gegenüberstellung der im Rahmen der Modellierung verwendeten Leistungsprozessvorlagen mit denen, die sich aus der Rückrechnung als optimal erweisen, erlaubt die Anpassung der in der Wissensbasis enthaltenen Leistungsprozessvorlagen, um Bauabläufe von zukünftigen Projekten effizienter entwickeln zu können.

6.2 Mustererkennung

Da der Erfolg und der Nutzen für das Modellieren komplexer Bauabläufe auf der Grundlage des hier beschriebenen Ansatzes entscheidend vom Spezifikationsaufwand abhängt, wird der kontinuierlichen Anpassung und Verbesserung der Basis an Leistungsprozessvorlagen große Aufmerksamkeit beigemessen.

Die Ermittlung optimaler Zerlegungen lässt sich als Suche nach Mustern in einem Graphen auffassen. Da die Muster, nach denen im Graphen gesucht werden soll, im Vorfeld nicht bekannt sind, sondern selbst ebenfalls einen Teil der Lösung darstellen, ist die Komplexität der Problemstellung exponentiell. Nur durch die Berücksichtigung von Randbedingungen, die sich aus dem Modellierungsansatz ergeben, lässt sich die Komplexität der Suche reduzieren.

Im Rahmen dieser Arbeit wird ein Branch-and-Bound-Verfahren beschrieben, das den Lösungsraum vollständig berücksichtigt und ohne die Verwendung von Heuristiken die Menge der optimaler Zerlegungen bestimmt. Ausgangspunkt der Untersuchung ist der gesamte Lösungsraum. Das Wissen darüber, dass Zerlegungen mit mehrfach auftretenden Mustern denjenigen überlegen sind, für die sich keine Wiederholungen ergeben, kann genutzt werden, um die Anzahl der zu betrachtenden Prozessmodelle deutlich herabzusetzen. Anstatt einzelne Kanten zu variieren, werden Komponentenkombinationen variiert. Nur diejenigen Komponenten, die zu einer Einsparung beitragen können, werden betrachtet. Dies bedeutet, dass mit jedem Schritt nicht eine einzelne Kante, sondern Kantenmengen variiert werden.

Die Laufzeit des Verfahrens durchläuft drei wesentliche Etappen: (1) Die Bestimmung der minimalen Komponenten, (2) das Entwickeln der Komponentenkombinationen auf der Grundlage der minimalen Komponenten und (3) die Suche nach den optimalen Prozessmodellen. Der Suche liegt ein Entscheidungsbaum zugrunde.

Das Prinzip des Entscheidungsbaums besteht darin, zunächst alle Kanten des Graphen als variabel zu betrachten. Mit jedem Schritt in den Baum hinein wird eine Teilmenge der Kanten fixiert. Mit einem Schritt zurück in Richtung der Wurzel wird eine bereits fixierte Teilmenge wieder variabel. Die Menge der Kanten, die mit einem Schritt verbunden ist, ergibt sich aus der Auswahl einer zulässigen Komponentenkombinationen. Erreicht man einen Knoten im Entscheidungsbaum, für den alle Kanten fixiert sind, dann beschreibt dieser ein Prozessmodell. Er stellt ein Blatt des Baums dar. Unterschiedliche Blätter beschreiben unterschiedliche Prozessmodelle. Gesucht sind diejenigen Prozessmodelle, deren Spezifikationsaufwand minimal ist.

Hiermit liegt eine Methodik vor, um die Qualität der Leistungsvorlagen rückblickend für einen gegebenen Prozess zu beurteilen. Die Auswertung abgeschlossener Projekte kann dazu beitragen, die Basis der Leistungsvorlagen für anstehende, ähnliche Projekte zu optimieren. Vor dem Hintergrund der Komplexität der Problemstellung ist die Größe der Graphen, für die die Menge der optimalen Prozessmodelle be-

rechnet werden kann, begrenzt. Hierbei lassen sich die Grenzen des Verfahrens nicht zweckmäßig durch die Größe der Graphen angeben. Ihre Struktur hat einen entscheidenderen Einfluss auf das Lösungsverhalten. In Abschnitt 5.5 wird eine Aufwandsbetrachtung vorgenommen und auf einen Aspekt eingegangen, aus dem ein weiteres bisher ungenutztes Terminierungskriterium abgeleitet werden kann.

Kapitel 7

Ausblick

7.1 Methodik zur Auswertung von optimalen Zerlegungen

In dieser Arbeit wird ein Verfahren beschrieben, um anhand der Muster von Teilabläufen auf optimale Zerlegungen zu schließen. Ziel der Zerlegung ist es, die Erfahrung aus abgeschlossenen Projekten zu nutzen, um die Basis an Leistungsgegenstandsvorlagen und Leistungsprozessvorlagen für die Spezifikation anstehender Projekte kontinuierlich zu verbessern.

Wie die Auswertungen aus Kapitel 5 zeigen, treten unterschiedliche Fälle bei dem Vergleich der Gestalt von Leistungsprozessvorlagen auf. Auf der einen Seite stehen die Leistungsprozessvorlagen, die in die Modellierung von Bauabläufen eingeflossen sind und auf der anderen Seite stehen die Leistungsprozessvorlagen, die sich aus den optimalen Zerlegungen ergeben. Es treten vier unterschiedliche Fälle auf: (1) Eine spezifizierte Leistungsprozessvorlage entspricht einer berechneten Leistungsprozessvorlage, (2) eine ursprüngliche Leistungsprozessvorlage zerfällt in kleinere Einheiten, (3) eine berechnete Leistungsprozessvorlage setzt sich aus ursprünglichen Leistungsprozessvorlagen zusammen und (4) die Grenzen der Leistungsprozessvorlagen sind nicht miteinander vergleichbar.

Vor dem Hintergrund der Fragestellung, wie die berechneten Ergebnisse methodisch zu einer Anpassung der vorhandenen Leistungsprozessvorlagen genutzt werden können, hat im Rahmen der bisherigen Arbeit noch keine systematische Untersuchung stattgefunden. Der einfachste Fall ist, die Basis um die berechneten Leistungsprozessvorlagen zu erweitern. Dies führt jedoch zu einer ständig wachsenden Anzahl an Leistungsprozessvorlagen, deren Verwaltung sich in zunehmendem Maße erschwert.

Insbesondere würde die Auswahl geeigneter Leistungsprozessvorlagen im Rahmen der Modellierung den planenden Ingenieur mit einer unnötigen Komplexität konfrontieren. Der Idealfall ist, einen formalen Ansatz zu entwickeln, der die Basis an Leistungsprozessvorlagen auf der Grundlage der berechneten Ergebnisse anpasst. Da jedoch die berechneten Leistungsprozessvorlagen nur in Bezug auf den ausgewerteten Bauablauf ein Optimum darstellen, ist es unbegründet, sie als optimal, bezüglich anstehender Projekte, anzusehen. Derzeit sind die Erfahrungen und das Fachwissen von Planungsingenieuren erforderlich, um die berechneten Ergebnisse zu bewerten. Auf der Grundlage der Bewertung kann entschieden werden, ob vorhandene Leistungsprozessvorlagen anzupassen, als Ganzes zusätzlich in die Basis aufzunehmen oder zu verwerfen sind.

7.2 Zusammenwirken mit modellbasierten Ansätzen

In zunehmendem Maße wird auf die ganzheitliche Betrachtung von Datenmodellen bzw. Produktmodellen im Bauwesen Wert gelegt. Ansätze, die sich mit dieser Zielsetzung befassen, werden dem Gebiet des *Building Information Modeling (BIM)*¹ zugeordnet. Hierbei ist es das Ziel, die Informationen zu einem Bauwerk - seien dies Informationen zum Entwurf, zum Tragwerk, zum Bauablauf, zur Kalkulation, zum Betrieb, u.a. - in einem integrierten Datenmodell vorzuhalten, das den gesamten Lebenszyklus eines Bauwerks umfasst. Dieses ist unabhängig von einzelnen Projektphasen, der Sichtweise von Projektbeteiligten und spezifischen Software-Systemen. Produktneutrale Schnittstellen erlauben das selektive Lesen und Bearbeiten von Daten. Einen Standard zur digitalen Beschreibung von Gebäudemodellen wurde von der *Industrieallianz für Interoperabilität e.V. - IAI*² entwickelt und in Form der *Industry Foundation Classes*, dem IFC-Modell etabliert.

Die in Kapitel 3 vorgestellte Modellierungsmethodik folgt in ihrer Logik dem modellbasierten Ansatz der BIM-Methodik und wird von dem Einsatz entsprechender Modelle in Bauprojekten profitieren. Dadurch, dass mit dem ersten Strich - Striche in einer CAD-Zeichnung sind nicht mehr nur noch Striche, sondern z.B. Wände mit konkreten Eigenschaften - Eingaben in ein projektbegleitendes, für alle Beteiligten verfügbares Bauwerksmodell entsteht, ist bereits in frühen Projektphasen ein hohes Maß

¹ [Eastman et al., 2008, BIM Handbook: A Guide to Building Information Modeling]

² [IFC, 2009, Industry Foundation Classes (IFC)]

an Informationen abrufbar, für die nach der klassischen Vorgehensweise zusätzlicher Spezifikationsaufwand erforderlich ist. Die Zerlegung eines Bauwerks in seine Bauteile lässt sich bei modellbasierten Ansätzen aus dem Produktmodell auslesen und muss nicht extra ermittelt und eingegeben werden. Außerdem lässt sich die Zuordnung der Leistungsprozessvorlagen zu den Leistungsgegenständen automatisieren, indem Objekttypen der CAE³-Systeme auf geeignete Vorlagen aus dem in Kapitel 3 beschriebenen Modell abgebildet werden.

7.3 Mustererkennung in Bauteilzusammenhängen

Das in Kapitel 4 beschriebene Verfahren zur Bestimmung optimaler Zerlegungen basiert auf dem Erkennen von sich wiederholenden Untergraphen in einem Graphen. Hierbei werden die Beschriftungen von Knoten, die zu einer Komponente gehören, unter Berücksichtigung ihres Zusammenhangs auf Zeichenketten abgebildet und miteinander verglichen. Je mehr überschneidungsfreie Komponenten existieren, die sich auf dasselbe Muster abbilden lassen, desto günstiger ist ihr Einfluss auf die Bewertung der Zerlegung, denen diese Komponenten angehören.

Die hier beschriebene Methodik wird im Rahmen dieser Arbeit darauf angewendet, Leistungsprozessvorlagen für das Gestalten von Bauabläufen zu ermitteln. Je nachdem, wofür ein Knoten des Graphen steht und welche Bedeutung der ihm zugeordnete Bezeichner hat, lässt sich die Methodik auch auf andere Problemstellungen anwenden.

Nachfolgend wird darauf eingegangen, wie sich diese Methodik der Mustererkennung einsetzen lässt, um einen Teil der zusätzlich zu spezifizierenden Voraussetzungen für Vorgänge - nach Abschnitt 3.4.3 - zu berechnen. Bei den zu spezifizierenden Voraussetzungen für Vorgänge handelt es sich um Kanten, die bezogen auf den Graphen die unabhängigen Komponenten miteinander vernetzen.

Jedes Bauteil eines Bauwerks wird auf einen Knoten abgebildet. Hierbei ordnet man dem Identifikator eines Knotens den Identifikator des Bauteils zu und dem Bezeichner eines Knotens den Identifikator der Bauteilvorlage. Entsprechend der Nachbarschaften der Bauteile im Bauwerk werden die Knoten über Kanten miteinander verknüpft. Der so entstehende

³ „Computer Aided Engineering (CAE) ist die rechnerunterstützte Lösung technischer und wissenschaftlicher Probleme vor und während des ingenieurmäßigen Entwicklungs- und Konstruktionsprozesses.“ [ITWissen [2009]]

Graph beschreibt den topologischen Zusammenhang der Bauteile. Sucht man in diesem Graphen nach der oben skizzierten Methodik die Menge der optimalen Zerlegungen, so erhält man die optimalen Einteilungen des Bauwerks in Bauteilgruppen. Eine Bauteilgruppe ergibt sich aus einer Komponente. Unterschiedliche Bauteilgruppen, die sich aus Komponenten mit gleichem Muster ergeben, stellen Bauteilgruppen mit gleichartigen Bauteilen und Bauteilzusammenhängen dar. Nur für eine dieser Bauteilgruppen sind die zusätzlich zu spezifizierenden Voraussetzungen anzugeben. Für alle anderen Bauteilgruppen können die zugrunde liegenden Leistungsprozesse der Bauteile analog verknüpft werden.

Dieses Vorgehen erlaubt es, den Spezifikationsaufwand beim Gestalten von Bauabläufen weiter herabzusetzen. Auch in diesem Zusammenhang haben modellbasierte Ansätze nach Abschnitt 7.2 einen großen Nutzen, da die topologischen Zusammenhänge der Bauteile aus den Produktmodellen ausgelesen werden können.

Literaturverzeichnis

- AHO: *Projektsteuerung - Untersuchungen zum Leistungsbild des §31 HOAI und zur Honorierung für die Projektsteuerung*. Ausschuss der Ingenieurverbände für die Honorarordnung e.V., Bundesanzeiger Verl., Berlin, 1998.
- Altrogge, G.: *Netzplantechnik*. Gabler, Wiesbaden, 1979.
- ARH-Tabellen: *Arbeitszeit-Richtwerte Hochbau - Gesamtausgabe 1982-2001*. Tarifvertragsparteien der deutschen Bauwirtschaft, Zeitechnik-Verlag, Neu-Isenburg, 2001.
- Bauer, H.: *Baubetrieb*. Springer, Berlin, dritte Auflage, 2006.
- BAUORG: *Unternehmer-Handbuch für Bauorganisation und Betriebsführung*. Zentralverband des Deutschen Baugewerbes, Berlin, 1998.
- Brandenberger, J. und Ruosch, E.: *Projektmanagement im Bauwesen*. Bau-fachverlag, dritte Auflage, 1991.
- Brandenberger, J. und Ruosch, E.: *Ablaufplanung im Bauwesen*. Bau-fachverlag, 1993.
- Carlier, J. und Pinson, E.: An algorithm for solving the job-shop problem. In: *Manage. Sci.*, Band 35(2): S. 164–176, 1989.
- Chevallier, N.J. und Russell, A.D.: Developing a draft schedule using templates and rules. In: *Journal of Construction Engineering and Management*, Band 127(5): S. 391–398, 2001.
- Clark, J. und Holton, D. A.: *Graphentheorie. Grundlagen und Anwendungen*. Spektrum Akademischer Verlag, Heidelberg, 1994.
- Darwiche, A., Levitt, R. und Hayes-Roth, B.: Oarplan: generating project plans by reasoning about objects, actions and resources. In: *Artificial*

- intelligence for engineering design, analysis and manufacturing*, Band 2(3): S. 169–181, 1988.
- Davenport, T.H.: *Process Innovation: Reengineering Work Through Information Technology*. Harvard Business School Press, 1992.
- DIN 276-1: *Kosten im Bauwesen - Teil 1: Hochbau*. Deutsches Institut für Normung e.V., Beuth-Verlag, 2008.
- DIN 69900: *Projektmanagement - Netzplantechnik*. Deutsches Institut für Normung e.V., Beuth-Verlag, 2009.
- DIN 69901-5: *Projektmanagement - Projektmanagementsysteme - Teil 5: Begriffe*. Deutsches Institut für Normung e.V., Beuth-Verlag, 2009.
- DIN 69902: *Projektwirtschaft - Einsatzmittel - Begriffe*. Deutsches Institut für Normung e.V., Beuth-Verlag, 2009.
- DIN EN ISO 9000: *Qualitätsmanagementsysteme*. Deutsches Institut für Normung e.V., Beuth-Verlag, 2005.
- Dzeng, R.J.: Boiler Erection Scheduling Using Product Models and Case-Based Reasoning. In: *Journal of Construction Engineering and Management*, Band 123: S. 338, 1997.
- Dzeng, R.J. und Tommelein, I.D.: Using Product Models to Plan Construction. In: *Proceedings 5th International Conference on Computing in Civil and Building Engineering*, ASCE, Anaheim, CA, S. 1778–1785. 1993.
- Eastman, C.M., Eastman, C., Teicholz, P., Sacks, R. und Liston, K.: *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. Wiley, 2008.
- Enge, F.: Zustandsmodellierung als Grundlage für Ausführungsterminpläne. In: F. Schley, L. Weber (Hg.) *Forum Bauinformatik*, Brandenburgische Technische Universität Cottbus, Cottbus, S. 123–130. 2005.
- Fischer, M. und Aalami, F.: Scheduling with Computer-Interpretable Construction Method Models. In: *Journal of Construction Engineering and Management*, Band 122(4): S. 337–347, 1996.
- Fox, M.S., Allen, B.P., Smith, S.F., Strohm, G.A., Surnniary, S., Fos, R.S. und Smith, S.F.: *ISIS: a constraint-directed reasoning approach to job shop scheduling*. Carnegie-Mellon University, Robotics Institute, Pittsburgh, Pennsylvania, 1983.

- Gadatsch, A.: *Grundkurs Geschäftsprozess-management: Methoden und Werkzeuge für die IT-praxis: Eine Einführung für Studenten und Praktiker*. Friedr. Vieweg & Sohn Verlag, 2007.
- Gaitanides, M.: *Prozessorganisation: Entwicklung, Ansätze und Programme des Managements von Geschäftsprozessen*. Verlag Vahlen, München, 1983.
- Greiner, P., Mayer, P.E. und Stark, K.: *Baubetriebslehre. Projektmanagement*. Vieweg Verlag, 2005.
- Götzke, H.: *Netzplantechnik*. Fachbuchverlag, Leipzig, 1969.
- Hendrickson, C., Zozaya-Gorostiza, C., Rehak, D., Baracco-Miller, E. und Lim, P.: Expert System for Construction Planning. In: *Journal of Computing in Civil Engineering*, Band 1(4): S. 253–269, 1987.
- HOAI: *Verordnung über die Honorare für Leistungen der Architekten und Ingenieure (HOAI)*. Eurofassung 2002, Kohlhammer, 2006.
- Hoffmann, M.: *Zahlentafeln für den Baubetrieb*. Teubner, 6. Auflage, 2002.
- Hoitomt, D.J., Luh, P.B. und Pattipati, K.R.: A practical approach to job-shop scheduling problems. In: *Robotics and Automation*, Band 9(1): S. 1–13, 1993.
- Homann, K.: *Immobiliencontrolling: Ansatzpunkte einer lebenszyklusorientierten Konzeption*. Dt. Univ.-Verl.; Gabler, 1999.
- Huhnt, W.: Generating Sequences of Construction Tasks. In: Scherer, R.J., Katranuschkov, P. und Schapke, S.-E. (Hg.) *Proceedings of 22nd CIB-W78, CIB-W78*, Dresden, S. 17–22. 2005.
- Huhnt, W. und Enge, F.: Can algorithms support the specification of construction schedules? In: P, Katranuschkov (Hg.) *Process Modelling, Process Management and Collaboration, ITcon*, Band 11, S. 547–564. 2006.
- Huhnt, W. und Enge, F.: Consistent Information Management for Structuring Construction Activities. In: Rebolj, D. (Hg.) *Proceedings of 24th CIB-W78 Conference, 14th EG-ICE Workshop, 5th ITC@EDU Workshop*, University of Maribor, Maribor, S. 607–614. 2007.
- Huhnt, W. und Lawrence, J.: Methodik zur Modellierung konsistenter Soll-Vorgaben für komplexe Planungsprozesse. In: *Bauingenieur*, Springer, Band 79, S. 50–56. 2004.

- IFC: *Industry Foundation Classes*. Standard, Industriellianz für Interoperabilität e.V. (IAI), 2009. URL <http://www.buildingsmart.de>, (Stand: 10.05.2009).
- ITWissen: Das große Online-Lexikon für Informationstechnologie. Lexikon, DATACOM Buchverlag, 2009. URL <http://www.itwissen.info>, (Stand: 10.04.2009).
- Keller, G., Nüttgens, M. und Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK). In: Scheer, W. (Hg.) *Heft 89*, Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), Universität des Saarlandes, Saarbrücken. 1992.
- Kochendörfer, B., Liebchen, J.H. und Viering, M.G.: *Bau-Projekt-Management: Grundlagen und Vorgehensweisen*. Teubner, 2007.
- Kuhne, C., Ripberger, A., Aalami, F. und Fischer, M.: Neue Ansätze zur Projektplanung und Baustellensteuerung. In: *Bauingenieur*, Band 75: S. 15–20, 2000.
- Kurbel, K.: *Entwicklung und Einsatz von Expertensystemen*. Springer, Berlin, 1992.
- Levitt, R.E.: Knowledge-based planning systems: An engineering perspective. In: *Proceedings of the First International Conference on Expert Planning Systems*, IEE, S. 181–185. 1990.
- MasterFormat: *Edition Numbers and Titles*. Standard, The Construction Specifications Institute and Construction Specifications Canada, 2004. URL <http://www.csinet.org>, (Stand: 10.05.2009).
- Messmer, B.T. und Bunke, H.: Efficient subgraph isomorphism detection: a decomposition approach. In: *Knowledge and Data Engineering, IEEE Transactions on*, Band 12(2): S. 307–323, 2000.
- Navinchandra, D., Sriram, D. und Logcher, R.D.: GHOST: Project Network Generator. In: *Journal of Computing in Civil Engineering*, ASCE, Band 2, S. 239–254. 1988.
- Newell, A., Shaw, JC und Simon, HA: Report on a general problem-solving program for a computer. In: *Proceedings of the International Conference on Information Processing*, UNESCO House, Paris, S. 256–264. 1960.

- Niemann, H.: *Klassifikation von Mustern*. Universität Erlangen, zweite Auflage, 2003. URL <http://www5.informatik.uni-erlangen.de/fileadmin/Persons/NiemannHeinrich/klassifikation-von-mustern/m00-www.pdf>, (Stand: 20.04.2009).
- Pahl, P.J. und Damrath, R.: *Mathematische Grundlagen der Ingenieurinformatik*. Springer, Berlin, erste Auflage, 2000.
- PMBOK: *A Guide to the Project Management Body of Knowledge*. Official German Translation. Project Management Institute, Pennsylvania, 2003. Übersetzung von: *A guide to the project management body of knowledge (PMBOK guide) 2000 ed.*
- Richter, M.M.: Fallbasiertes Schließen. In: Görz, G. und Rollinger, C.-R. (Hg.) *Handbuch der künstlichen Intelligenz*, Oldenbourg Wissenschaftsverlag, Kapitel 11, S. 407–430. 2003.
- Scheer, A.-W.: *ARIS. Vom Geschäftsprozeß zum Anwendungssystem*. Springer, Berlin, dritte Auflage, 1998.
- Scheer, A.W.: *ARIS-Modellierungsmethoden, Metamodelle, Anwendungen*. Springer, Berlin, 2001.
- Schiller, K.: *Praktische Baukalkulation: Kostensicher kalkulieren von STLB-bau-dynamische Baudaten bis zum EFB-nachweis*. Beuth, 2003.
- Schuh, G.: *Produktionsplanung und -steuerung: Grundlagen, Gestaltung und Konzepte*. Springer, Berlin, dritte Auflage, 2006.
- Schwarze, J.: *Projektmanagement mit Netzplantechnik*. NWB Verlag, 9. Auflage, 2006.
- Seeling, R.: *Projektsteuerung im Bauwesen*. Teubner, Stuttgart, 1996.
- Seidlmeier, H.: *Prozessmodellierung mit Aris: Eine beispielorientierte Einführung für Studium und Praxis*. Vieweg Verlag, 2006.
- Stapf, W.: *Geschäftsprozessmanagement - Eine Konzeption zur prozessorientierten Unternehmens- (Re-)Organisation*. Dissertation, Universität Stuttgart, 2000.
- Stark, K. und Brecheler, W.: *Baubetriebslehre - Grundlagen: Projektbeteiligte, Projektplanung, Projektlauf*. Vieweg + Teubner Verlag, 2006.

- Waterman, D.A.: *A guide to expert systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1985.
- Winstanley, C.: A Planning System for Large-Scale Industrial Projects. In: *Proceedings of the IEE Colloquium on "Knowledge-Based Approaches to Automation in Construction"*, IEE, Band 129. 1995.
- Würfele, F. und Bielefeld, B.: *Bauobjektüberwachung: Kosten-Qualitäten-Termine-Organisation-Leistungsinhalt-Rechtsgrundlagen-Haftung-Vergütung*. Friedr. Vieweg & Sohn Verlag, 2007.
- Zimmermann, H.-J.: *Operations Research: Methoden und Modelle. Für Wirtschaftsingenieure, Betriebswirte, Informatiker*. Friedr. Vieweg & Sohn Verlag, Wiesbaden, zweite Auflage, 2008.

Abbildungsverzeichnis

2.1	Einsparungsmöglichkeiten im Projektverlauf	17
2.2	Strukturierung des Projekts in Phasen	19
3.1	Elemente einer Datensicht (eERM)	32
3.2	Elemente einer Prozesskette (eERM)	33
3.3	Projekt Garage	34
3.4	Projektunabhängige und -spezifische Informationen	35
3.5	Zusammenhang der Informationen am Beispiel Garage	36
3.6	Strukturieren von Elementen	37
3.7	eERM Leistungsgegenstand und -vorlage	39
3.8	eERM Leistungsprozess und -vorlage	40
3.9	eERM Leistungsgegenstand und Leistungsprozess	41
3.10	eERM Vorgang und -vorlage	43
3.11	Flussdiagramm einer Vorgangsposition	44
3.12	eERM Ereignis und -vorlage	45
3.13	eERM Vorgang und Ereignis	45
3.14	eEPK Funktion	46
3.15	Ereignisgraph	48
3.16	Projekt Garage - Leistungsgegenstandsvorlagen	51
3.17	Projekt Garage - Ereignisvorlagen und Vorgangsvorlagen	53
3.18	Projekt Garage - Leistungsprozessvorlagen	55
3.19	Prinzip Projektstrukturbaum	56
3.20	Projekt Garage - Projektstrukturbaum	57
3.21	Projekt Garage - Leistungsprozesse	58
3.22	Projekt Garage - Voraussetzungen	59
3.23	Projekt Garage - Gesamtprozess als eEPK	61
4.1	Projekt Garage - Ereignisgraph	70
4.2	Ein Teilgraph von G_E , bestehend aus drei Untergraphen	72
4.3	Beispielgraph für die Mustererkennung	81
4.4	Ausschnitt eines Ereignisgraphen	86

4.5	Komponenten, nach Muster sortiert	87
4.6	Auflösen von Überschneidungen	87
4.7	Komponentenkombinationen	88
4.8	Selektion einer Komponentenkombination	90
4.9	Prinzip des Entscheidungsbaums	91
4.10	Entscheidungsbaum	92
4.11	Verzweigen	95
4.12	Wiederkehrende Kombinationen auf unterschiedlichen Pfaden	97
4.13	Einführung minimaler Komponenten	101
4.14	Kontext um Knoten mit gleicher Beschriftung	103
4.15	Bereinigte Kontextkomponenten	105
4.16	Minimale Komponenten	105
4.17	Entwickeln der Vorgängerkomponenten	110
4.18	Entwickeln der Nachfolgerkomponenten	111
4.19	Komponenten nach Muster	113
4.20	Komponentenkombinationen für Muster d	116
4.21	Komponentenkombinationen nach Muster	117
4.22	Komponentenkombinationen	119
4.23	Normierte Bewertung der Komponentenkombinationen	121
4.24	Verlauf des Algorithmus	125
4.25	Optimale Lösungen	128
5.1	Leistungsprozessvorlagen	131
5.2	Prozessmodell - Generierter Gesamtprozess ohne Voraussetzungen	133
5.3	Prozessmodell - Spezifizierter Gesamtprozess	134
5.4	Prozessmodell - Minimale Komponenten	136
5.5	Verlauf der Suche	139
5.6	Beispiel Wände - Ergebnis der Mustererkennung	143
5.7	Vergleich von Komponenten	147
5.8	Beispiel 2 - Spezifiziertes Prozessmodell ω_{spez}	151
5.9	Beispiel 2 - Minimale Komponenten ω_{min}	151
5.10	Beispiel 2 - Optimale Zerlegung und Verlauf des Verfahrens	152
5.11	Gegenüberstellung von Beispiel 1 und Beispiel 2	154

Symbolverzeichnis

a	Bewertung eines Ereignisgraphen. Maß für den Aufwand ohne Berücksichtigung von gleichartigen Teilprozessen	77
\ddot{A}_K	Enthält die Äquivalenzrelation ein Paar $(M_{E,k}, M_{E,j})$, so lassen sich die Komponenten $M_{E,k}$ und $M_{E,j}$ auf dasselbe Muster abbilden	76
a_ω	Bewertung einer Prozessmodells. Maß für den Aufwand, unter Berücksichtigung wiederkehrender Teilprozesse und der zusätzlich zu spezifizierenden Kanten	79
$a_{\mathbf{w}\Omega}^{erf}$	verbleibender erforderlicher Aufwand	99, 100
$a_{\omega,opt}$	minimaler Wert, nach der Auswertung der Abbildung $a_\omega(\omega)$ für alle Elemente von Ω	81
$a_{\mathbf{w}\Omega}^{verf}$	verbleibender nicht ausgeschöpfter Aufwand oder auch verbleibender verfügbarer Aufwand	99
$a_{\mathbf{w}\Omega}$	ausgeschöpfter Aufwand für den Vektor $\mathbf{w}_{\Omega,x,y}$	98
A_w	Aufwandswert	42, 43
a_Z	Bewertung einer Zerlegung. Maß für den Aufwand, unter Berücksichtigung sich wiederholender Teilprozesse	78
C_K	Menge an Komponentenkombinationen	85
$C_{M_{K,i}^p}$	eine Komponentenkombination: Menge an überschneidungsfreien Komponenten gleichen Musters	85

e_i	Ereignis	45
e_i^v	Ereignisvorlage	45
f_i	Funktion	46
f_I	Abbildung einer Zerlegung auf einen Identifikator	74
f_P	Abbildung eines Ereignisgraphen auf ein Muster	75
G	schlichter Graph	69
G_E	Ereignisgraph	69
K_E	Komponente eines Ereignisgraphen G_E	72
lg_i	Leistungsgegenstand	38
lg_i^v	Leistungsgegenstandsvorlage	38, 51
lp_i	Leistungsprozess	39
lp_i^v	Leistungsprozessvorlage	40, 62, 63
L_w	Leistungswert	42
M_E	Menge an Ereignissen	69
M_F	Menge an Funktionen	69
M_K	Menge an Komponenten einer Zerlegung	72
M_K^p	Menge an Komponenten, für die das Muster gleich ist	84
M_R	Repräsentantensystem von M_K , das genau eine Komponente aus jeder Äquivalenzklasse enthält	78
n_Z	normierte Bewertung: ein Maß, das den relativen Effekt beschreibt, der sich aus der Berücksichtigung wiederkehrender Teilprozesse ergibt	78

$n_{Z,min}$	minimale normierte Bewertung	96
Ω	Menge von Prozessmodellen	60, 73
ω	Prozessmodell	60, 73
Ω_{opt}	Menge an Prozessmodellen, für die die Bewertung $a_\omega(\omega)$ minimal ist	81
R_{E-E}	Menge der Kanten eines Ereignisgraphen	69
R_{E-F}	Menge an Verknüpfungen, die ein Ereignis als Voraussetzung für eine Funktion definieren	69
R_{F-E}	Menge an Verknüpfungen, die ein Ereignis als Ergebnis für eine Funktion definieren	69
S_{E-E}	Menge an Kanten, die unabhängige Komponenten in einen Zusammenhang bringen	71
$t_{N,x}$	Menge der Nachfolger von x	102
$t_{V,x}$	Menge der Vorgänger von x	102
v_i	Vorgang	42, 43
vp_i	Vorgangsposition	42, 43
vp_i^v	Vorgangspositionvorlage	43
v_i^v	Vorgangsvorlage	43
\mathbf{w}	boolescher Vektor, der ein Prozessmodell beschreibt	82
\mathbf{w}_Ω	boolescher Vektor, der eine Menge an Prozessmodellen beschreibt	91
\mathbf{w}_ω	boolescher Vektor, der ein Prozessmodell beschreibt	91
z_i	Zustand	45
Z_E	Zerlegung eines Ereignisgraphen G_E in Komponenten	71
Z_{E-E}	Komplement der Menge S_{E-E} bezüglich R_{E-E}	71

176

z_i^v Zustandsvorlage

45

Abkürzungsverzeichnis

AHO	Ausschuss der Ingenieurverbände für die Honorarordnung e.V.	18
ARH	Arbeitszeit-Richtwerte Hochbau	42
ARIS	Architektur integrierter Informationssysteme	31–33
BRMS	Business Rule Management Systeme	24
CBR	Case-Based-Reasoning	24
CPM	Critical Path Method	22
eEPK	erweiterte ereignisgesteuerte Prozesskette	33, 46–48, 54, 60
eERM	erweitertes Entity-Relationship-Model	32, 44, 46, 47
EKN	Ereignisknoten-Netzplan	22
EPK	ereignisgesteuerte Prozesskette	29
ERM	Entity-Relationship-Model	32
GPS	General Problem Solver	23
HOAI	Verordnung über die Honorare für Leistungen der Architekten und Ingenieure	18
KI	künstliche Intelligenz	23
MBR	Model-Based-Reasoning	26
MPM	Metra Potential Method	22

PERT	Program Evaluation and Review Technique	22
PSP	Projektstrukturplan	54, 56
STLB	Standardleistungsbuch für das Bauwesen	52
VKN	Vorgangsknoten-Netzplan	22
VPN	Vorgangspfeil-Netzplan	22

Anhang A

Begriffe aus der Graphentheorie

Die in diesem Abschnitt zusammengefassten Grundbegriffe und Definitionen dienen der Abgrenzung, um ein einheitliches Verständnis zu schaffen. Es besteht nicht der Anspruch, die zugrunde liegende Algebra axiomatisch herzuleiten. Der Betrachtungsweise dieser Arbeit liegen die mathematischen Grundlagen von Pahl und Damrath [2000] zugrunde.

A.1 Mengen

Menge: Der Begriff Menge wird verstanden als eine Zusammenfassung bestimmter, wohlunterscheidbarer Objekte unserer Anschauung zu einem Ganzen. Die Objekte einer Menge werden als Elemente bezeichnet. Diese Formulierung geht auf den Mathematiker Georg Cantor¹ zurück. Eine Menge wird spezifiziert durch Aufzählen der Elemente oder durch die Formulierung einer Bildungsregel.

$$\begin{aligned} M &= \{a, b, c\} \\ M &= \{x \mid x \text{ ist eine natürliche Zahl}\} \end{aligned} \tag{A.1}$$

In einer Menge existiert keine Reihenfolge. Eine Menge kann leer, endlich oder unendlich sein. In einer Menge kann ein Element nur einmal auftreten. Ist ein Objekt x ein Element einer Menge M , so wird dies ausgedrückt durch $x \in M$. Die Negation lautet $x \notin M$. Eine Menge kann aus Elementen bestehen, die selbst Mengen sind. Handelt es sich bei den Elementen einer Menge M ausschließlich um Mengen, so beschreibt die Menge M ein

¹ Georg Ferdinand Ludwig Philipp Cantor (*1845-1918). Cantor gilt als der Begründer der Mengenlehre.

Mengensystem. Im Rahmen dieser Arbeit sind die betrachteten Mengen endlich.

Eigenschaften von Mengen werden über Ausdrücke formuliert. Die Auswertung eines Ausdrucks für ein Element einer Menge liefert entweder wahr oder falsch. Für die Verknüpfung von Ausdrücken und den Elementen einer Menge werden Quantoren verwendet.

$$\begin{aligned} \bigwedge_{x \in M} \dots & \quad \text{für alle } x \text{ gilt } \dots \\ \bigvee_{x \in M} \dots & \quad \text{es existiert ein } x, \text{ für das } \dots \text{ gilt} \end{aligned} \quad (\text{A.2})$$

Der Quantor \bigwedge gibt an, dass ein Ausdruck für alle Elemente einer Menge gilt. Der Quantor \bigvee gibt an, dass ein Ausdruck für mindestens ein Element einer Menge gilt. Erschließt sich der Bezug zu der Menge aus dem Kontext, kann auf seine Angabe unterhalb des Quantors verzichtet werden.

Verknüpfungen: Eine Mengenverknüpfung definiert im Allgemeinen auf der Grundlage der Aussagenlogik wie sich aus zwei gegebenen Mengen eine dritte Menge berechnen lässt. Die speziellen Mengenverknüpfungen Gleichheit und Inklusion dagegen liefern als Ergebnis die logische Konstante wahr oder falsch. Die Gleichheit von endlichen Mengen kann durch den Vergleich der enthaltenen Elemente ermittelt werden. Zwei Mengen A und B sind gleich, wenn sie dieselben Elemente enthalten.

$$A = B \quad :\Leftrightarrow \quad \bigwedge_x (x \in A \Leftrightarrow x \in B) \quad (\text{A.3})$$

Inklusion beschreibt Teilmengen. Sind die Elemente einer Menge A vollständig in einer Menge B enthalten, dann ist A eine Teilmenge von B . Existiert in der Menge B darüber hinaus mindestens ein Element, das nicht in A enthalten ist, so ist A eine echte Teilmenge von B .

$$\text{Teilmenge:} \quad A \subseteq B \quad :\Leftrightarrow \quad \bigwedge_x (x \in A \Rightarrow x \in B) \quad (\text{A.4})$$

$$\text{Echte Teilmenge:} \quad A \subset B \quad :\Leftrightarrow (A \subseteq B) \wedge (A \neq B) \quad (\text{A.5})$$

Ausgehend von den Mengen A und B werden die Verknüpfungen Durchschnitt, Vereinigung und Differenz definiert, deren Ergebnis eine neue Menge ist.

$$\text{Durchschnitt:} \quad A \cap B \quad := \quad \{x \mid x \in A \wedge x \in B\} \quad (\text{A.6})$$

$$\text{Vereinigung:} \quad A \cup B \quad := \quad \{x \mid x \in A \vee x \in B\} \quad (\text{A.7})$$

$$\text{Differenz:} \quad A \setminus B \quad := \quad \{x \mid x \in A \wedge x \notin B\} \quad (\text{A.8})$$

Spezielle Mengen: Eine Menge die keine Elemente enthält wird als leere Menge bezeichnet.

$$\text{Leere Menge: } \emptyset := \{\} \quad (\text{A.9})$$

Fasst man alle Teilmengen, die sich auf der Grundlage einer Menge M bilden lassen, zu einem Mengensystem zusammen, so erhält man die zu einer Menge M gehörige Potenzmenge. Die Potenzmenge enthält die Mengen M und \emptyset .

$$\text{Potenzmenge: } P(M) \quad (\text{A.10})$$

Sei A eine Teilmenge von B ist. In diesem Fall, stellt die Differenzmenge $B \setminus A$ das Komplement von A bezüglich B dar und wird mit \bar{A} bezeichnet.

$$\begin{aligned} \text{Komplement: } \bar{A} &:= B \setminus A \\ A \cap \bar{A} &= \emptyset \quad \wedge \quad A \cup \bar{A} = B \end{aligned} \quad (\text{A.11})$$

Führt die Schnittmenge zweier Mengen A und B zu einer leeren Menge, so werden die verknüpften Mengen als disjunkte Mengen bezeichnet.

$$\begin{aligned} \text{Disjunkte Mengen: } A \cap B &= \emptyset \\ A \text{ und } B &\text{ sind disjunkt} \end{aligned} \quad (\text{A.12})$$

Sind die Elemente eines Mengensystems paarweise disjunkt, so wird das Mengensystem als disjunkt bezeichnet. Bildet man auf der Grundlage einer Menge M ein disjunktes Mengensystem, so erhält man eine Zerlegung für M in n Teilmengen T_i . Für eine Zerlegung gilt: Jedes Element von M ist Element genau einer Teilmenge T_i , keine Teilmenge T_i ist leer und die Vereinigungsmenge aller Teilmengen T_i ist die Menge M .

$$\text{Zerlegung: } M = \bigcup_{i=1}^n T_i \quad :\Leftrightarrow \quad \bigwedge_i \bigwedge_j (i = j \vee T_i \cap T_j = \emptyset) \quad (\text{A.13})$$

Mengen algebra: Die Mengenalgebra definiert für Mengen und ihre Verknüpfungen Rechenregeln. Die Rechenregeln folgen aus den Definitionen der Mengenverknüpfungen und den Auswertungsvorschriften für Junktoren der Aussagenlogik. Zulässige algebraische Ausdrücke ergeben sich aus der Kombination von Mengenausdrücken. Der Mengenwert M stellt einen Mengenausdruck dar. Das Komplement nach Gleichung A.11 ist ein Mengenausdruck und die Mengenverknüpfungen nach den Gleichungen A.6 - A.8 sind Mengenausdrücke. Für eine ausführlichere Beschreibung der Mengenalgebra sei auf die Literatur Pahl und Damrath [2000] verwiesen.

A.2 Unäre Relationen

Unäre Relation: Gegeben sei eine nichtleere Menge M . Jede Teilmenge von M lässt sich ausdrücken als unäre Relation in M . Sie beschreibt die Zugehörigkeit eines Elementes aus M zu einer Teilmenge. Die Zugehörigkeit wird ausgedrückt über die Verknüpfung einer Aussage R mit einem Element x der Menge M . Die Auswertung der Verknüpfung liefert wahr oder falsch. Diejenigen Elemente $x \in M$, für die die Auswertung des Ausdruckes Rx wahr ist, werden als zugehörig definiert.

$$\text{Unäre Relation: } u := \{x \in M \mid Rx\} \subseteq M \quad (\text{A.14})$$

Eine unäre Relation u lässt sich darstellen als boolescher Vektor \mathbf{u} . Jedem Koeffizienten u_i ist einem Element x_i der Menge M zugeordnet. Der Koeffizient selbst enthält die Auswertung des Ausdruckes Rx_i . Die Wahrheitswerte seien $W = \{0, 1\}$.

$$\mathbf{u} = \begin{array}{c} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{array} \begin{array}{c} u_1 \\ \vdots \\ u_i \\ \vdots \\ u_n \end{array} \quad \begin{array}{l} x_i \in M \Rightarrow u_i = 1 \\ x_i \notin M \Rightarrow u_i = 0 \end{array}$$

Spezielle unäre Relationen: Ist eine Teilmenge leer, dann wird sie als Nullrelation bezeichnet. Enthält eine Teilmenge genau ein Element, dann wird sie als Punktrelation bezeichnet. Enthält eine Teilmenge alle Elemente der ursprünglichen Menge M , dann wird sie als Allrelation bezeichnet.

$$\text{Nullrelation: } \emptyset := \{\} \quad (\text{A.15})$$

$$\text{Punktrelation: } x := \{x\} \quad (\text{A.16})$$

$$\text{Allrelation: } e := M \quad (\text{A.17})$$

$$\emptyset = \begin{array}{c} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{array} \begin{array}{c} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{array} \quad \mathbf{x} = \begin{array}{c} x_1 \\ \vdots \\ x = x_i \\ \vdots \\ x_n \end{array} \begin{array}{c} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{array} \quad \mathbf{e} = \begin{array}{c} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{array} \begin{array}{c} 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \end{array}$$

Verknüpfungen: Die unären Verknüpfungen Gleichheit und Inklusion liefern für unäre Relationen als Ergebnis die logische Konstante wahr oder falsch. Die Gleichheit drückt aus, dass zwei Teilmengen u und v einer Menge M gleich sind. Die Inklusion beschreibt, dass eine Teilmenge u von M in einer zweiten Teilmenge v von M enthalten ist.

$$\text{Gleichheit: } u = v \quad :\Leftrightarrow \bigwedge_x (x \in u \Leftrightarrow x \in v) \quad (\text{A.18})$$

$$\text{Inklusion: } u \sqsubseteq v \quad :\Leftrightarrow \bigwedge_x (x \in u \Rightarrow x \in v) \quad (\text{A.19})$$

Das Komplement einer unären Relation u stellt eine unäre Verknüpfung dar. Es enthält genau diejenigen Elemente von M , die nicht in u enthalten sind.

$$\text{Komplement: } \bar{u} \quad := \quad \{x \in M \mid x \notin u\} \quad (\text{A.20})$$

Binäre Verknüpfungen verlangen zwei gegebene unäre Relationen. Definiert werden die Verknüpfungen Durchschnitt \sqcap und Vereinigung \sqcup . Gegeben seien die Relationen u und v . Sie seien Teilmengen von M .

$$\text{Durchschnitt: } u \sqcap v \quad := \quad \{x \mid x \in u \wedge x \in v\} \subseteq M \quad (\text{A.21})$$

$$\text{Vereinigung: } u \sqcup v \quad := \quad \{x \mid x \in u \vee x \in v\} \subseteq M \quad (\text{A.22})$$

Unäre Relationenalgebra: Unäre Relationen stellen Teilmengen einer Menge M dar. Die Potenzmenge $P(M)$ umfasst alle voneinander verschiedenen Teilmengen von M . Die Potenzmenge $P(M)$ enthält die Nullrelation und die Allrelation. Auf der Grundlage der Potenzmenge $P(M)$ und den beschriebenen Verknüpfungen lässt sich für endliche Mengen eine boolesche Vektoralgebra definieren, für die alle Eigenschaften und Rechenregeln aus der Mengenalgebra gelten. Für eine ausführliche Betrachtung der Eigenschaften und der Rechenregeln sei auf die Literatur Pahl und Damrath [2000] verwiesen.

A.3 Binäre Relationen

Binäre Relation: Besteht zwischen Elementen einer oder mehrerer Mengen eine Beziehung, so kann diese durch ein n -Tupel, das die n -Elemente der jeweiligen Mengen umfasst, zum Ausdruck gebracht werden. Durch die Angabe einer Semantik wird die Bedeutung einer Beziehung beschrieben. Fasst man gleichartige n -Tupel mit gleicher Semantik zu einer Menge zusammen, so bezeichnet man diese Menge als Relation. Relationen, für

die die Reihenfolge der Elemente im n -Tupel unbedeutend ist, werden als ungerichtete Relationen bezeichnet. Das n -Tupel hat den Charakter einer Menge mit n Elementen. Ist die Reihenfolge der Elemente im n -Tupel für die Relation von Bedeutung, so wird die Relation als gerichtet bezeichnet. Für die Festlegung der Reihenfolge nutzt man das Konzept des geordneten Paares. Ein geordnetes Paar besteht aus einer Menge von zwei Komponenten. Eine Komponente wird gegenüber der zweiten Komponente hervorgehoben.

$$\begin{aligned} \text{Geordnetes Paar: } (a, b) &:= \{\{a\}, \{a, b\}\} \\ a &\text{ erste Komponente} \\ b &\text{ zweite Komponente} \end{aligned} \quad (\text{A.23})$$

Die Komponenten a und b können Elemente sein, sie können aber auch selbst Mengen sein. Durch rekursives Verschachteln geordneter Paare ist die Festlegung einer Reihenfolge für n -Tupel möglich, für die $n > 2$ ist. Bestehen die n -Tupel einer Relation aus genau zwei Elementen, so bezeichnet man die Relation als binäre Relation. Die nachfolgenden Ausführungen beziehen sich auf binäre Relationen. Diese bilden die Grundlage für die später folgenden graphentheoretischen Betrachtungen. Wenn nicht ausdrücklich etwas anderes angegeben ist, wird im Rahmen dieser Arbeit unter einer Relation eine binäre Relation verstanden.

Die Menge der geordneten Paare, die auf der Grundlage zweier Mengen A und B gebildet werden können, wird durch das kartesische Produkt ausgedrückt.

$$\text{Kartesisches Produkt: } A \times B := \{(a, b) \mid a \in A \wedge b \in B\} \quad (\text{A.24})$$

Eine Relation stellt eine Teilmenge des kartesischen Produktes dar. Die Teilmenge wird durch die Angabe einer Semantik festgelegt. Der Ausdruck aRb , sprich a steht in Relation zu b , beschreibt die Semantik einer Beziehung. Seine Auswertung liefert wahr oder falsch. Eine Relation umfasst diejenigen Elemente des kartesischen Produktes, für die dieser Ausdruck wahr ist. Entstammen die Elemente a und b derselben Menge, dann bezeichnet man die Relation als homogen.

$$\begin{aligned} \text{Binäre Relation:} \\ R &:= \{(a, b) \in A \times B \mid aRb\} \subseteq A \times B \\ \text{Homogene binäre Relation:} \\ R &:= \{(a, b) \in M \times M \mid aRb\} \subseteq M \times M \end{aligned} \quad (\text{A.25})$$

Eine binäre Relation R lässt sich darstellen als boolesche Matrix \mathbf{R} . Jedem Koeffizienten r_{ij} ist ein geordnetes Paar (a_i, b_j) des kartesischen Produktes $A \times B$ zugeordnet. Der Koeffizient selbst enthält die Auswertung des Ausdrucks $a_i R b_j$. Die Wahrheitswerte seien $W = \{0, 1\}$.

$$\mathbf{R} = \begin{array}{c} \begin{array}{c} a_1 \\ \vdots \\ a_i \\ \vdots \\ a_n \end{array} \begin{array}{|c|c|c|c|} \hline b_1 & & b_j & \\ \hline r_{11} & \dots & r_{1j} & \dots & r_{1m} \\ \hline & \ddots & & & \\ \hline r_{i1} & & r_{ij} & & \\ \hline & & & \ddots & \\ \hline r_{n1} & & & & r_{nm} \\ \hline \end{array} \end{array} \quad \begin{array}{l} (a_i, b_j) \in A \times B \Rightarrow r_{ij} = 1 \\ (a_i, b_j) \notin A \times B \Rightarrow r_{ij} = 0 \end{array}$$

Die Eigenschaften einer Relation ergeben sich aus der Auswertung der Ausdrücke aRb und bRa . Nachfolgend werden diejenigen Eigenschaften genannt, auf die im Rahmen dieser Arbeit Bezug genommen wird.

$$R := \{(a, b) \in A \times B \mid aRb\} \subseteq A \times B$$

$$R \text{ ist reflexiv} \quad :\Leftrightarrow \bigwedge_a (aRa) \quad (\text{A.26})$$

$$R \text{ ist symmetrisch} \quad :\Leftrightarrow \bigwedge_a \bigwedge_b (aRb \Rightarrow bRa) \quad (\text{A.27})$$

$$R \text{ ist asymmetrisch} \quad :\Leftrightarrow \bigwedge_a \bigwedge_b (aRb \Rightarrow \neg bRa) \quad (\text{A.28})$$

$$R \text{ ist transitiv} \quad :\Leftrightarrow \bigwedge_a \bigwedge_b \bigwedge_c (aRb \wedge bRc \Rightarrow aRc) \quad (\text{A.29})$$

$$R \text{ ist linkstotal} \quad :\Leftrightarrow \bigwedge_a \bigvee_b (aRb) \quad (\text{A.30})$$

$$R \text{ ist rechtstotal} \quad :\Leftrightarrow \bigwedge_b \bigvee_a (aRb) \quad (\text{A.31})$$

$$R \text{ ist bitotal} \quad :\Leftrightarrow R \text{ ist linkstotal} \\ \wedge \text{ rechtstotal} \quad (\text{A.32})$$

$$R \text{ ist linkseindeutig} \quad :\Leftrightarrow \bigwedge_a \bigwedge_b \bigwedge_c (aRb \wedge bRc \Rightarrow a = c) \quad (\text{A.33})$$

$$R \text{ ist rechtseindeutig} \quad :\Leftrightarrow \bigwedge_a \bigwedge_b \bigwedge_c (aRb \wedge aRc \Rightarrow b = c) \quad (\text{A.34})$$

$$\begin{aligned}
 R \text{ ist eineindeutig} & \quad :\Leftrightarrow R \text{ ist linkseindeutig} \\
 & \quad \wedge \text{ rechtseindeutig} \qquad \qquad \qquad (A.35)
 \end{aligned}$$

Spezielle homogene binäre Relationen: Ist die Menge der geordneten Paare einer homogene binäre Relation leer, dann wird sie als Nullrelation bezeichnet. Besteht eine homogene binäre Relation nur aus geordneten Paaren, deren Elemente gleich sind, dann wird sie als Identitätsrelation bezeichnet. Enthält eine homogene binäre Relation für jedes Element des kartesischen Produktes ein geordnetes Paas, dann wird die Relation als Allrelation bezeichnet.

$$\text{Nullrelation:} \quad \emptyset := \{ \} \subseteq M \times M \qquad (A.36)$$

$$\text{Identitätsrelation: } I := \{ (x, x) \mid x \in M \} \subseteq M \times M \qquad (A.37)$$

$$\text{Allrelation:} \quad E := \{ (x, y) \mid x \in M \wedge y \in M \} \subseteq M \times M \quad (A.38)$$

Verknüpfungen: Die binären Verknüpfungen Gleichheit und Inklusion liefern als Ergebnis die logische Konstante wahr oder falsch. Die Gleichheit drückt aus, dass die Menge der geordneten Paare zweier Relationen R_A und R_B gleich sind. Die Inklusion beschreibt, dass die geordneten Paare einer Relation R_A in einer Relation R_B enthalten sind. R_A und R_B seien Teilmengen von $M \times M$.

Gleichheit:

$$R_A = R_B \quad :\Leftrightarrow \bigwedge_x \bigwedge_y ((x, y) \in R_A \Leftrightarrow (x, y) \in R_B) \subseteq M \times M \quad (A.39)$$

Inklusion:

$$R_A \subseteq R_B \quad :\Leftrightarrow \bigwedge_x \bigwedge_y ((x, y) \in R_A \Rightarrow (x, y) \in R_B) \subseteq M \times M \quad (A.40)$$

Die weiteren Verknüpfungen werden unterschieden in unäre und binäre Verknüpfungen. Sie definieren, wie sich aus gegebenen Relationen neue Relationen ableiten lassen. Unäre Verknüpfungen verlangen eine gegebene Relation. Das Komplement und die Transponierte stellen unäre Verknüpfungen dar. Das Komplement einer homogenen binären Relation besteht aus genau der Menge an geordneten Paaren, die sich ergibt, wenn man die Differenzmenge der geordneten Paare aus der Allrelation und der Relation bildet. Die Transponierte ergibt sich durch Umkehren der Reihenfolge der Elemente für jedes geordnete Paar einer homogenen binären

Relation.

Komplement:

$$\bar{R} := \{(x, y) \in M \times M \mid (x, y) \notin R\} \subseteq M \times M \quad (\text{A.41})$$

Transponierte:

$$R^T := \{(y, x) \in M \times M \mid (x, y) \in R\} \subseteq M \times M \quad (\text{A.42})$$

Binäre Verknüpfungen verlangen zwei gegebene Relationen. Definiert werden die Verknüpfungen Durchschnitt \sqcap , Vereinigung \sqcup und Kettung \circ . Diese sind für die Formulierung einer Wegalgebra erforderlich. Gegeben seien die Relationen R und S . R und S seien Teilmengen von $M \times M$. Der Durchschnitt der Relationen R und S berechnet sich als Schnittmenge der geordneten Paare von R und S .

Durchschnitt:

$$R \sqcap S := \{(x, y) \mid (x, y) \in R \wedge (x, y) \in S\} \subseteq M \times M \quad (\text{A.43})$$

Die Vereinigung von R und S berechnet sich als Vereinigungsmenge der geordneten Paare von R und S .

Vereinigung:

$$R \sqcup S := \{(x, y) \mid (x, y) \in R \vee (x, y) \in S\} \subseteq M \times M \quad (\text{A.44})$$

Gegeben sei die Relation R auf den Mengen A und B , sowie die Relation S auf den Mengen B und C . Existiert in R ein geordnetes Paar (a, b) und gleichzeitig in der Relation S ein geordnetes Paar (b, c) , wobei b in beiden Paaren dasselbe Element darstellt, dann enthält die Kettung das geordnete Paar (a, c) . Die Kettung wird auch als das Produkt bezeichnet. Als Verknüpfungssymbol wird daher \circ verwendet.

Kettung:

$$R \circ S := \{(a, c) \mid \bigvee_b ((a, b) \in R \wedge (b, c) \in S)\} \subseteq A \times C \quad (\text{A.45})$$

Bildet man das Produkt einer homogenen binären Relation R mit sich selbst, dann stellen die in der Relation $R \circ R$ enthaltenen geordneten Paare anschaulich betrachtet einen Weg dar. Eingesetzt in Gleichung A.45 lässt sich dieser Weg als ein Weg von einem Element a vorbei an einem Element b hin zu dem Element c beschreiben. Die Relation $R \circ R$ enthält alle Wege der Länge 2, für die es in R ein Element b gibt, das die Elemente a und c miteinander verknüpft. Die Relation $R \circ R$ wird auch als Potenz R^2 ausgedrückt. Durch weiteres Heranmultiplizieren lassen sich höherwertige Potenzen n von R erzeugen. Der Wert der Potenz n gibt die Länge des Weges bezogen auf die in R enthaltenen Wege der Länge $n = 1$ an.

Hülle: Eine Erweiterung der Definition einer Relation führt zu der Definition der Hülle. Die Hülle einer Relation ist eine Teilmenge ihres kartesischen Produktes. Hüllen werden dazu verwendet, Relationen um diejenigen geordneten Paare zu ergänzen, so dass sie die formalen Bedingungen für gewünschte Eigenschaften erfüllen. Der Bezeichner einer Hülle besteht aus dem Namen der Relation, eingefasst in spitzen Klammern. Ein Index weist auf die Eigenschaft hin, die der Bildungsregel zugrunde lag.

$$\begin{aligned} \text{Reflexive Hülle:} \quad \langle R \rangle_r &:= \{(x, y) \mid (x, y) \in R \vee x = y \in M\} \\ \langle R \rangle_r &= I \sqcup R \end{aligned} \quad (\text{A.46})$$

$$\begin{aligned} \text{Symmetrische Hülle:} \quad \langle R \rangle_s &:= \{(x, y) \mid (x, y) \in R \vee (y, x) \in R\} \\ \langle R \rangle_s &= R \sqcup R^T \end{aligned} \quad (\text{A.47})$$

Eine wichtige Rolle für die weiteren Betrachtungen spielt die transitive Hülle. Sie ergibt sich aus der Vereinigung der Relationen R^n für die Potenzen $n = 1 \dots s$. Ändert sich in Schritt $s + 1$ die Vereinigungsmenge nicht mehr, so wird s als Stabilitätsindex bezeichnet. Die transitive Hülle enthält alle Wege, von einem Element x zu einem Element y über beliebig viele Zwischenstationen. Wegen ihrer Sonderrolle haben die folgenden Hüllen verkürzte Schreibweisen.

$$\begin{aligned} \text{Transitive Hülle:} \quad \langle R \rangle_t &:= \{(x, y) \mid (x, y) \in R \vee x = y \in M\} \\ R^+ &:= \langle R \rangle_t = \bigsqcup_{i=1}^s R^i \end{aligned} \quad (\text{A.48})$$

$$\text{Reflexiv-transitive Hülle: } R^* := \langle R \rangle_{rt} = I \sqcup R^+ \quad (\text{A.49})$$

Struktur: Eine Menge M ist nach ihrer Definition eine unstrukturierte Zusammenfassung von Elementen. Sie kann auf der Grundlage einer Relationen R strukturiert werden. Werden gleichzeitig unterschiedliche Relationen betrachtet, dann sind Regeln erforderlich, die ihre Verträglichkeit sicherstellen. Eine Struktur beschreibt ein Gebilde $(M; R)$, für die dieselben Regeln gelten.

$$\begin{aligned} \text{Struktur:} \quad &(M; R) && (\text{A.50}) \\ &M && \text{eine oder mehrere Mengen} \\ &R && \text{eine oder mehrere Relationen} \end{aligned}$$

Gehören mehrere Mengen oder Relationen zu einer Struktur, so werden die Mengen und die Relationen untereinander durch Kommas getrennt und der Übergang von den Mengen zu den Relationen durch ein Semikolon (z.B. $G = (P(M); \sqcap, \sqcup, \sqsupset)$). Beschreibt jede Relation aus R eine Verknüpfung in M auf Elementbasis, so handelt es sich um eine algebraische Struktur.

Abbildung: Eine Abbildung ist eine Relation $f \subseteq A \times Z$, für die bestimmte Eigenschaften zutreffen. Sie ist linkstotal und rechtseindeutig. Dies bedeutet, dass jedes Element a einer Ausgangsmenge A über eine Abbildungsvorschrift f eindeutig auf ein Element z einer Zielmenge Z abgebildet wird. Auf Elementebene können folgende Ausdrücke äquivalent verwendet werden: $f : a \rightarrow z \Leftrightarrow f(a) = z$. f bildet das Element $a \in A$ auf das Abbild $z \in Z$ ab.

$$\begin{array}{ll} \text{Abbildung: } f : A \rightarrow Z & f \text{ bildet } A \text{ auf } Z \text{ ab.} \\ A & \text{Ausgangsmenge} \\ Z & \text{Zielmenge, Abbildungsmenge} \end{array} \quad (\text{A.51})$$

$$f \text{ ist eine Abbildung} \quad :\Leftrightarrow f \text{ ist linkstotal} \wedge \text{ rechtseindeutig}$$

Bei einer Abbildung können die Elemente der Zielmenge auch mehrmals als Abbild auftreten. Fasst man die durch f adressierten Abbilder z in einer Menge S zusammen, so ist S eine Teilmenge von Z . Ist S eine echte Teilmenge von Z , also $S \subset Z$, dann wird die Abbildung f als injektiv bezeichnet. Die Anzahl der Elemente in A und in S ist gleich. Die Anzahl der Elemente in S ist kleiner als in Z .

$$f \text{ ist injektiv} \quad :\Leftrightarrow f \text{ ist linkstotal} \wedge \text{ eineindeutig} \quad (\text{A.52})$$

Ist $S = Z$ und ist die Anzahl der Elemente in A und in Z unterschiedlich, dann wird die Abbildung als surjektiv bezeichnet.

$$f \text{ ist surjektiv} \quad :\Leftrightarrow f \text{ ist bitotal} \wedge \text{ rechtseindeutig} \quad (\text{A.53})$$

Ist $S = Z$ und ist die Anzahl der Elemente in A und in Z gleich, dann wird die Abbildung als bijektiv bezeichnet.

$$f \text{ ist bijektiv} \quad :\Leftrightarrow f \text{ ist bitotal} \wedge \text{ eineindeutig} \quad (\text{A.54})$$

Gegeben seien die geordneten Paare (A, R_a) und (Z, R_z) , die Teil einer Struktur sind. Eine Abbildung $f : A \rightarrow Z$ wird als strukturerhaltend bezeichnet, wenn folgender Ausdruck wahr ist: $f(a_1 R_a a_2) = f(a_1) R_z f(a_2)$.

Ist eine Abbildung strukturerhaltend und bijektiv, dann bezeichnet man die Struktur als isomorph.

$$f \text{ ist isomorph} \quad :\Leftrightarrow \quad f \text{ ist strukturerhaltend} \wedge \text{ bijektiv} \quad (\text{A.55})$$

Binäre Relationenalgebra: Gegeben sei eine Menge M . Aus dem kartesischen Produkt $M \times M$ ergibt sich die Menge der geordneten Paare, die aus den Elementen von M gebildet werden können. Die Potenzmenge $P(M \times M)$ umfasst all diejenigen Relationen, deren Menge an geordneten Paaren eine Teilmenge von $M \times M$ ist. Die Potenzmenge $P(M \times M)$ enthält die Nullrelation und die Allrelation. Auf der Grundlage der Menge M , der Potenzmenge $P(M \times M)$ und den beschriebenen Verknüpfungen lässt sich eine Algebra definieren. Diese Algebra gilt für homogene binäre Relationen. Für endliche Mengen M führt dies zu einer booleschen Matrixalgebra. Die Eigenschaften und Rechenregeln dieser Algebra ergeben sich aus der Definition ihrer Verknüpfungen und den Verträglichkeitsbedingungen der Verknüpfungen untereinander. Für eine ausführliche Betrachtung der Eigenschaften und der Rechenregeln sei auf die Literatur Pahl und Damrath [2000] verwiesen.

A.4 Graphen

Ein Graph ist ein Knoten-Kanten-Diagramm, das der visuellen Veranschaulichung von Beziehungen zwischen Elementen dient. Die Elemente werden als Knoten dargestellt und die Beziehungen werden als Kanten dargestellt. Begrifflichkeiten und Eigenschaften werden oft nach ihrem visuellen Erscheinen bzw. nach ihren visuellen Eigenschaften benannt. In diesem Abschnitt wird auf die Grundbegriffe der Graphentheorie eingegangen, die im Rahmen dieser Arbeit von Bedeutung sind. Den Betrachtungen liegt die Literatur von Pahl und Damrath [2000] und Clark und Holton [1994] zugrunde. Wenn bei der Verwendung des Begriffes *Graph* nicht ausdrücklich differenziert wird, dann wird auf schlichte Graphen Bezug genommen.

Graph: Ein Graph ist ein Gebilde, bestehend aus zwei endlichen Mengen. Der Menge der Knoten V (engl. vertices) und der Menge der Kanten E (engl. edges). Die Menge der Knoten V ist eine nicht-leere endliche Menge, bestehend aus Elementen. Die Menge der Kanten E ist eine endliche Menge, bestehend aus Knotenpaaren. Die Menge der Kanten E kann

leer sein. Kanten können ungerichtet oder gerichtet sein. Im Rahmen dieser Arbeit werden nur gerichtete Kanten betrachtet. Das Konzept des geordneten Paares (a, b) nach Definition (A.23) beschreibt im Kontext der Graphentheorie eine gerichtete Kante e . Die Komponenten a und b sind Elemente der Menge V . Das Element a wird als Anfangsknoten und das Element b wird als Endknoten einer Kante bezeichnet. Die Knoten a und b werden als adjazent bezeichnet, wenn es eine Kante gibt, die sie verknüpft. Sie sind benachbart. Aus der Sicht der Kante gilt: e ist inzident mit den Knoten a und b . Die Kante e ist verknüpft mit a und b . Die Menge der Kanten E stellt eine homogene binäre Relation in der Menge V dar.

$$\text{Schlichter Graph: } G = (V; E) \quad (\text{A.56})$$

$$E = \{(a, b) \in V \times V \mid aRb\}$$

V Knotenmenge

E Kantenmenge

Die Eigenschaften schlichter Graphen ergeben sich aus den Eigenschaften für homogene binäre Relationen. Diese sind in Abschnitt A.3 zusammengestellt und gelten gleichermaßen für gerichtete Graphen.

Bipartiter Graph: Lässt sich die Menge der Knoten eines schlichten Graphen in zwei disjunkte Teilmengen zerlegen und existieren ausschließlich Kanten, deren Knotenpaare nicht zu derselben Knotenmenge gehören, dann wird der Graph als bipartiter Graphen bezeichnet. Ein bipartiter Graph ist immer antireflexiv. Für einen bipartiten Graphen existiert immer ein assoziierter gerichteter Graph.

$$\text{Bipartiter Graph: } G = (V; E) \quad (\text{A.57})$$

$$\text{Knotenmenge: } V = \{V_A \cup V_B \mid V_A \cap V_B = \emptyset\}$$

$$\text{Kantenmenge: } E = \{E_1 \cup E_2 \mid E_1 \subseteq V_A \times V_B \\ \wedge E_2 \subseteq V_B \times V_A\}$$

Verknüpfungen: Auf der Grundlage der Relationenverknüpfungen werden die Gleichheit und Inklusion für Graphen definiert. Gegeben seien die Graphen $G_1 = (V_1; E_1)$ und $G_2 = (V_2; E_2)$. Die Gleichheit drückt aus, dass sowohl die Menge der Knoten als auch die Menge der Kanten gleich ist.

$$\text{Gleichheit: } G_1 = G_2 \quad :\Leftrightarrow \quad V_1 = V_2 \wedge E_1 = E_2 \quad (\text{A.58})$$

Die Inklusion unterscheidet zwei Fälle: Teilgraph und Untergraph. Gegeben sei ein Graph $G = (V; E)$. Ein Teilgraph $G_T = (V_T; E_T)$ ergibt sich aus dem Graphen G durch Entfernen von Kanten aus der Menge E , ohne die Menge der Knoten V zu verändern.

$$\text{Teilgraph: } G_T \subseteq G \quad :\Leftrightarrow \quad V_T = V \wedge E_T \subseteq E \quad (\text{A.59})$$

Ein Untergraph $G_U = (V_U; E_U)$ ergibt sich aus dem Graphen G durch Entfernen von Knoten und Kanten. Eine Kante e wird entfernt, indem aus der Relation E_U das zu e gehörende geordnete Paar gelöscht wird. Das Entfernen eines Knotens v verlangt sowohl das Löschen aus V_U als auch das Entfernen derjenigen Kanten aus E_U , die mit v inzident sind.

$$\text{Untergraph: } G_U \subseteq G \quad :\Leftrightarrow \quad V_U \subseteq V \wedge E_U \subseteq E \cap (V_U \times V_U) \quad (\text{A.60})$$

Enthält der Untergraph G_U alle Kanten aus der ursprünglichen Kantenmenge E , für die die Knoten a und b des geordneten Paares (a, b) in der Knotenmenge V_U enthalten sind, so spricht man von einem induzierten Untergraphen. Für einen induzierten Untergraphen gilt:

$$\bigvee_{a,b \in E} (a, b) \in E \Rightarrow (a, b) \in E_U$$

Induzierter Untergraph:

$$G_U \subseteq G \quad :\Leftrightarrow \quad V_U \subseteq V \wedge E_U = E \cap (V_U \times V_U) \quad (\text{A.61})$$

Die Vereinigung zweier Graphen berechnet sich aus der Vereinigungsmenge aller Knoten und der Vereinigungsmenge aller Kanten. Entsprechend berechnet sich der Durchschnitt zweier Graphen als Schnittmenge aller Knoten und aller Kanten. Gegeben seien die Graphen $G_1 = (V_1; E_1)$ und $G_2 = (V_2; E_2)$.

$$\text{Durchschnitt: } G_1 \cap G_2 \quad := \quad (V_1 \cap V_2; E_1 \cap E_2) \quad (\text{A.62})$$

$$\text{Vereinigung: } G_1 \sqcup G_2 \quad := \quad (V_1 \cup V_2; E_1 \sqcup E_2) \quad (\text{A.63})$$

Abbildungen von Graphen: Gegeben sei ein Graph $G = (V; E)$ und eine Abbildung $F : V \rightarrow Z$. Erzeugt man für alle Kanten $(v_1, v_2) \in E$ nach der Vorschrift $(f(v_1), f(v_2))$ Bildkanten und fasst diese in einer Kantenmenge E_Z zusammen, dann bezeichnet man den Graphen $G_Z = (Z; E_Z)$

als induzierten Graphen.

$$\begin{aligned}
 \text{Graph:} & & G &= (V; E) \\
 \text{Abbildung:} & & F &: V \rightarrow Z \\
 \text{Induzierte Relation:} & & & \\
 E_Z & \rightarrow \{ (z_1, z_2) \in Z \times Z \mid \bigvee_{(v_1, v_2) \in E} [(z_1, z_2) = (f(v_1), f(v_2))] \} \\
 \text{Induzierter Graph:} & & G_I &= (Z; E_Z) \tag{A.64}
 \end{aligned}$$

Existiert für alle Kanten (v_1, v_2) aus E nach der Vorschrift $(f(v_1), f(v_2))$ eine Bildkante in E_Z , dann bildet die Abbildung $F : V \rightarrow Z$ den Graphen G homomorph auf G_Z ab.

$$\begin{aligned}
 \text{Homomorphe Abbildung:} & & & \tag{A.65} \\
 F \text{ ist homomorph} & :\Leftrightarrow \bigwedge_{v_1} \bigwedge_{v_2} ((v_1, v_2) \in E \Rightarrow (f(v_1), f(v_2)) \in E_Z)
 \end{aligned}$$

Ist die Abbildung $F : V \rightarrow Z$ bijektiv, dann existiert eine Umkehrabbildung $F^{-1} : Z \rightarrow V$. Eine bijektive Abbildung bildet den Graphen G isomorph auf G_Z ab. Isomorphe Graphen besitzen dieselbe Struktur.

$$\begin{aligned}
 \text{Isomorphe Abbildung:} & & & \tag{A.66} \\
 F \text{ ist isomorph} & :\Leftrightarrow (F \text{ ist homomorph}) \wedge (F^{-1} \text{ ist homomorph})
 \end{aligned}$$

A.5 Wege und Zyklen

Die Struktur eines Graphen $G = (V; E)$ ist durch die Struktur der Kantenrelation E festgelegt. Gelingt es, einen Graphen, dessen Struktureigenschaften bekannt sind, durch Abbildungen in Untergraphen zu zerlegen, kann aus den Struktureigenschaften der Untergraphen auf Eigenschaften des Graphen geschlossen werden. Wege und Zyklen stellen einfache Untergraphen dar. Im Folgenden werden verschiedene Begriffe eingeführt, die der Beschreibung der Struktur dienen.

Vorgänger und Nachfolger: Die Verknüpfung der Kantenrelation E mit der unären Punktrelation x führt zu der Menge der direkten Vorgänger und der Menge der direkten Nachfolger.

$$\text{Vorgänger von } x: \quad t_V(x) = E \circ x \tag{A.67}$$

$$\text{Nachfolger von } x: \quad t_N(x) = E^T \circ x \tag{A.68}$$

Verwendet man an Stelle der Kantenrelation E die Kantenrelation E^n , dann erhält man eine Berechnungsvorschrift für indirekte Vorgänger und Nachfolger. Die Potenz n gibt an, dass es für jeden Knoten y aus $t_V^{(n)}$ und $t_N^{(n)}$ einen zusammenhängenden Kantenzug der Länge n gibt, der x mit y verbindet.

$$n\text{-te Vorgänger von } x: \quad t_V^{(n)}(x) = E^n \circ x \quad (\text{A.69})$$

$$n\text{-te Nachfolger von } x: \quad t_N^{(n)}(x) = E^{nT} \circ x \quad (\text{A.70})$$

Verwendet man an Stelle der Kantenrelation E ihre transitive Hülle E^+ , dann erhält man eine Berechnungsvorschrift für alle indirekten Vorgänger und Nachfolger.

$$\text{alle Vorgänger von } x: \quad t_V^{(+)}(x) = E^+ \circ x \quad (\text{A.71})$$

$$\text{alle Nachfolger von } x: \quad t_N^{(+)}(x) = E^{+T} \circ x \quad (\text{A.72})$$

Knotengrad: Der Knotengrad gibt an, zu wievielen Kanten ein Knoten inzident ist. Bei schlichten Graphen wird zwischen dem Eingangsgrad und dem Ausgangsgrad unterschieden, da die Kanten gerichtet sind. Die Anzahl der in $t_V(x)$ und $t_N(x)$ enthaltenen Elemente wird als Knotengrad bezeichnet.

$$\text{Vorgängergrad von } x: \quad g_V(x) = |t_V(x)| = |E \circ x| \quad (\text{A.73})$$

$$\text{Nachfolgergrad von } x: \quad g_N(x) = |t_N(x)| = |E^T \circ x| \quad (\text{A.74})$$

Weg: Ein Weg ist eine Folge an Kanten, wobei für benachbarte Kantenpaare in der Folge gilt, der Endknoten der Vorgängerkante ist identisch mit dem Anfangsknoten der Nachfolgekante. Gegeben sei ein Graph $G = (V; E)$. Ein Weg ist definiert als eine Abbildung einer Indexmenge auf die Menge der Kanten.

$$F : N \rightarrow E \text{ ist ein Weg} \quad :\Leftrightarrow \quad e_i = a_{i+1} \quad \text{für } i = 1, \dots, n-1 \quad (\text{A.75})$$

$$N = \{1, \dots, n\}$$

$$f(i) \quad \text{gerichtete Kante } (a_i, e_i)$$

$$a_i \quad \text{Anfangsknoten der Kante } f(i)$$

$$e_i \quad \text{Endknoten der Kante } f(i)$$

Die Existenz eines Weges in einem Graphen kann aus der transitiven Hülle der Kantenrelation ermittelt werden.

$$\text{Es existiert ein Weg } \neq \emptyset \text{ von } x \text{ nach } y: \quad \Leftrightarrow \quad x \circ y^T \subseteq E^+ \quad (\text{A.76})$$

Die Länge eines Weges ergibt sich aus der Anzahl der Elemente in N , also der Anzahl der Kanten, die zu dem Weg gehören. Ein Weg kann leer sein.

Zyklus: Ein Weg $F : N \rightarrow E$, dessen Anfangs- und Endknoten identisch sind, wird als Zyklus bezeichnet. Ist ein Weg ein Zyklus mit der Weglänge 1, dann wird der Weg als Schlinge bezeichnet. Eine Schlinge ist eine Kante mit identischem Anfangs- und Endknoten. Ein Graph, der keine Zyklen und keine Schlingen enthält wird als azyklischer Graph bezeichnet. Im Rahmen dieser Arbeit werden nur azyklische Graphen betrachtet.

$$\text{Azyklischer Graph} \quad :\Leftrightarrow \quad E^+ \cap E^{+T} = \emptyset \quad (\text{A.77})$$

Zusammenhang: Ist ein Knoten x über einen Weg mit einem Knoten y verbunden, so heißt der Knoten y von einem Knoten x erreichbar. Durch Aufheben der Richtung kommt man zu dem Begriff des Zusammenhangs. Zwei Knoten stehen in einem Zusammenhang, wenn es einen Weg gibt, der x mit y verbindet oder umgekehrt. Ein Graph heißt zusammenhängend, wenn alle Knoten paarweise zusammenhängen. Gegeben sei ein Graph $G = (V; E)$. Die reflexive transitive Hülle der Vereinigung $E \sqcup E^T$ heißt Zusammenhangsrelation Z . Sie enthält alle Knotenpaare die zusammenhängen. Ist sie identisch mit der Allrelation E , dann hängen alle Knoten paarweise zusammen.

$$\text{Zusammenhangsrelation:} \quad Z := (E \sqcup E^T)^* \quad (\text{A.78})$$

$$\text{Zusammenhängender Graph} \quad :\Leftrightarrow \quad Z = E \quad (\text{A.79})$$

Ist $Z \neq E$ dann lässt sich der Graph G eindeutig in n zusammenhängende Untergraphen G_k zerlegen. Diese Untergraphen G_k werden als Zusammenhangskomponenten bezeichnet.

$$\text{Zusammenhangskomponente:} \quad G_k := (V_k; E_k) \quad (\text{A.80})$$

Für schlichte Graphen $G = (V; E)$ gilt:

$$G = \bigsqcup_{k=1}^n G_k$$