

Abbildbarkeit unstrukturierter Prozessmodelle auf strukturierte Workflows

Eine Untersuchung am Beispiel BPMN und ADEPT



Bachelorarbeit
der Fakultät für Ingenieurwissenschaften und
Informatik der Universität Ulm

vorgelegt von

Nicolas Mundbrod
August 2008

Prüfer der Arbeit:
Professor Dr. Manfred Reichert
Institut für Datenbanken und Informationssysteme

Amtierender Dekan: Prof. Dr. Helmuth Partsch

Gutachter: Prof. Dr. Manfred Reichert

Tag der Abgabe: 15. August 2008

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich diese Bachelorarbeit selbständig angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle wörtlich oder sinngemäß übernommenen Ausführungen wurden als solche gekennzeichnet. Weiterhin erkläre ich, dass ich diese Arbeit in gleicher oder ähnlicher Form nicht bereits einer anderen Prüfungsbehörde vorgelegt habe.

Ulm, den 15. August 2008

Nicolas Mundbrod

Danke

Zu allererst möchte ich meinen Betreuern Prof. Dr. Manfred Reichert und Jens Kolb meinen Dank für ihre Hilfe bei der Erstellung dieser Arbeit aussprechen. Die Diskussionen mit ihnen, ihre Ratschläge und ihre große Hilfsbereitschaft haben mir bei der Erstellung der Arbeit sehr geholfen.

Weiterhin möchte ich mich bei meiner Freundin Sarah Fritz für ihren großen Rückhalt während der Erstellung dieser Arbeit bedanken.

Nicht zuletzt möchte ich meiner Familie danken, die mir durch ihre uneingeschränkte Unterstützung das Studium und somit auch diese Arbeit ermöglichten.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Prozessorientierung	1
1.2	Problemstellung und Zielsetzung	3
1.3	Struktur der Arbeit	5
2	Grundlagen	6
2.1	Prozesse	6
2.2	Workflows	8
2.3	Zusammenhang und Abgrenzung	10
3	Strukturelle Eigenschaften	13
3.1	Strukturelle Klassifizierung von Workflows	13
3.2	Bewertung und Folgen struktureller Eigenschaften	18
4	Modellierung	25
4.1	Bekannte Formen	25
4.2	BPMN	26
4.3	ADEPT	32
5	Abbildbarkeit der BPMN auf ADEPT	39
5.1	Äquivalenz	39
5.2	Grundlagen der Abbildbarkeit	41
5.3	Abbildbarkeit von Modellen mit strukturellen Mängeln	44
5.4	Automatisierbarkeit	53
6	Schlussbetrachtung	56

Zusammenfassung

Die letzten Jahre haben eine Vielzahl an unterschiedlichen Notationen zur Modellierung von Prozessen oder Workflows hervorgebracht. Die Notationen unterscheiden sich neben ihrer unterschiedlichen Symbolik besonders in der Ausdrucksmächtigkeit.

So existieren Notationen ohne Einschränkungen bis hin zu Notationen mit sehr weit reichenden syntaktischen Restriktionen, wie z.B. die Blockstruktur. Das Voraussetzen von strukturellen Einschränkungen vereinfacht es Workflow-Management-Systemen (WfMS) Analysen durchzuführen und z.B. Deadlocks zu erkennen. Die zur Modellierung benötigten Notationen weisen durch syntaktische Restriktionen im Vergleich zu syntaktisch uneingeschränkten Notationen eine höhere Komplexität und oftmals eine geringere Ausdrucksmächtigkeit auf. Daher existiert eine Konkurrenzsituation zwischen den Ausdrucksmöglichkeiten und der Ausdrucksmächtigkeit der Notation, sowie den Anforderungen der WfMS um unabdingbare Eigenschaften des Workflows zu sichern.

Das Ziel dieser Arbeit ist es, die Abbildbarkeit unstrukturierter Prozessmodelle auf strukturierte Workflows wissenschaftlich zu untersuchen und den Stand der bereits publizierten Verfahren zu erfassen. Es werden Muster innerhalb eines unstrukturierten Prozess-Modells identifiziert, die eine Abbildung auf strukturierte Modelle ausschließen. Darüber hinaus soll versucht werden, die Abbildungen in automatische, semi-automatische und manuelle zu unterteilen.

Zur Veranschaulichung wurden die Business Process Modeling Notation (BPMN) der Object Management Group (OMG) als unstrukturierte Notation und ADEPT¹ des Instituts für Datenbanken und Informationssysteme (DBIS) als strukturierte Notation ausgewählt.

¹ADEPT steht für *Application Development based on Encapsulated Premodelled Process Templates*

Kapitel 1

Einleitung

Die meiste Zeit geht dadurch verloren, daß man nicht zu Ende denkt.

Alfred Herrhausen (1930-1989)
dt. Bankier, Vorstandspr. Dt. Bank

1.1 Prozessorientierung

Fritz Nordsieck, einer der Hauptvertreter der betriebswirtschaftlichen Organisationslehre, erkannte schon 1932 in seiner Dissertation *Die schaubildliche Erfassung und Untersuchung der Betriebsorganisation* den Vorteil einer Orientierung an Prozessen:

„Der Betrieb ist in Wirklichkeit ein fortwährender Prozess, eine ununterbrochene Leistungskette. [...] Anzustreben ist in jedem Fall eine klare Prozessgliederung.“[1]

Im Zuge des stärker werdenden globalen Wettbewerbes, der einen zusätzlichen Kosten- und Preisdruck verursacht, rücken die Flexibilität auf neue Marktgegebenheiten zu reagieren und eine hohe Automatisierbarkeit in den Fokus.

Eine Prozessorientierung bietet in diesem Fall Vorteile, denn es wurde erkannt, dass jedes von einem Unternehmen angebotene Produkt (Geschäftsziel) das Resultat einer koordinierten Abfolge von Aktivitäten (Geschäftsprozess) ist. Ein Geschäftsprozess und das daraus entstehende Produkt sind folglich zwei Seiten einer Medaille (Schaubild 1.1).

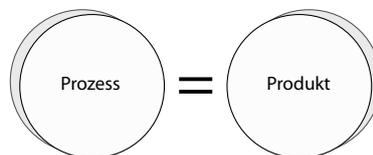


Schaubild 1.1: Prozess und Produkt – zwei Seiten einer Medaille

Diese Erkenntnis ist wichtig, folgt doch aus ihr, dass durch die Verbesserung des Geschäftsprozesses, die dabei entstehenden Produkte ebenfalls verbessert werden kön-

nen. Angehörige der Wirtschaftswissenschaften sind deshalb daran interessiert, Geschäftsprozesse im Bezug auf Effizienz und Qualität optimieren zu können. Sie wollen unter anderem die „Time-to-Market“ neuer Produkte wie auch die Anpassungszeit bestehender Produkte an neue Marktgegebenheiten minimieren, die Kosten verringern und die Kundenzufriedenheit steigern.

Um Geschäftsprozesse zu analysieren und zu optimieren, werden in der Regel Modelle anhand einer graphischen Notation erstellt. Bis heute hat sich keine Notation den Stellenwert in der Prozessmodellierung erworben, der vergleichbar wäre mit zum Beispiel der von der Object Management Group (OMG, <http://www.omg.org>) publizierte Unified Modeling Language (UML, [2]) für objektorientiertes Design und Analyse. Doch strebt die OMG mit ihrer Notation zur Prozessmodellierung namens Business Process Modeling Notation (BPMN) ebenfalls eine Standardisierung in diesem Bereich an. Die OMG präzisiert die Ziele der BPMN in der Spezifikation der BPMN vom Februar 2008 im Vorwort des Textes wie folgt:

„The primary goal of BPMN is to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes.“ [3]

Gerade durch das Ziel der leichten Verständlichkeit aller am Prozess beteiligten Personen und der entsprechenden Umsetzung erreichte die BPMN in kurzer Zeit einen relativ hohen Verbreitungsgrad [4], obwohl sie die direkte Unterstützung der Prozesse durch Informationstechnik nicht anbietet.

Die Wurzeln des Gedankens, Geschäftsprozesse durch Informationssysteme zu unterstützen und zu begleiten, findet sich in den 1970er Jahre, ebenfalls angesprochen durch Fritz Nordsieck:

„Denken wir [...] an eine moderne Datenverarbeitung. Auch sie stellt einen deutlichen Prozeß dar, der sogar in verschiedenen Abschnitten mit dem Betriebsprozeß verknüpft ist und ihn ständig begleitet, wenn nicht sogar steuert.“ [5], S.9

Um einen Geschäftsprozess oder Teile dessen zu automatisieren, können Workflows (dt. Arbeitsfluss) verwendet werden (Definition und Details siehe später in Kapitel 2). Kurzgefasst könnte ein Workflow als die verfeinerte, operative technische Sicht auf einen zuvor fachlich, eventuell abstrakt erstellten Geschäftsprozess, oder auf Teile dessen, angesehen werden. Während bei einem Geschäftsprozess das Hauptaugenmerk darauf liegt, was und in welcher Reihenfolge etwas getan wird, erweitert man in Bezug auf den Workflow die Sicht hauptsächlich wie etwas und von wem etwas getan wird. Ein Software-System, das die Ausführung von Workflows definiert, vornimmt und verwaltet, wird als Workflow-Management-System (WfMS) bezeichnet. Die Spezifikation eines Workflows für die Ausführung durch ein WfMS findet, analog zur Modellierung von Prozessen, durch Modelle statt. Jedoch setzt ein WfMS zumeist strukturelle Anforderungen an das Modell eines Workflows, damit Laufzeitfehler, wie zum Beispiel

Deadlocks, verhindert werden.

Das WfMS ADEPT2¹ des Institutes für Datenbanken und Informationssysteme (DBIS) der Universität Ulm setzt für die Modellierung der Workflows in der zum System gehörenden Notation ADEPT eine Blockstruktur für die Modellierung von Workflows voraus. Damit können die aus einer fehlerhaften Modellierung resultierende Laufzeitfehler von vornherein ausgeschlossen werden. Jedoch bietet ADEPT zusätzliche Kannten zur Synchronisation an, die die Ausdrucksstärke der blockstrukturierten Notation zusätzlich erweitert (siehe Kapitel 4).

1.2 Problemstellung und Zielsetzung

Der Übergang von der Erfassung und Analyse von Geschäftsprozessen durch eine Notation wie BPMN hin zur Umsetzung der Prozesse durch Workflows, in zum Beispiel ADEPT, ist leider nicht als nahtlos anzusehen und resultiert aus den konzeptionellen Unterschieden der Notationen zu Modellierung von Geschäftsprozessen und Workflows.

Die Geschäftsprozessmodelle werden in der Regel von Personen erstellt, die einem der Informatik fremden Umfeld zuzuordnen sind. Notation zur Modellierung von Geschäftsprozessen sollten somit den technischen Aspekt der späteren Umsetzung vernachlässigen, damit die Prozessmodelle leicht verständlich sind und als Diskussionsgrundlage dienen können. Die Modellierung der Prozesse kann nur gelingen, wenn die entsprechenden Fachkräfte der Abteilungen in die Modellierung der Prozesse mit eingebunden werden. Die Geschäftsleitung, wie auch der Modellierer selbst, sind ebenfalls an einer möglichst selbsterklärenden Notation interessiert. Nicht zuletzt sollen Geschäftsprozesse auch sehr abstrakt und generell erfasst werden können.

Um möglichst effizient und ungebunden die Prozesse erfassen zu können, ist der Wunsch nach einer Notation, die möglichst ausdrucksstark und keinerlei Beschränkungen unterworfen ist, ebenfalls verständlich. Deshalb setzt die BPMN in der Modellierung der Geschäftsprozesse keinerlei strukturelle Einschränkungen, die für eine eventuell spätere Ausführung durch eine WfMS notwendig wären, voraus.

Falls nun ein Geschäftsprozess, oder ein Teil dessen, durch IT unterstützt und dies durch ein WfMS geschehen soll, stellt sich zwangsläufig die Frage, ob es möglich ist, die bereits existenten Prozessmodell in das WfMS übernehmen zu können und diese gegebenenfalls „nur“ noch um technische und operative Details verfeinern zu können. Deshalb ist es das Ziel dieser Arbeit, den Stand der bereits publizierten Arbeiten über die Abbildbarkeit von unstrukturierten Prozessmodellen auf (block-)strukturierte Workflows zu untersuchen, um zu ergründen, ob und in wie weit die Lücke zwischen der Prozessmodellierung und Workflowmodellierung zu schließen ist. In Bezug auf komplexe, groß dimensionierte Prozessmodelle wäre eine Abbildung ein erheblicher Vorteil gegenüber einer expliziten Nachmodellierung der Modelle für das WfMS. Ohne gene-

¹Die Entwicklung von ADEPT2 wurde durch das Land Baden-Württemberg im Rahmen des AristaFlow-Projektes (<http://www.aristaflow.de>) finanziert

relle Aussagen zur Abbildung einzuschränken, sollen zur Konkretisierung und Veranschaulichung zwei Notationen in die Untersuchung mit eingebunden werden. Zum Einen die bereits erwähnte BPMN als unstrukturierte Notation und zum Anderen die Notation ADEPT des gleichnamigen WfMS des Instituts DBIS der Universität Ulm als blockstrukturierte Notation.

Um die Zielsetzung dieser Arbeit zu präzisieren, sollen an dieser Stelle für diese Untersuchung der Abbildbarkeit relevante Fragen aufgeführt werden. Nach der Untersuchung der Abbildbarkeit sollen diese Fragen während der Schlussbetrachtung in Kapitel 6 nochmals aufgegriffen und beantwortet werden.

- *Ist jedes Prozessmodell modelliert in der BPMN auf ADEPT abbildbar?*
- *Falls nicht, existiert eine Teilmenge an Elementen in der BPMN, die eine vollständige Abbildung nach ADEPT ermöglichen?*
- *Kann jedes Prozessmodell, das erfolgreich beendet werden kann (well-behaved), in einen in ADEPT modellierten Workflow umgewandelt werden?*
- *Gibt es bereits existente Abbildungen der BPMN auf andere blockstrukturierte Notationen als ADEPT?*
- *In welcher Form wirken sich Erweiterungen der blockstrukturierten Notation ADEPT auf die Abbildbarkeit aus?*
- *Könnte die blockstrukturierte Notation ADEPT um weitere Elemente bzw. Konstrukte erweitert werden, um die Abbildbarkeit zu erhöhen?*

1.3 Struktur der Arbeit

An dieser Stelle soll ein Überblick über den weiteren Verlauf dieser Arbeit gegeben werden. So sollen im folgenden Kapitel 2 die Begriffe Prozess und Workflow präzisiert werden und der Zusammenhang dieser Begriffe erläutert werden. In Kapitel 3 wird die Bedeutung und Folgen von Struktur in einem Modell eines Prozesses bzw. eines Workflows untersucht werden. Dazu werden unstrukturierte und blockstrukturierte Modelle klar definiert und die Unterschiede dieser Modelle betont. Die Folgen von strukturellen Mängeln, die in unstrukturierten Modellen auftreten können, werden ebenfalls aufgeführt und eingeordnet. Kapitel 4 soll die Notationen BPMN und ADEPT vorstellen und zuvor einen kurzen Anriss über weitere Möglichkeiten zur Modellierung von Prozessen geben. Es sollen die für eine Abbildung interessanten Elemente vorgestellt werden und anschließend die Modellierung von grundlegenden Modellen anhand von Schaubildern aufgezeigt werden. Eine Einbeziehung der Ergebnisse aus Kapitel 3 erfolgt ebenfalls.

Nach der Schaffung der Grundlagen soll in Kapitel 5 die Untersuchung der Abbildbarkeit der Notation BPMN auf die Notation ADEPT durchgeführt werden. Es wird zuerst der Begriff der Äquivalenz von Prozessmodellen und Workflows erläutert, um dann anhand von Ergebnisse aus bereits publizierten Arbeiten die Abbildbarkeit zu untersuchen. Es wird außerdem auf die Automatisierbarkeit der Abbildungen Bezug genommen.

Abschließend findet in Kapitel 6 eine Zusammenfassung und Bewertung der Ergebnisse statt.

Kapitel 2

Grundlagen

Denn wer schwimmen will, muss zu den Flüssen gehen.

Isaac Newton (1643-1727)
englischer Physiker und Mathematiker

Hinter den Begriffen *Prozessmodell*, *Workflow*, *BPMN* und *ADEPT* aus dem Titel dieser Arbeit verbergen sich viele Begriffe und Themen, die in den folgenden Seiten näher erläutert und präzisiert werden sollen. Die Begriffe werden auch erläutert um Missverständnissen vorzubeugen, da in der Literatur oftmals Begriffe wie *Prozess* und *Workflow* unterschiedlich und wechselseitig verwendet werden. Es soll weiterhin der Zusammenhang zwischen Prozessen und Workflows herausgestellt werden.

2.1 Prozesse

In der Einleitung dieser Arbeit wurde bereits die Idee der Prozessorientierung kurz vorgestellt. Um eine spätere Untersuchung von Prozessmodellen zu erlauben, bietet es sich zuerst an, zuerst die Bedeutung des Wortes „Prozess“ im deutschen Sprachgebrauch zu ergründen. Neben der Bedeutung der *gerichtlichen Durchführung von Rechtstreitigkeiten*, die im Folgenden nicht berücksichtigt wird, präzisiert der Duden das Wort *Prozess* mit den Worten *Vorgang*, *Ablauf*, *Verfahren* und *Entwicklung*. Eine präzise Definition erfolgt durch die Literatur in [6]:

Definition 2.1 Ein **Prozess** ist ein allgemeiner Ablauf mehrerer Abschnitte, bei denen es sich um Aufgaben, Ausführungen, Arbeitsschritte o.ä. handeln kann. Zwischen diesen Prozessabschnitten bestehen bestimmte Abhängigkeiten.

Intuitiv stellt ein Prozessmodell ein Modell eines Prozesses dar. Es folgt aber auch hier eine ausführliche Definition analog zu [6].

Definition 2.2 Ein **Prozessmodell** beschreibt die Struktur eines realen Prozesses. Es bestimmt alle möglichen Pfade entlang des Prozesses und bestimmt die Regeln für die Wahl der Pfade. Weiterhin bestimmt das Prozessmodell alle Aktivitäten, die ausgeführt werden müssen.

Die praktische Umsetzung der Modellierung eines Prozesses soll anhand eines Beispiels aufgezeigt werden. Betrachtet werden soll ein Bestellprozess eines fiktiven Versandhandels. Dieser Prozess besteht aus einer Menge an Aktivitäten, die in koordinierter Form ausgeführt werden. Nach Eingang einer Bestellung wird eine Rechnung versandt und gleichzeitig soll der Artikel für den Versand vorbereitet werden. Nach Erhalt des Betrags der Rechnung und nach Abschluss der Vorbereitungen wird das bestellte Produkt versandt. Das Schaubild 2.1 soll diesen kleinen Beispielprozess anhand eines Prozessmodells illustrieren.

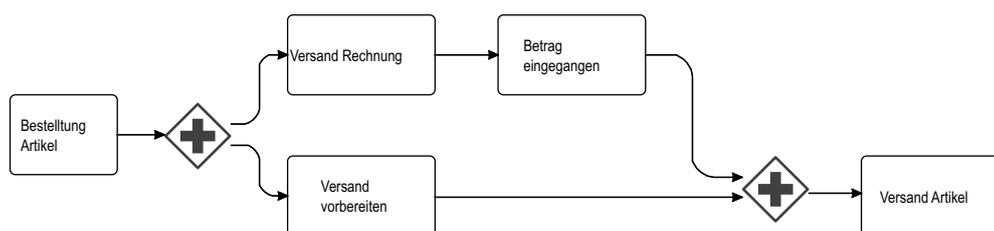


Schaubild 2.1: Einfacher Bestellprozess eines Versandhandels

Man erkennt, dass Modelle besser geeignet sind, Abhängigkeiten zwischen Aktivitäten darzustellen, als textuelle Beschreibungen. Im Speziellen für großdimensionierte und damit im zunehmenden Maße komplex werdende Prozesse sind Modelle unerlässlich, da sie textuell kaum in einem angemessenen und verständlichen Rahmen zu erfassen wären. Außerdem ist es möglich, Prozesse abstrakt zu modellieren, den technischen Aspekt der Realisierung zu vernachlässigen, und somit den Schwerpunkt auf die Prozessstruktur und die Interaktion zwischen abstrakten Aktivitäten zu setzen. Dies ist ein bedeutender Punkt, da Prozessmodelle somit prinzipiell nicht an bestimmte Strategien oder Plattformen zur Umsetzung gebunden sind.

Natürlich kann die graphische Modellierung von Prozessen in verschiedenen Arten und Notationen erfolgen. Die Granularität der Modellierung, also wie detailliert der Prozess modelliert wird, kann ebenfalls variieren und ist, wie die Wahl der Notation, dem Modellierer überlassen. Zur Modellierung des Beispielprozesses wurde eine vereinfachte Version der Business Process Modelling Notation (BPMN) verwendet, die später in Kapitel 4 ausführlich beschrieben wird. Die Ausführungsbeschränkungen, die zuvor textuell beschrieben wurden und die sich darauf beziehen, welche Aktivität wann ausgeführt wird, werden durch Kontrollflusselemente realisiert.

Im Beispiel sind dies die Pfeile und die Rauten. Ein Pfeil von Aktivität „Versand Rechnung“ nach „Empfange Betrag“ bedeutet, dass erst die Rechnung verschickt wird und danach der Betrag in Empfang genommen werden kann. Die erste Raute mit dem Plus in der Mitte bedeutet, dass die Zweige, in denen einerseits die Rechnung verschickt und

danach die Zahlung eingeht und andererseits der Artikel zum Versand vorbereitet wird, parallel ablaufen. Die zweite Raute mit dem Plus, in der die zwei parallelen Zweige einmünden, bedeutet, dass der Artikel erst versandt wird, wenn die Aktivitäten beider einmündeten Zweige beendet sind. Nach dem Versand des Artikels ist der Prozess beendet.

Prozesse, wie der Beispielprozess, die sich auf ein Unternehmen und der Verwirklichung dessen Geschäftsziele beziehen, werden als Geschäftsprozess (GP) oder aus dem Englischen ebenfalls gebräuchlich als Business Processes (BP) bezeichnet. In den neunziger Jahren des letzten Jahrhunderts entstand ein deutlicher Trend zur Orientierung nach Prozessen innerhalb von Unternehmen. Unter anderem legten Michael Hammer und James Champy in ihrer Arbeit [7] aus dem Jahr 1993 den Grundstein für das sogenannte Business Process Reengineering und empfahlen die radikale Neugestaltung der Geschäftsprozesse eines Unternehmens. Selbst in der Neufassung der Normenreihe des Qualitätsmanagement (ISO 9000 ff.) im Jahr 2000 wurde der Prozessgedanke in den Vordergrund gestellt [6].

In der Folge soll die Definition eines Geschäftsprozesses nach [8], basierend bzw. ähnlich zu den Definitionen in [7], [9] und [6], erfolgen:

Definition 2.3 *Ein Geschäftsprozess besteht aus einer Menge an Aktivitäten, die koordiniert in einem organisatorischem und fachlichem Umfeld ausgeführt werden. Diese Aktivitäten verwirklichen gemeinsam ein Geschäftsziel. Jeder Geschäftsprozess wird von einer einzelnen Organisation durchgeführt, aber er kann mit Geschäftsprozesse anderer Organisationen interagieren.*

Um Irritationen zu vermeiden soll angemerkt werden, dass wenn in der Folge von Prozessen oder Prozessmodellen die Rede ist, in der Regel Geschäftsprozesse bzw. Geschäftsprozessmodelle gemeint sind.

2.2 Workflows

In Bezug auf die Automatisierbarkeit spielen Informationssysteme eine wichtige Rolle. Der Wunsch in einer durchgängigen, vorgangsorientierten Rechnerunterstützung von Teilen oder des gesamten Geschäftsprozesses mündet im Begriff des „Workflows“, zu Deutsch „Arbeitsablauf“.

Definition 2.4 *Ein Workflow ist ein teilweise oder vollständig automatisierter Geschäftsprozess, in dem Dokumente, Informationen oder Aufgaben zwischen Teilnehmern entsprechend einer Menge von Ausführungsregeln übertragen werden.*

Kurzgefasst ist ein Workflow die verfeinerte, operative technische Sicht auf einen zuvor fachlich, eventuell abstrakten Geschäftsprozess oder auf Teile dessen. Während bei einem Geschäftsprozess das Hauptaugenmerk darauf liegt, was und in welcher Reihenfolge etwas getan wird, erweitert man in Bezug auf den Workflow die Sicht hauptsächlich auf wie etwas und von wem etwas getan wird. Die Definition des Workflows

hält sich weitgehend an die Definition der Workflow Management Coalition (WfMC) [10]. Die WfMC wurde im August 1993 gegründet um die Aktivitäten von Herstellern, Benutzern und Akademikern rund um das Thema Workflow zu bündeln. Heute hat die WfMC über 300 Mitglieds-Organisationen. Unter Hersteller sind hauptsächlich die Produzenten von Workflow Management Systeme (WfMS) zu verstehen. Eine Präzisierung dieser Systeme erfolgte ebenfalls durch die WfMC [8]. Die folgende Definition ist daran angelehnt.

Definition 2.5 *Ein Workflow Management System ist ein Softwaresystem, das mit Hilfe von Software die Ausführung von Workflows definiert, vornimmt und verwaltet. Es basiert auf einer oder mehreren Workflow-Engines und ist in der Lage, Definitionen von Prozessen zu interpretieren, mit Beteiligten des Workflows zu interagieren und, falls es erforderlich ist, die Nutzung von IT-Werkzeuge und IT-Programme in Anspruch zu nehmen.*

Unter der „Definition von Prozessen“ in der Definition des WfMS ist die Spezifikation der Workflows, typischerweise durch ein Modell, zu verstehen. Die Modellierung der Workflows erfolgt weitestgehend analog zur bereits vorgestellten Modellierung eines Prozesses. Im Vordergrund steht koordinierte Abfolge von Aktivitäten. Jedoch werden in Modellen für Workflows weitaus mehr die operativen technischen Faktoren berücksichtigt und es werden sehr oft strukturelle Eigenschaften der Modelle vorausgesetzt, damit die Workflows korrekt ausgeführt werden können (detailliert in Kapitel 3). Das Wort Workflow wird, wie im Titel dieser Arbeit, oft auch synonym für die entsprechende Spezifikation oder Modell des Workflows verwendet. Eine Übersicht, welche Konzepte und Begriffe mit dem Term „Workflow“ abgedeckt und verbunden werden, ist in der Arbeit [11] von Diimitriou Georgakopoulos et al. zu finden.

Natürlich sind Workflow-Management-Systeme nicht die einzige Möglichkeit, einen Geschäftsprozess oder Teile dessen durch IT zu unterstützen. Gegenüber einer vielleicht klassischen Lösung, bei der eine maßgeschneiderte Software zur Unterstützung des Prozesses geschaffen wird, oder Spezialsoftware wie zum Beispiel ein CRM-System¹, führt jedoch ein WfMS den Prozessgedanken zu Ende, da es die Vorteile der Prozessorientierung übernimmt. Die klassische Lösung einer maßgeschneiderten Software würde der Prozessorientierung des Unternehmens eventuell nicht gerecht werden, da es die Flexibilität in der Anpassbarkeit der Prozesse stark beeinträchtigen würde. Nach einer Änderung eines Prozesses würde die entsprechende Software für diesen Fall umgeschrieben oder eventuell sogar neugeschrieben werden müssen, was oftmals mit enormen Kosten und Zeitaufwand verbunden ist. Workflows sind, durch ihre explizite Repräsentierung in Modellen, leichter anpassbar. Die Programmlogik ist in den Anwendungen, die zur Unterstützung der einzelnen Aktivitäten vorhanden sind oder geschaffen werden, gekapselt. Der Wunsch nach Integration von heterogener Unternehmenssoftware und Komposition in Prozessen mündet auch in dem aktuellen stark aufkommenden Interesse in service-orientierten Architekturen (SOA) und BPEL. Leider fehlt es der BPEL bisher in der Unterstützung der menschliche Interaktion mit Prozessen, die in Workflows realisierbar ist. Abhilfe soll hier der Draft BPEL4People

¹CRM-System = System zur Kundenpflege

schaffen. Die Vorstellung dieser Bestrebung und SOA im Generellen würde aber den Rahmen dieser Arbeit sprengen. Es sei hier auf entsprechende Literatur wie [12] verwiesen.

Zusätzlich reizvoll am Workflow-Konzept ist die Überlegung, die fachlich erstellten Prozessmodelle eventuell „nur“ verfeinern und um operative technische Aspekte erweitern zu müssen. Doch sind die Anforderungen der WfMS bzw. der BPMS hinsichtlich eines ausführbaren Modells ausgeprägter. Die Modelle müssen unter anderem Benutzerzuordnungen und Datenspezifikation enthalten und es müssen zur Bearbeitung von Aktivitäten zum Beispiel Standardsoftware eingebunden werden. Eine unabdingbare Anforderung eines jeden WfMS muss es sein, dass die Workflowmodelle nicht zu Laufzeitfehlern des Systems führen. Deshalb setzen WfMS im Allgemeinen strukturelle Eigenschaften ihrer Modelle voraus, da diese die Modellierung von inkorrekten Modellen, die zum Beispiel Deadlocks verursachen, verhindern. Es wird später im Detail darauf eingegangen und gezeigt, dass es Konstrukte innerhalb von Prozessmodellen gibt, die nicht ausführbar sind und sein dürfen. Beispielsweise werden, um Verklemmungen, die das erfolgreiche Beenden eines Workflows verhindern, zu vermeiden, Schleifen nur in bestimmter, geordneter Weise zugelassen. Bei den blockstrukturierten Modellierungssprachen, die strukturell am stärksten eingeschränkten Notationen, korrespondiert sogar jede spezifische Verzweigung im Kontrollfluss mit einem gleichbedeutenden Zusammenführen des Kontrollflusses und ergeben somit einen Block. Es kann bewiesen werden, dass diese Modelle keine Verklemmungen verursachen und somit den Anforderungen des WfMS gerecht werden. Aalst et al. zeigen in ihrer Arbeit *Workflow Patterns* [13], dass alle aktuellen WfMS blockstrukturierten Workflowmodelle unterstützen. Außerdem bieten viele von ihnen sogar die Möglichkeit an, das Laufzeitverhalten der Workflows vor der Einführung zu simulieren und entsprechend zu prüfen.

2.3 Zusammenhang und Abgrenzung

Nach Klärung der erwähnten Begrifflichkeiten ist es nun möglich, Beziehungen und Abhängigkeiten zu identifizieren und eine Einordnung vorzunehmen. Rund um das Thema Prozessorientierung innerhalb eines Unternehmens ergeben sich vielfältige Aufgaben. Das Modellieren und Erfassen von Geschäftsprozessen, unabhängig ob diese hinzukommen oder schon bestehen, ist ein Grundvorgang. Wie schon erwähnt stellt sich bei diesem Vorgang unter anderem die nicht unerhebliche Frage, in welcher Form Prozesse erfasst werden sollen.

Nach der Modellierung ergibt sich zwangsläufig die Frage nach der Umsetzung der Geschäftsprozesse. Unter der Annahme, dass zumindest Teile aus den Geschäftsprozessen durch IT unterstützt werden sollen, stellt sich das Problem, wie dies geschehen soll. Es ist natürlich dem Unternehmen überlassen, ob es auf ein Workflow Management System, auf eine Business Rules Engine² oder auf eine Spezialsoftware, wie zum

²Ein System zur effizienten Ausführung von Geschäftsregeln (Business Rules)

Beispiel ein CRM-System, zurückgreift. Natürlich ist wie bereits erwähnt eine eigens in Auftrag gegebene Software denkbar.

Der Umsetzung folgend sollte die Ausführung der Prozesse möglich sein. Dies bedeutet die Prozesse nach den gegebenen Modellen instanziiieren zu können und sie dabei überwachen zu können. Hierbei sollten Kennzahlen über die Ausführung festgehalten werden um zum Beispiel den Status einer Prozessinstanz erfassen zu können. Eventuell unterstützt die Ausführungsplattform auch Anpassungen der Prozesse in der Laufzeit. Diese Anpassungen sollten ebenfalls dokumentiert werden.

Die Metriken und Dokumentation der Anpassungen erweisen sich als nützlich, wenn die Prozesse evaluiert werden sollen. Die Evaluation sollte regelmäßig erfolgen und hat den Sinn einer konstanten Hinterfragung und Optimierung der Prozesse. Eine Anpassung der Prozesse würde selbstverständlich wieder in einer Modellierung bzw. Verfeinerung bestehender Prozesse münden.

Die erwähnten Vorgänge der Modellierung, Umsetzung, Durchführung und Evaluation können im Begriff des Geschäftsprozessmanagement (GPM) zusammengefasst werden. Die englische Version „Business Process Management“ (BPM) ist im deutschen Sprachraum ebenfalls häufig vorzufinden. Eine präzise Definition erfolgt analog nach [8]:

Definition 2.6 *Geschäftsprozessmanagement* beinhaltet Konzepte, Methoden und Techniken für das Design, die Administration, die Konfiguration, das Erlassen und die Analyse von Geschäftsprozessen.

Um eine Einordnung vorzunehmen, wird in Schaubild 2.2 die Idee des holländischen Forschers Will van der Aalst aufgenommen, der in [14] die Kernaspekte des GPM erläutert und anschaulich als sogenannten Business Process Lifecycle (dt. Geschäftsprozess Lebenszyklus) darstellt.

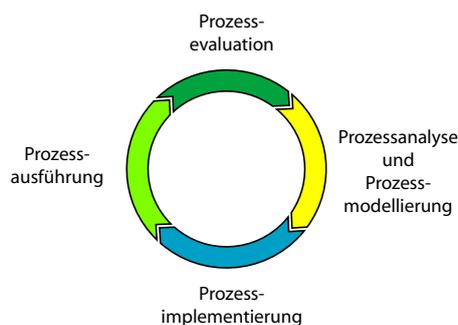


Schaubild 2.2: Geschäftsprozess-Lebenszyklus

Das Schaubild soll die Grundidee des Geschäftsprozessmanagement unterstreichen, dass nach der erfolgreichen Umsetzung und laufenden Durchführung die Be-

trachtung der Prozesse nicht abgeschlossen ist, sondern durch die Evaluation eine neue Phase in der Modellierung, Umsetzung und anschließender Durchführung beginnt. Geschäftsprozesse sind damit einem ständigen Verbesserungsprozess unterworfen und sind in dieser Weise der Wirklichkeit sehr nahe. Die Anforderungen an Unternehmen und an ihre Produkte unterliegen gerade im Zuge der Globalisierung einem ständigen Wandel und deshalb nimmt die Bedeutung der Fähigkeit, die Produkte rasch den neuen Gegebenheiten anzupassen zu können, immer mehr zu.

Es ist anzumerken, dass Hersteller der klassischen Workflow Management Systeme ihre Systeme dahingehend anpassen, dass sie den kompletten Zyklus des Geschäftsprozessmanagements unterstützen wollen [14]. Der Wandel in diese Richtung betrifft auch die Begrifflichkeiten der „Prozesse“ und „Workflows“ dahingehend, dass die Terme Prozess und Workflow sich mehr und mehr in ihrem Gebrauch und Bedeutung überlappen. Die Entwicklung der WfMS in Richtung Geschäftsprozessmanagement spiegelt sich in der Einführung der sogenannten Geschäftsprozessmanagementsysteme wider (nach [8]):

Definition 2.7 *Ein Geschäftsprozessmanagementsystem ist ein generisches Softwaresystem, das durch die explizite Prozessrepräsentation angetrieben, die Ausführung von Geschäftsprozessen koordiniert.*

Ob Notationen, die ursprünglich zur Modellierung von Workflows erdacht wurden, zur Modellierung von abstrakten und fachlichen Prozessen geeignet sind, sollte durchaus kritisch hinterfragt werden. Gerade in Unternehmen wäre eine einheitliche Notation zu Modellierung von Prozessen und Workflows als Diskussionsgrundlage besonders interessant. Die wirft aber weitere Fragen auf, die im nächsten Abschnitt detailliert behandelt werden.

Kapitel 3

Strukturelle Eigenschaften

Zeitmanagement ist Unsinn. Sie können die Zeit nicht managen - nur Ihr Verhalten.

Michael Kastner (*1946)
dt. Psychologe u. Hochschullehrer

In diesem Kapitel sollen die bereits erwähnten, für eine Abbildung interessanter struktureller Eigenschaften von Prozessen bzw. Workflows untersucht werden. Es wurde bereits erläutert, dass WfMS strukturelle Merkmale der Workflow-Modelle für eine korrekte Ausführung voraussetzen. Falls erweiternde Aspekte außen vor gelassen werden und Prozesse wie auch Workflows hauptsächlich aus einer zeitlichen Koordination von Elementen betrachtet werden, ist eine strukturelle Klassifikation von Prozessen prinzipiell gleichbedeutend mit einer Klassifikation von Workflows. Deshalb wird angemerkt, dass der Begriff Workflow in diesem Fall synonym für Prozessmodell bzw. Workflow-Modell verwendet wird. Dies erklärt sich aus dem Umstand, dass ein Workflow die betont operative technische Sicht eines Prozesses ist. Da das Ziel dieser Arbeit eine Abbildung eines Prozessmodells auf einen Workflow ist, ist die Semantik der Elemente in Bezug auf die Ausführung im Besonderen von Bedeutung.

3.1 Strukturelle Klassifizierung von Workflows

Die folgenden Passagen werden eine Klassifizierung von Workflows vornehmen, um Aussagen über ihr Verhalten und Eigenschaften treffen zu können. Workflows können anhand ihrer unterschiedlichen strukturellen Restriktionen unterschieden werden. Die mächtigste Klasse sind die strukturell unbeschränkten Workflows. Sie unterliegen keinerlei Beschränkung und besitzen deshalb eine hohe Ausdruckskraft. Aber es obliegt der Verantwortung des Modellierers einen Workflow korrekt, also z.B. ohne Deadlocks, zu modellieren und ist deshalb für die Anwendung in unternehmenskritischen Bereichen keinesfalls zu empfehlen. Strukturell beschränkte Notationen hingegen besitzen strukturelle Einschränkungen in der Modellierung eines Workflows. Sie besitzen aber den Vorteil, dass sie durch ihre Struktur per Definition schon Eigenschaften, wie zum Beispiel die Verhinderung von Deadlocks, besitzen.

Um theoretische Grundlagen zu schaffen, und die Ergebnisse nicht durch spezielle Eigenschaften einer bestimmten Notation zu verfälschen, sollen im Folgenden zwei Metanotationen, die auf gemeinsamen Elementen basieren, zur Modellierung von unstrukturierten und blockstrukturierten Workflows definiert werden.

Die Metanotationen bilden die Grundlage für die theoretischen Ergebnisse der folgenden Abschnitte und beschränken sich dabei auf reine Kontrollaspekte ohne syntaktische Zusätze, die in Notationen in der Praxis zwangsläufig von Nöten sind. Außerdem ist durch die Schaffung entsprechender Grundlagen in diesen Notationen eine Transportierung der Ergebnisse für die beiden Notationen BPMN und ADEPT gut möglich.

3.1.1 Unstrukturierte Workflows

Definition 3.1 Ein *unstrukturierter Workflow* ist ein aus Aktivitäten, Kontrollelementen und Kanten bestehender gerichteter Graph (nach [15]). Formal definiert ist ein unstrukturierter Workflow ein Tripel $\Gamma_u = \{A, E, K\}$. Es sei A die Menge der Aktivitäten, E die Menge der Kanten und K die Menge der Kontrollelementen.

Es gelten folgende Annahmen:

1. Jedes Kontrollelement $\kappa \in K$ ist von einem der folgenden Typen: Start, Ende, Split-Parallel¹, Join-Parallel², Split-Auswahl, Join-Auswahl.
2. Kanten werden benützt, um Aktivitäten und Kontrollelemente zu verbinden. Eine Kante $e \in E$ von Element u zu Element v ($u, v \in A \cup K$) wird durch $\epsilon = (u, v)$ repräsentiert.
3. Der Eingangsgrad $d^-(n)$ gibt die Anzahl der Kanten an, die in einen Knoten (Kontrollelement oder Aktivität) n innerhalb eines Modells einmünden. Der Ausgangsgrad $d^+(n)$ gibt an, wie viele Kanten von diesem Knoten n ausgehen.
4. Jedes Modell hat einen Start- und einen Endknoten. Es gilt für einen Startknoten $g \in K$ vom Typ Start: $d^-(g) = 0$ und $d^+(g) = 1$. Für einen Endknoten $f \in K$ vom Typ Ende gilt: $d^-(f) = 1$ und $d^+(f) = 0$. Die Beschränkung auf einen Startknoten bzw. Endknoten ist keine Einschränkung der Ausdrucksmächtigkeit, da das Verhalten von mehreren parallelen Startknoten oder Endknoten durch Kontrollelemente nachbildbar wäre.
5. Jede Aktivität $a \in A$ hat genau einen Vorgänger und einen Nachfolger, es folgt: $d^-(a) = d^+(a) = 1$.
6. Ein Kontrollelement $s \in K$ vom Typ Split-Parallel oder Split-Auswahl hat genau einen Vorgänger und zwei Nachfolger: $d^-(s) = 1$ und $d^+(s) = 2$. Ein Kontrollelement $j \in K$ vom Typ Join-Parallel oder Join-Auswahl hat genau zwei Vorgänger und einen Nachfolger: $d^-(j) = 2$ und $d^+(j) = 1$. Somit kann durch den Eingangs- und Ausgangsgrad bestimmt werden, ob ein Kontrollelement ein Split- oder

¹engl. to split = sich teilen

²engl. to join = verbinden

ein Join-Element ist.

Die Festlegung auf zwei eingehende bzw. ausgehenden Kanten ist keine Einschränkung, da Kontrollelemente hintereinander ausführbar sind, und dient rein der besseren Anschaulichkeit in diesem Kapitel.

7. Es wird festgelegt, dass in mindestens einem Pfad von einem Split- zu einem Join-Element eine Aktivität vorhanden sein muss.
8. Jede Aktivität bzw. jedes Kontrollelement liegt auf mindestens einem Pfad vom Start- zum Endknoten. Folglich gibt es keine Elemente, die nicht mit dem Workflow verbunden sind.

Die Elemente eines von uns definierten Workflows wurden in der folgenden Abbildung 3.1 graphisch dargestellt.

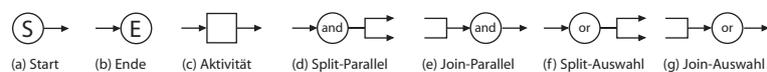


Schaubild 3.1: Graphische Notation der Workflow-Elemente

3.1.1.1 Ausführungssemantik

Um eventuellen Missverständnissen vorzubeugen und die Vorstellung der Notationen BPMN und ADEPT zu erleichtern, soll auf die Semantik der definierten Elemente detaillierter eingegangen werden.

Die Kanten eines Workflows dienen als Kontrollkanten, die den zeitlichen Ablauf innerhalb eines Workflows definieren. Beispielsweise wird bei einer Kante von Aktivität A nach Aktivität B, B erst dann ausgeführt, wenn A erfolgreich beendet wurde. Die nach einem Split-Parallel nachfolgenden Pfade werden zeitlich parallel ausgeführt. Analog folgt, dass ein Join-Parallel nur dann ausgeführt werden kann, wenn beide eingehenden Pfade als erfolgreich ausgeführt markiert wurden. Wir setzen außerdem voraus, dass nach einem Split-Auswahl-Kontrollelement genau ein Pfad ausgeführt wird. Dies entspricht einer sogenannten XOR-Semantik. Eine Or-Semantik, also einer Auswahl m aus n Pfaden, wobei $1 \leq m \leq n$, ist durch die Kombination von Split-Parallel- und Split-Auswahl-Kontrollelementen darstellbar.

Für das Join-Auswahl-Kontrollelement ergeben sich für die Bewertung der eingehenden Kanten zwei Möglichkeiten:

1. *Diskriminator - Einfache Ausführung*: Das Join-Auswahl Element wird genau dann einmal ausgeführt, wenn der erste eingehende Pfad ausgeführt wurde. Der zweite Pfad werden nach seiner Ausführung nicht beachtet und verworfen.
2. *Mehrfache Ausführung*: Jede erfolgreich ausgeführte eingehende Kante bewirkt das Ausführen des Kontrollelementes. Dabei werden die Aktivitäten nach dem Kontrollelement eventuell mehrmals ausgeführt, was zu Korrektheitsproblemen des Workflows führen kann.

Eine Untersuchung der Folgen der unterschiedlichen Semantik des Join-Auswahl Elementes findet in Abschnitt 3.2.1.2 statt.

3.1.2 Blockstrukturierte Workflows

Ein blockstrukturierter Workflow soll aus den Elementen, die bereits durch vorige Definition der strukturell unbeschränkten Workflows bekannt sind, definiert werden. Obwohl die Syntax und Semantik der Elemente identisch sind, wird durch die Form der Definition eine Struktur in der Anordnung der Elemente explizit vorgegeben.

Definition 3.2 Ein *blockstrukturierter Workflow* $\Gamma_s = \{A, E, K\}$ ist ebenfalls ein Tripel aus Aktivitäten, Kontrollelementen und Kanten. Der Aufbau der Modelle sei jedoch über die Kontrollelemente Start S und Ende E ($S, E \in K$) und einen strukturellen Platzhalter X definiert. Es existieren eine Kante von S zu X und eine Kante von X zu E . Dieser Sachverhalt ist zur Anschauung in Schaubild 3.2 dargestellt.



Schaubild 3.2: Strukturierter Workflows – Basisstruktur

Der Platzhalter X kann durch eine einzelne Aktivität oder durch eines der Konstrukte, die im Folgenden definiert werden, ersetzt werden. Es ist außerdem möglich, X durch eine Kante vom Kontrollelement Start zum Kontrollelement Ende zu ersetzen, und somit einen Workflow ohne Aktivitäten und nur einem Start und Ende zu konstruieren.

Für die Definition der Konstrukte werden die Variablen Y und Z ebenfalls als Platzhalter benutzt. Sie können wie X wiederum durch eines der Konstrukte oder durch eine einzelne Aktivität ersetzt werden. Des Weiteren ist es möglich, innerhalb der Sequenz-, der Auswahl- und der Schleifen-Struktur, eine der Variablen (also entweder Y oder Z) inklusive der einmündenden und ausgehenden Kante durch eine Kante vom Vorgänger- zum Nachfolger-Element zu ersetzen. Beispiele hierzu sehen Sie in der Abbildung 3.4.

- Seien Y und Z strukturelle Platzhalter. Eine Konkatenation der strukturellen Platzhalter Y und Z , bei der eine Kante von Y nach Z führt, wird Sequenz genannt.
- Seien $s \in K$ ein Split-Auswahl Element, $j \in K$ ein Join-Choice Element, Y und Z strukturelle Platzhalter. Das Modell mit den Kanten (s,Y) , (Y,j) , (s,Z) und (Z,j) stellt eine Auswahl-Struktur dar.
- Seien $s \in K$ ein Split-Parallel Element, $j \in K$ ein Join-Parallel Element, Y und Z strukturelle Platzhalter. Ein Konstrukt mit den Kanten (s,Y) , (Y,j) , (s,Z) und (Z,j) stellt eine Parallele-Struktur dar.

- Seien $j \in K$ ein Join-Auswahl Element, $s \in K$ ein Split-Auswahl Element, Y und Z strukturelle Platzhalter. Ein Konstrukt mit den Kanten (j,Y) , (Y,s) , (s,Z) und (Z,j) stellt eine Schleifen-Struktur dar.

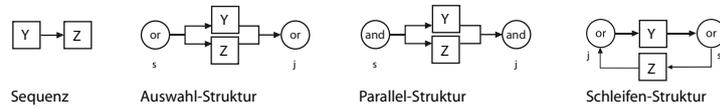


Schaubild 3.3: Strukturierte Workflows – Konstrukte

Wie bereits angeführt, Beispiele für Variationen der Basiskonstrukte als Schaubild:

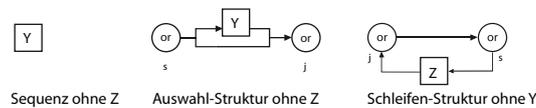


Schaubild 3.4: Strukturierte Workflows – Variationen der Konstrukte

Die Definition der Platzhalter Y und Z verhindern das Benötigen sogenannter „Null-Aktivitäten“, um zum Beispiel den direkten Weg eines Split-Elements zu einem Join-Element zu simulieren.

Um anschaulich zu verdeutlichen, dass ein blockstrukturiertes Modell aus den korrekt ineinander verschachtelten Grundkonstrukten besteht und diese dann „Blöcke“ bilden, wurde Schaubild 3.5 gewählt.

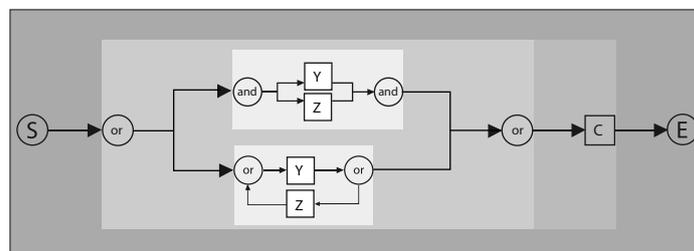


Schaubild 3.5: Strukturierte Workflows - Veranschaulichung der Blockstruktur

Es ist anzumerken, dass jede Aktivität innerhalb eines blockstrukturierten Workflows wie aus Schaubild 3.5 auch selbst wiederum einen blockstrukturierten Workflow darstellen kann.

Diese hierarchische Verschachtelung, in der eine Aktivität auch ein *Subworkflow* bzw. ein *Subprozess* darstellen kann, ist nicht auf blockstrukturierte Modelle beschränkt und kann natürlich auch in unstrukturierten Modellen existieren.

3.1.3 Weitere Formen

Natürlich existieren Zwischenformen innerhalb der Spanne zwischen den strukturell unbeschränkten unstrukturierten Workflows und den strukturell stark eingeschränkten blockstrukturierten Workflows. Einige WfMS, wie z.B. IBM MQ Series/Workflow und InConcert, lassen die Modellierung strukturell unbeschränkter azyklischer Workflows zu, während die Modellierung von Schleifen auf ein vorgefertigtes Konstrukt eingeschränkt wird [16]. Die Klasse der Workflows mit eingeschränkten Schleifen (engl. Restricted Loop Workflow Models, RLWM) soll aber nicht Gegenstand weiterer Untersuchungen sein. Der Grund hierfür ist, dass die Ergebnisse dieser Untersuchung auf diese Klasse portierbar sind. Im folgenden Abschnitt wird dies ersichtlich werden.

3.2 Bewertung und Folgen struktureller Eigenschaften

Ein wesentliches Bestandteil der blockstrukturierten Workflows ist es, dass durch den Umstand der Einschränkung der Modellierung auf blockstrukturierte Konstrukte eine eins-zu-eins Beziehung zwischen jedem Split- und Join-Element desselben Typs gibt. In einem unstrukturierten Workflow kann diese Beziehung ebenfalls auftreten, ist aber keineswegs eine Voraussetzung. Folglich kann festgestellt werden, dass die Klasse der blockstrukturierten Workflows eine Teilmenge der Klasse der strukturell unbeschränkten Workflows ist. Um einen Zusammenhang zur Ausdrucksmächtigkeit einer Notation herzustellen, soll dieser Begriff erläutert werden.

Definition 3.3 Die *Ausdrucksmächtigkeit* einer Notation sei definiert als Anzahl der verschiedenen Modelle, die in einer Notation modellierbar sind.

Es folgt somit, dass unstrukturierte Workflows eine höhere Ausdrucksmächtigkeit besitzen als blockstrukturierte Workflows. Diese Sachverhalte ergeben sich unmittelbar aus den Definitionen des blockstrukturierten Workflows und bedürfen keines Beweises. Eine Einordnung der Klasse der Workflow-Modelle mit beschränkten Schleifen als Teilmenge der strukturell unbeschränkten Workflows und als Obermenge der blockstrukturierten Workflows ist ebenfalls in der Literatur gegeben [16]. Insgesamt ergibt sich ein Zusammenhang dieser drei Klassen, der in Schaubild 3.6 grafisch illustriert wurde.

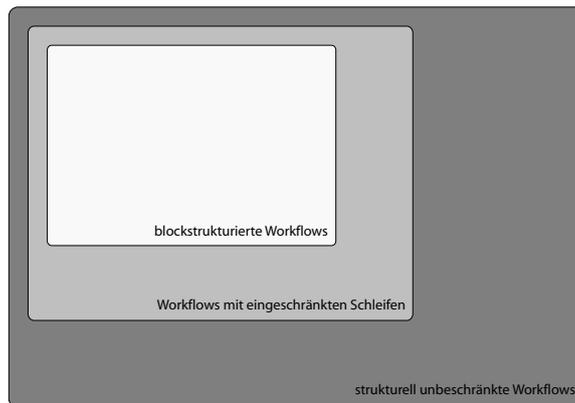


Schaubild 3.6: Zusammenhang der Workflow-Klassen

Eine Untersuchung der Abbildungsmöglichkeiten von unstrukturierten Modellen auf blockstrukturierte schließt durch diesen Zusammenhang die Untersuchung der Abbildung der Klassen der Workflows mit eingeschränkten Schleifen auf blockstrukturierte Workflows mit ein.

Der unterschiedliche Grad an Struktur innerhalb der Workflow-Modelle impliziert Folgen für die Modellierung, die im folgenden Abschnitt untersucht werden sollen. Es wird auf die zwei Hauptprobleme in diesem Zusammenhang, namentlich *Deadlocks* und *multiple Instanzen einer Aktivität* im Detail eingegangen.

Abschließend soll ein Zusammenhang zwischen Ausdrucksmächtigkeit und Sicherheit gegenüber Fehlern aus strukturellen Mängeln hergeleitet werden, um das Kapitel abzurunden.

3.2.1 Folgen struktureller Mängel

Durch die, gegenüber blockstrukturierten Workflows, erweiternden Möglichkeiten der Modellierung ist es in unstrukturierten Workflows möglich, Konstrukte zu modellieren, die schwerwiegende Folgen für die spätere Instanziierung der Modelle haben können. Gerade bei Workflows, die in einem hohem Maße wichtig für den Geschäftsbetrieb eines Unternehmens sind, kann dies schwerwiegende Folgen mit sich ziehen. So können beispielweise Produktionsausfälle und Datenverluste mit den entsprechenden Folgen für ein Unternehmen verursacht werden. Bei Workflows im klinischen Umfeld sind Laufzeitfehler eventuell mit noch gravierenderen Konsequenzen verbunden. Es ist deshalb notwendig mit gebotener Sorgfalt zu prüfen, ob ein Workflow strukturelle Mängel besitzt und welche Folgen diese Mängel mit sich ziehen. Im Folgenden wird auf die zwei Hauptprobleme im Näheren eingegangen.

3.2.1.1 Deadlocks

Durch die mangelnde Vorschrift an Struktur für unstrukturierte Workflows, kann der Modellierer Konstrukte modellieren, die zu Verklemmungen (engl. Deadlocks) führen. Verklemmungen haben zur Folge, dass der Endzustand, also hier das Kontrollelement Ende, niemals erreicht werden kann. Das Schaubild 3.7 zeigt einen einfachen Workflow, der in dieser Form niemals den Endzustand erreichen wird. Der Join-Parallel kann erst Aktivität C aktivieren, wenn in den beiden einmündenden Pfaden alle Aktivitäten, in diesem Fall A und B, ausgeführt wurden. Da nur ein Pfad durch den Split-Auswahl zur Ausführung gebracht wird, kann somit Aktivität C niemals ausgeführt werden.

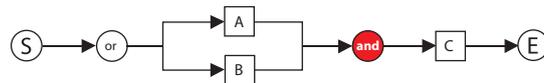


Schaubild 3.7: Erste Deadlock-Variante

Das Modell in Schaubild 3.8 zeigt einen verschachtelten Workflow, in dem der letzte Join-Parallel ebenfalls zwei erfolgreich ausgeführte Pfade erwartet. Durch den rot gekennzeichneten Split-Auswahl wird aber nur ein Pfad ausgeführt werden. Der Workflow kann somit nicht terminieren.

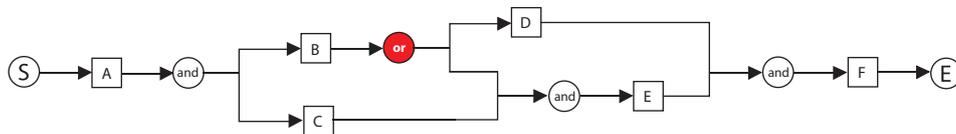


Schaubild 3.8: Zweite Deadlock-Variante

Einen Sonderfall spiegelt der folgende Workflow in Schaubild 3.9 wider. Ein Schleife, die in dieser Form durch einen Join-Parallel und einen Split-Parallel konstruiert wurde, führt ebenfalls dazu, dass der Workflow nicht terminieren kann. Da der Join-Parallel auf die beiden eingehenden Kanten wartet, aber eine Kante nur ausgeführt werden kann, wenn der Join-Parallel bereits geschaltet wurde, ist eine Verklemmung die Folge.

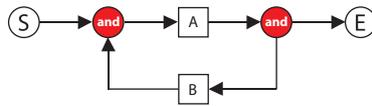


Schaubild 3.9: Dritte Deadlock-Variante

3.2.1.2 Multiple Instanzen einer Aktivität

Neben Deadlocks, die eine Terminierung des Workflows verhindern, ergeben sich durch die fehlende strukturelle Einschränkung bei unstrukturierten Workflows eine weitere Problemquelle. Es wurde bereits erwähnt, dass der Join-Auswahl semantisch einerseits als Diskriminator wirken kann oder andererseits eine mehrfache Ausführung unterstützt. Falls der Join-Auswahl eine mehrfache Ausführung unterstützt, ergeben sich durch die Modellierung bestimmter Konstrukte innerhalb eines Modells Folgen, die in der Literatur als Problem der *multiphen Instanzen einer Aktivität* oder als Problem der *fehlenden Synchronisation* bezeichnet werden.

Der Workflow in Schaubild 3.10 soll dieses Problem illustrieren. Nach dem Split-Parallel werden beide Pfade aktiviert und die Aktivitäten A und B ausgeführt. Nachdem die Aktivitäten erfolgreich beendet wurden, wird Aktivität C zweimal ausgeführt, da der Join-Auswahl bei entsprechender Semantik der mehrfachen Ausführung Aktivität C zweimal aktiviert. Besonders kritisch ist dieser Fall, falls Aktivität C nun auf Ressourcen wie Daten zugreift. So kann es beispielsweise zu unvorgesehenem Verhalten kommen, falls nachdem Aktivität C das erste Mal beendet wurde und die Ressource für eine andere Aktivität freigegeben wird, aber die zweite Instanz der Aktivität C nochmals auf die Ressource zugreift. Ebenfalls kritisch zu sehen ist der Fall, dass nach dem Beenden der ersten Instanz von C, der Workflow terminiert werden soll, obwohl eine zweite Instanz der Aktivität C eventuell noch nicht beendet wurde.

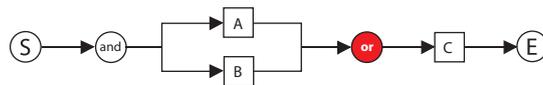


Schaubild 3.10: Multipl. Instanzen einer Aktivität – erste Variante

In Schaubild 3.11 treten prinzipiell die gleichen Probleme wie in Schaubild 3.10 auf. Es soll aber auch aufgezeigt werden, dass es, wie in Schaubild 3.8, schwieriger ist, das Problem zu identifizieren.

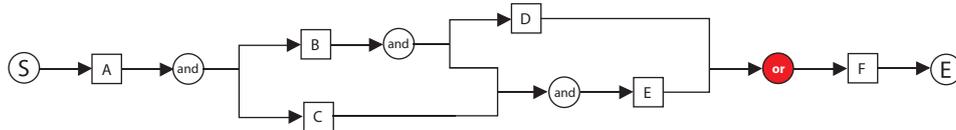


Schaubild 3.11: Multipl. Instanzen einer Aktivität – zweite Variante

Das Schaubild 3.12 zeigt eine Schleifenvariante, die ebenfalls zu multiplen Instanzen einer einzelnen Aktivität führen kann. Ein weiteres Problem ist es, dass diese Schleife niemals beendet werden kann und somit eine Endlosschleife innerhalb des Workflows darstellt.

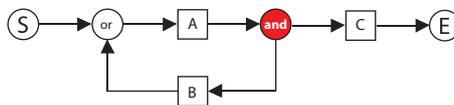


Schaubild 3.12: Multipl. Instanzen einer Aktivität – dritte Variante

3.2.2 Zusammenhang Ausdrucksmächtigkeit und Sicherheit

Durch die vorgestellten Folgen struktureller Mängel kann es also vorkommen, dass Workflows nicht erfolgreich terminieren können. Eine erfolgreiche Beendigung eines Workflow beinhaltet das Erreichen des Kontrollelements Ende, ohne dass weitere Aktivität zu diesem Zeitpunkt vor der Ausführung stehen oder in diesem Moment ausgeführt werden. Es wird deshalb der Begriff *well-behaved* (zu deutsch *brav, artig*) analog zu [17] verwendet:

Definition 3.4 Ein Workflow ist *well-behaved*, wenn er unter allen Ausführungsmöglichkeiten erfolgreich terminiert.

Jeder blockstrukturierte Workflow ist automatisch *well-behaved*. Dies folgt durch den Umstand, dass blockstrukturierte Workflows rein aus der Komposition der Grundkonstrukte bestehen und diese nicht zu Deadlocks führen können [16]. Die Umkehrung dieses Satzes gilt aber nicht. Es gibt also Workflows, die nicht blockstrukturiert sind, aber dennoch *well-behaved* sind.

Ein Beispiel für diesen Sachverhalt gibt der Workflow in Schaubild 3.13.

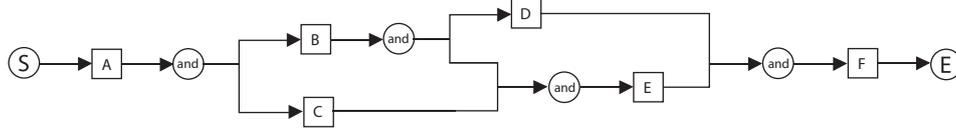


Schaubild 3.13: Beispiel - Well-behaved und nicht blockstrukturierter Workflow

Die Behauptung, dass dieser Workflow keine blockstrukturierte, äquivalente Form besitzt, soll bewiesen werden.

Beweis 1 Als Beweis ist zu betrachten, dass die Aktivitäten B und C nicht voneinander abhängig sind. Sie müssen deshalb in einer blockstrukturierten Variante des Workflows in einer parallelen Struktur auf verschiedenen Pfaden sein. Es ist ebenfalls zu beobachten, dass die Aktivität D von Aktivität B abhängt und die Aktivität E von Aktivität C abhängt. Deshalb muss Aktivität D innerhalb der parallelen Struktur nach B kommen und Aktivität E nach C. Aktivität D kann nicht außerhalb der Struktur sein, da sie dann auch von C abhängen würde. Falls E außerhalb der Struktur wäre, würde diese Aktivität erst nach Aktivität D ausgeführt werden. Die bisherigen Erkenntnisse sind zum besseren Verständnis im folgenden Schaubild 3.14 dargestellt.

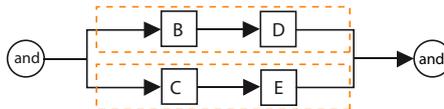


Schaubild 3.14: Beweisskizze unstrukturierte artige Workflows

Da aber Aktivität E von den Aktivitäten B und C abhängt, ist ein Widerspruch gegeben. Denn Aktivität E muss sich, wie bereits erklärt, innerhalb der parallelen Struktur befinden. Es gibt also für diesen unstrukturierten Workflow kein blockstrukturiertes Äquivalent.

Im folgenden Kapitel wird im Rahmen der Vorstellung von ADEPT die Erweiterung der blockstrukturierten Notationen um Synchronisationkanten aufgezeigt. Mit Zuhilfenahme dieser Kanten ist es möglich, einen blockstrukturierten Workflow mit den entsprechenden Eigenschaften zu modellieren, der äquivalent zum Workflow in Schaubild 3.13 ist. Nach dem Beweis, dass es Workflows gibt, die *well-behaved*, aber nicht blockstrukturiert sind, soll ein Zusammenhang zwischen den strukturellen Einschränkungen, die zu einer Verminderung der Ausdrucksstärke einer Notation führen, und der wachsenden Sicherheit gegenüber einer Modellierung von Laufzeitfehlern verursachenden Konstrukten hergestellt werden.

Es wurde bereits erläutert durch welche Konstrukte Deadlocks oder multiple Instanzen einer einzelnen Aktivität auftreten können. Äquivalente blockstrukturierte Abbildungen dieser Konstrukte sind nicht möglich, da diese ebenfalls zu Deadlocks oder multiple Instanzen führen müssten und diese in einer blockstrukturierten Notation nicht

vorkommen können. Außerdem verhindern die Klasse der Workflows mit beschränkten Schleifenkonstrukten die Modellierung von Deadlocks und multiplen Instanzen über Schleifenkonstrukten. Es werden also die Möglichkeiten zur Modellierung von Laufzeitfehlern eingeschränkt, wenngleich nicht vollkommen ausgeschlossen.

Aus diesen Erkenntnissen folgt, dass die Sicherheit einer Notation vor Modellierung von Laufzeitfehler S_N wie z.B. Deadlocks reziprok proportional zur Ausdrucksmächtigkeit einer Notation A_N ist. Der Sachverhalt, illustriert in Schaubild 3.15, ist rein von qualitativer Aussage und keinesfalls quantitativ zu verstehen.

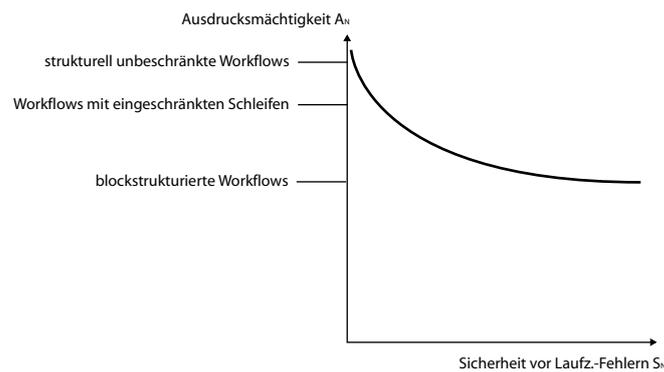


Schaubild 3.15: Vergleich Ausdrucksmächtigkeit u. Sicherheit v. Notationen

Mit dieser Erkenntnis soll dieses Kapitel abgeschlossen werden. Eine im höchsten Maße ausdrucksstarke Notation, die gleichzeitig das Modellieren von Laufzeitfehlern verhindert, kann somit als konzeptioneller Widerspruch angesehen werden.

Kapitel 4

Modellierung

Niemand plant, zu versagen, aber die meisten versagen beim Planen.

Lee Iacocca (*1924)
amerik. Manager und Rhetoriker

Nach den theoretischen Ausführungen über den Zusammenhang von Ausdruckstärke und Sicherheit vor Laufzeitfehlern, sowie der theoretischen Einführung der unstrukturierten wie blockstrukturierten Notationen in Kapitel 3, sollen nun die zwei Notationen BPMN und ADEPT¹ vorgestellt werden. Weiterhin werden die Ergebnisse aus Kapitel 3 auf diese Notationen übertragbar. Zuvor soll ein kurzer Überblick über die verschiedenen Möglichkeiten der Modellierung erfolgen.

4.1 Bekannte Formen

Bei der Wahl einer Notation zu Modellierung von Prozessen oder Workflows stellt sich die Frage nach dem Zweck der Modellierung und ebenfalls dem fachlichen Hintergrund des Modellierers. Für eine Erfassung, Analyse und Optimierung von Geschäftsprozessen auf strategischem Level ist eine Notation zur Modellierung von Workflows möglicherweise zu überladen und bietet eventuell zu wenig Abstraktionsmöglichkeiten. Dazu können, wie bereits angeführt, die Details und strukturellen Beschränkungen, die für eine spätere Ausführung benötigt werden, ein Hindernis darstellen. Jedoch enthalten die bereits erwähnten BPMS proprietäre Notationen, die zur Modellierung der strategischen Geschäftsprozesse und späteren Verfeinerung dienen sollen. Bekannte herstellerunabhängige Notationen, die sich zur Modellierung von Geschäftsprozessen auf strategischem Level anbieten sind die bereits erwähnte BPMN, die in diesem Kapitel detaillierter vorgestellt wird, die ereignisgesteuerten Prozessketten (EPK), die wesentlicher Bestandteil des ARIS-Konzeptes sind und die UML Aktivitätendiagramme [2].

Auf eine explizite Nennung herstellerbezogene Notationen zur Modellierung von

¹ADEPT steht für *Application Development based on Encapsulated Premodelled Process Templates*

Workflows für ein WfMS soll an dieser Stelle verzichtet werden. Ein Überblick bekannter Notation zur Modellierung von Workflows ist beispielsweise in [18] und [19] gegeben. In diesen Arbeiten werden die Notation unter anderem anhand von Workflow-Patterns [13] bezüglich ihrer Ausdrucksmächtigkeit und Angemessenheit evaluiert. Um Ausführungseigenschaften der Workflows zu analysieren beruhen einige Notationen auf formale Modelle, wie beispielsweise die Petri-Netze [20].

Das WfMS ADEPT2 [21, 22] des Instituts für Datenbanken und Informationssysteme der Universität Ulm besitzt mit der Notation ADEPT eine eigene Notation zur Modellierung von Workflows, die ebenfalls auf einer fundierten formalen Grundlage beruht [23]. In ADEPT wurde neben der Sicherheit der Workflows vor Laufzeitfehlern, wie Deadlocks, der Schwerpunkt auf ein, für eine blockstrukturierte Notation, hohes Maß an Flexibilität gelegt. Eine detaillierten Vorstellung erfolgt in Abschnitt 4.3.

4.2 BPMN

Die BPMN wurde ursprünglich von der BPMI (Business Process Management Initiative) entwickelt [3]. Diese schloss sich 2003 mit der OMG zusammen, so dass die BPMN unter dem Dach der 1993 gegründeten OMG weiterentwickelt wird. Die Notation setzt ihren Schwerpunkt auf den Kontrollfluss. Die momentan aktuelle Version 1.1 ist datiert auf den 6. Februar 2006, eine verbesserte Version 2.0 ist in Planung. Die Ziele der OMG [24] waren es, eine Notation zu spezifizieren, die von allen Beteiligten eines Geschäftsprozesses schnell verstanden wird. Nach einer Analyse in [4] wurde dieses Ziel erreicht, da BPMN im geschäftlichem Umfeld (u.a. Prozessdokumentation, Prozessoptimierung, Unternehmensanalyse) wie im technischen Bereich (u.a. Prozesssimulation) gleichermaßen Akzeptanz findet. Nach der Auffassung der OMG soll BPMN sich zum Standard für die Modellierung von Prozessen entwickeln, vergleichbar in der Bedeutung mit der Unified Modeling Language (UML) für objekt-orientiertes Design und Analyse [8]. Im Juni 2008 unterstützen laut der OMG (www.bpmn.org) 47 Hersteller von Werkzeugen zur Modellierung von Geschäftsprozessen BPMN. Es ist zwar möglich, neben abstrakten Prozessen auch bis zu einem gewissen Grad operative Prozessmodelle zu modellieren, jedoch unterstützt die BPMN nicht nativ die direkte Ausführung der Modelle. Doch existieren bereits Abbildungen [25, 26, 27] zum Beispiel zur ausführbaren Business Process Execution Language for Web Services (BPEL4WS oder BPEL, [28]), dem de facto Standard in der Implementierung von Geschäftsprozessen auf Basis von Web-Services [12]. Es kann aber festgehalten werden, dass es ohne Nacharbeiten nicht möglich ist, operative Prozesse mit BPMN zu modellieren und diese direkt nach der Abbildung in BPEL auszuführen. Eine Arbeit bezüglich den konzeptionellen Unterschieden der BPMN und BPEL stellt [27] dar.

Die Modelle werden in der BPMN in einem sogenannten Business Process Diagramm erfasst, das aus den BPMN-Elementen besteht. Im Folgenden werden nur die Kernelemente der BPMN, die relevant für eine spätere Untersuchung der Abbildung sind, vorgestellt. Analysen und Bewertungen der BPMN und die Spezifikationen der Elemente der BPMN sind in [3, 29, 8, 30] zu finden.

4.2.1 Elemente

Viele der folgenden Elemente haben ein Äquivalent in der bereits in Kapitel 3 vorgestellten unstrukturierten Metanotation. Eine deutsche Kurzübersicht über alle vorhandenen Elemente der BPMN inklusive kurzen Erläuterungen ist auf der Webseite des Hasso-Plattner-Instituts² der Universität Potsdam zu finden.

Analog zur Metanotation besitzt ein Modell in BPMN ein Start- und ein End-Ereignis und natürlich Aktivitäten. Bezüglich der Möglichkeiten der Anordnung der Elemente in der BPMN sei an dieser Stelle auf die Spezifikation in [3] verwiesen. Im Folgenden werden rein die für eine kontrollflussbasierte Untersuchung der Abbildung relevanten Elemente gemäß der Spezifikation vorgestellt. In jedem Abschnitt wird durch ein Schaubild die Symbolik illustriert und die Bedeutung durch Ergänzungen festgelegt.

4.2.1.1 Aktivitäten

Eine Aktivität wird in der BPMN als Rechteck mit gerundeten Kanten dargestellt. Unter anderem haben zum Beispiel Subprozesse ebenfalls die gleiche Erscheinung. Diese sind aber mit einem Plus in einem kleinen Quadrat innerhalb des Rechteckes zusätzlich gekennzeichnet. Schaubild 4.1 zeigt eine Aktivität in BPMN.

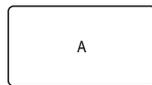


Schaubild 4.1: BPMN – Aktivität

4.2.1.2 Kanten

Es gibt in BPMN drei verschiedene Kanten. Den Sequenzfluss, den bedingten Fluss, dem eine Bedingung angeheftet ist, und den Standardfluss. Der bedingte Fluss ermöglicht es in BPMN, eine Verzweigung ohne ein Kontrollelement zu realisieren und dieser Verzweigung eine Bedingung zu Grunde zu legen. Die Verzweigung ohne Gateway wird in Abschnitt 4.2.1.4 behandelt. Mit dem Standardfluss ist es möglich, bei einer Verzweigung den Pfad zu kennzeichnen, der gewählt werden soll, wenn die Bedingungen der restlichen Pfade nicht zutreffen. Der Sequenzfluss ist jedoch die Kante der Wahl um auszudrücken, dass eine Aktivität nach einer anderen ausgeführt werden soll. Sie ist die Kante, die wir in der Untersuchung der Abbildbarkeit berücksichtigen werden, da die weiteren Kanten eine Untersuchung der Abbildung komplexer werden lassen.

²dt.BPMN-Poster:http://bpt.hpi.uni-potsdam.de/pub/Public/BPMNCorner/BPMN1_1_Poster_DE.pdf

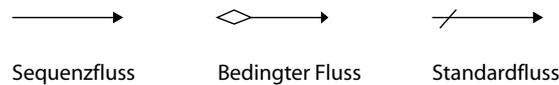


Schaubild 4.2: BPMN – Kanten

4.2.1.3 Ereignisse

Um in BPMN den Start- bzw. das Ende eines Prozesses auszudrücken, werden Ereignisse (auch Events genannt) verwendet. Das *Startereignis* ist ein symmetrischen Kreis mit einfacher Kontur. Das *Endereignis* zur Markierung des Endes eines Prozesses ist ebenfalls ein symmetrischer Kreis. Dieser besitzt aber eine deutlichere Kontur. BPMN erlaubt es die Kreise mit dem Grund des Ereignisse zu füllen, also z.B. könnte das Eintreffen einer Nachricht den Start eines Prozesses auflösen. Auf diese Erweiterungen soll in dieser Arbeit verzichtet werden, da diese von keiner Relevanz für diese Arbeit sind. Ebenfalls weitgehend vernachlässigt werden sogenannte *Zwischenereignisse*, die während eines Prozesses auftreten können oder den Prozess verzögern können. Beispiele für Zwischenereignisse sind das Ereignis des Empfangs oder Versands einer Nachricht. Details zu Ereignissen sind in [3] und in kürzerer Fassung in [8] zu finden.



Schaubild 4.3: BPMN – Ereignisse

4.2.1.4 Gateways

Gateways dienen in BPMN zur Steuerung des Kontrollflusses. Das *parallele Gateway* besitzt wie das *Daten-basierte exklusive Gateway* (im Verlauf der Arbeit schlicht als „exklusives Gateway“ benannt) eine Entsprechung in den Kontrollelementen der Metanotation aus Kapitel 3. So bringt das parallele Gateway, wenn es als Split-Element eingesetzt wird, zum Ausdruck, dass die ausgehenden Pfade parallel ausgeführt werden sollen. Wenn das parallele Gateway als Synchronisierer mehrerer Pfade dient,

wird der ausgehende Pfad nur im Falle der erfolgreichen Ausführung aller eingehenden Pfade aktiviert. Bei der Verwendung des exklusiven Gateway als Split-Element wird genau ein Pfad zur Ausführung ausgewählt, während bei der Verwendung zur Synchronisation der ausgehende Pfad, nachdem ein eingehender Pfad erfolgreich ausgeführt wurde, aktiviert wird. Die restlichen eingehenden Pfade werden nicht ignoriert und somit erlaubt das exklusive Gateway die Ausführungssemantik einer mehrfachen Ausführung. Das exklusive Gateway kann einerseits als Raute ohne Inhalt oder als Raute mit dem Buchstaben „X“ als Markierung dargestellt werden.

Neben diesen zwei für die Untersuchung hauptsächlich relevanten Elemente existieren noch das *Ereignis-basierte exklusive Gateway*, das *inklusive Gateway* und das *komplexe Gateway*. Das Ereignis-basierte exklusive Gateway dient als exklusives Split-Element, das auf den Eintritt von Ereignissen wartet. Das inklusive Gateway erlaubt eine Auswahl m aus n Pfaden (für $1 \leq m \leq n$) sowie das Synchronisieren der entsprechenden Pfade. Das komplexe Gateway als Split-Element ist zur Ausführung von komplexen Bedingungen für die Pfade zu verwenden. Eine Simulation der Bedingungen durch parallele und exklusive Gateways sollte möglich sein, wenngleich diese Umwandlung von komplexer Natur sein kann. Als Join-Element kann das komplexe Gateway, mit den entsprechenden Attributen versehen, zur Verwendung als Diskriminator in BPMN dienen. Das so eingesetzte komplexe Gateway aktiviert somit nach erfolgreicher Ausführung eines Pfades das folgende Element und ignoriert alle weiteren erfolgreich ausgeführten Pfade.

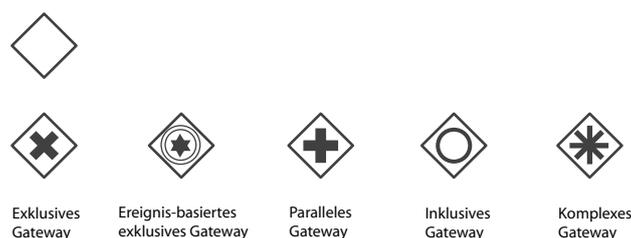


Schaubild 4.4: BPMN – Gateways

BPMN erlaubt im Gegensatz zu der Metanotation aus Kapitel 3 und ADEPT die Modellierung von Verzweigungen und Synchronisationen ohne entsprechende Split- und Join-Elemente. Eine Verzweigung von zwei (oder mehr Kanten) kann in ein paralleles Gateway (mehrere parallele Gateways) transformiert werden. Die Berücksichtigung von bedingtem Fluss und Standardfluss würde in einer Transformation von diesen Kanten in exklusive und eventuell parallele Gateways münden. Schaubild 4.5 zeigt ein Prozessmodell mit einer Verzweigung ohne Gateway. Diese Verzweigung entspricht einer Verzweigung durch ein paralleles Gateway.

Das Schaubild 4.5 zeigt außerdem ein Beispiel für das Zusammenführen zweier Kanten in eine Aktivität ohne ein Gateway. Diese Form der Modellierung ist gleichwertig zur Modellierung mit einem exklusiven Gateway. Dieses Prozessmodell besitzt also das

Problem der multiplen Instanzen einer Aktivität.

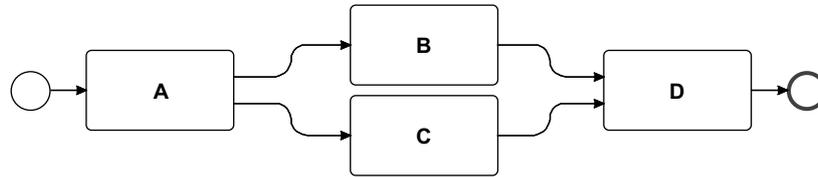


Schaubild 4.5: BPMN – Prozessmodell ohne Gateways

Es sei darauf hingewiesen, dass die Verzweigungen bzw. das Zusammenführung von Kanten ohne Gateways für die Untersuchung der Abbildbarkeit in Kapitel 5 in äquivalente Formen mit Gateways, wie bereits vorgestellt, überführt werden.

4.2.1.5 Daten

Die Dokumentation der Zugriffe auf Daten durch Aktivitäten erfolgt durch *Datenobjekte*. Die Verbindung dieser Objekte zu Aktivitäten wird durch Assoziationen, die als gestrichelte Linien aufgeführt sind, hergestellt. Assoziationen können Richtungen haben, die durch Pfeile ausgedrückt werden. Es kann somit der Schreibvorgang oder der Lesevorgang einer Aktivität auf ein Datenobjekt dargestellt werden.

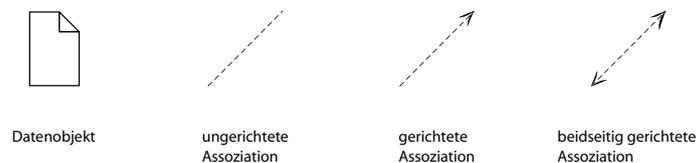


Schaubild 4.6: BPMN – Daten

4.2.1.6 Weitere Elemente

Auf eine detaillierte Vorstellung weiterer Elemente, wie zum Beispiel Swimlanes und Pools, die zur Repräsentation von Rollen dienen, oder Transaktionen, die zur Fehlerbehandlung in Prozessen genutzt werden können, wird an dieser Stelle verzichtet. Der Grund liegt im zeitlichen Rahmen dieser Arbeit.

4.2.2 Grundlegende Modelle in BPMN

Um einen Eindruck in die Modellierung von Modellen durch die BPMN zu vermitteln, werden in den folgenden Schaubildern die Modellierung der blockstrukturierten Grundkonstrukten gezeigt. Außerdem soll ein ausführlicher, unstrukturierter Beispielprozess gezeigt werden. Für die Konstrukte wurde auf das Einfügen der Start- und

Endereignisse verzichtet, da diese bei einer Abbildung in der Regel nicht alleinstehend, sondern innerhalb eines größeren Modelles vorkommen werden.

Die Schaubilder 4.7, 4.8 und 4.9 dürften aufgrund der Vorstellung der Elemente in vorhergehenden Abschnitten und den Ergebnisse aus Kapitel 3 selbsterklärend sein. Eine Besonderheit stellt die Modellierung von Schleifen in BPMN dar. Schleifen können beispielsweise wie in Schaubild 4.10 modelliert werden. Natürlich ist das exklusive Gateway als Join-Element nicht notwendig, da die Kante auch direkt in die folgende Aktivität münden könnte. Es existieren aber auch noch weitere Sonderelemente in der BPMN zur Modellierung von Schleifen. Es darf aber darauf hingewiesen werden, dass diese Elemente sich prinzipiell auf das Konstrukt in Schaubild 4.10 abbilden lassen.

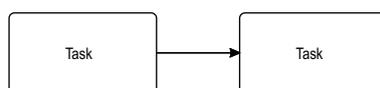


Schaubild 4.7: BPMN – grundl. Modelle – Sequenz

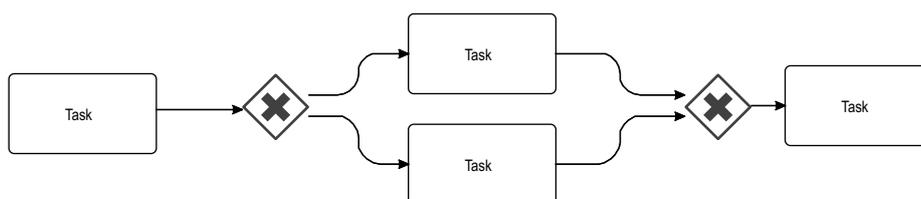


Schaubild 4.8: BPMN – grundl. Modelle – Auswahlstruktur

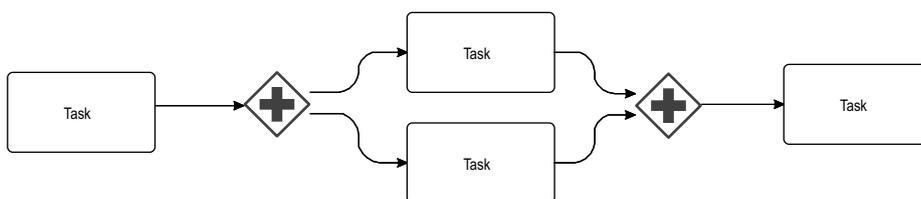


Schaubild 4.9: BPMN – grundl. Modelle – Parallele Struktur

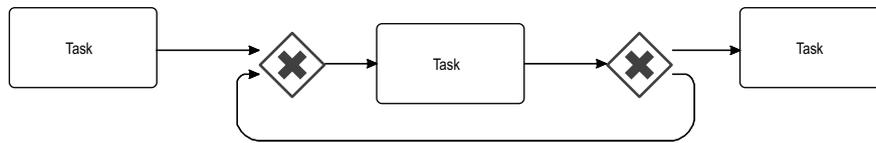


Schaubild 4.10: BPMN – grundl. Modelle – Schleifenmöglichkeit

Um die Modellierung eines kompletten Prozesses in BPMN zu demonstrieren, wurde Schaubild 4.11 ausgewählt. Es wurde bewusst auf ein blockstrukturiertes Modell verzichtet, um die Modellierung unstrukturierter Modelle in BPMN zu demonstrieren.

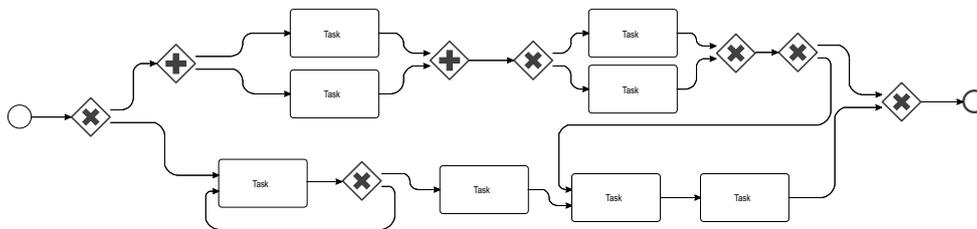


Schaubild 4.11: BPMN – grundl. Modelle – Beispiel eines kompl. Prozesses

4.2.3 Einordnung

Die BPMN schreibt für die Modellierung der Modelle bzw. der Diagramme keine Vorgaben bezüglich der Einschränkung von Schleifen oder der Schaffung von symmetrischen Blöcken vor. Die Ergebnisse für unstrukturierte Workflows aus Kapitel 3 sind damit auf BPMN übertragbar. Damit ist BPMN eine sehr ausdrucksstarke Notation, die keinerlei strukturellen Einschränkungen besitzt, um die Modellierung von Deadlocks, multiplen Instanzen einer Aktivität oder Endlosschleifen zu verhindern.

4.3 ADEPT

Nach der Vorstellung der unstrukturierten Notation BPMN soll nun die blockstrukturierte Notation ADEPT vorgestellt werden. ADEPT entstand im Zusammenhang mit der Entwicklung des gleichnamigen WfMS am Institut für Datenbanken und Informationssysteme der Universität Ulm [31]. Die erste verfügbare Version des Workflow-Management-Systems ADEPT stammt aus dem Jahr 1998. Die Grundlagen für die Ent-

wicklung dieses WfMS liegen unter anderem in Forschungsarbeiten zur Demonstration von Adhoc-Flexibilität in einer blockstrukturierten Notation [23] und der verteilten Ausführung von Workflows [32]. Um neuen Forschungsergebnissen gerecht zu werden, wurde die Entwicklung und Implementierung des WfMS ADEPT2 im Rahmen des AristaFlow-Projektes³ im Jahre 2004 begonnen. Im Jahr 2008 erfolgte die Ausgründung in die AristaFlow GmbH und die Entwicklung zur Produktreife.

In die Implementierung des WfMS ADEPT2 sollen aktuelle Forschungskonzepte einfließen. So soll ADEPT2 durch Eigenschaften bestehen, die in zu dieser Zeit verfügbaren Workflow-Management-Systemen wenn überhaupt nur bedingt unterstützt werden. Beispiele hierfür sind die Prozess-Schema-Evolution, die Prozesskomposition per „plug-and-play“ oder die Beachtung von semantischen Einschränkungen [31]. Das Prinzip der Prozess-Schema-Evolution [33] erlaubt es, Änderungen am Workflowmodell (Schema) vorzunehmen und diese Änderungen, falls der Zustand der Instanzen dies erlaubt, auf bereits laufende Instanzen zu übertragen.

4.3.1 Elemente

Die Modellierung in ADEPT erfolgt wie in BPMN graphisch als gerichteter Graph. Im Gegensatz zu BPMN ist die Menge an Symbolen jedoch überschaubarer. Jedes Modell besteht grundsätzlich aus Knoten und Kanten. Kontrollelemente, die zum Aufbau von z.B. parallelen Strukturen benötigt werden, sind bezüglich ihrer Darstellung an die Knoten angeheftet. Knoten und Kanten werden durch ihren Typ unterschieden. Jedes Modell eines Workflows in ADEPT besitzt ebenfalls wie die Metanotation und BPMN einen Start- und einen End-Knoten. Im Folgenden werden die für eine kontrollflussbasierte Untersuchung der Abbildung relevanten Elemente vorgestellt (nach [23]). In jedem Abschnitt wird durch ein Schaubild die Symbolik illustriert und die Bedeutung durch Ergänzungen erläutert.

4.3.1.1 Knoten

Ein Knoten in ADEPT wird generell als Rechteck dargestellt. Ein Knoten ohne nähere Typbezeichnung sei im Folgenden als Aktivität zu betrachten und entspricht somit der Aktivität der Metanotationen sowie der Aktivität in BPMN. Die Knoten, die den Start bzw. das Ende eines Workflows repräsentieren, besitzen jeweils einen eigenen Typ. So ist der eindeutige Startknoten vom Typ (in Schaubildern annotiert mit NT, engl. nodetype, dt. Knotentyp) „STARTFLOW“ und der eindeutige Endknoten vom Typ „ENDFLOW“. Ebenfalls einen eigenen Typ besitzen die Knoten zum Starten (NT = STARTLOOP) und zum Beenden (NT = ENDLOOP) eines Schleifenkonstruktes. Die Modellierung eines Schleifenkonstruktes wird in Abschnitt 4.3.2 vorgestellt.

³<http://www.aristaflow.de>

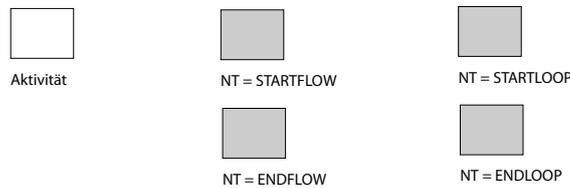


Schaubild 4.12: ADEPT – Knoten

4.3.1.2 Kanten

Kanten können, wie die bereits vorgestellten Knoten, von unterschiedlichem Typ (in Schaubildern annotiert mit ET, engl. edgetype, dt. Kantentyp) sein. Die Kante zur Kennzeichnung der sequentiellen Abfolge zweier Aktivitäten ist vom Typ „CONTROL_E“. Kanten werden generell durch einen einfachen Pfeil mit gefüllte Pfeilspitze dargestellt (vgl. Schaubild 4.13).

Die Kante zur Verbindung des letzten Knotens einer Schleife (NT = ENDLOOP) mit dem ersten Knoten der Schleife (NT = STARTLOOP) besitzt den Typ „LOOP_E“. Weitere existente Kanten sind Kanten, die nach dem Auftreten eines Fehlers bei einer Aktivität einen Rücksprung erlauben (ET = FAILURE_E) und sogenannte Synchronisationskanten. Die Kanten zum Abfangen und Behandeln von Fehlern werden aufgrund des zeitlichen Rahmens nicht behandelt.

Synchronisationskanten erweitern jedoch die Ausdrucksstärke der blockstrukturierten Notation ADEPT. Anschaulich werden die Linien der Synchronisationskanten gestrichelt dargestellt. Sie unterstützen die Synchronisation der zeitlichen Ausführung von Aktivitäten aus unterschiedlichen Zweigen einer parallelen Struktur. Es werden zwei Arten der Synchronisation unterstützt:

- *Sanfte Synchronisation:* Eine Synchronisationskante vom Typ „SOFT_SYNC_E“ von einer Aktivität A zu Aktivität B bedeutet, dass B erst dann ausgeführt wird, wenn entweder A erfolgreich ausgeführt wurde, oder A nicht mehr ausgeführt werden kann.
- *Strikte Synchronisation:* Eine Synchronisationskante vom Typ „STRICT_SYNC_E“ von einer Aktivität A zu Aktivität B bedeutet, dass B erst dann ausgeführt wird, wenn zuvor A erfolgreich ausgeführt wurde.

Um redundante Ausführungseinschränkungen zwischen Aktivitäten zu vermeiden, unterliegt die Nutzung der Synchronisationskanten Einschränkungen. Wie bereits erwähnt sollen nur Aktivitäten aus unterschiedlichen Zweigen einer parallelen Struktur verbunden werden. Weiterhin ist eine Verbindung einer Aktivität innerhalb einer Schleife mit einer Aktivität außerhalb dieser untersagt.

Schaubild 4.13 zeigt einen durch Synchronisationskanten erweiterten blockstrukturierten Workflow in ADEPT, der außerdem äquivalent zum unstrukturierten Prozessmodell aus Schaubild 3.14 (aus Kapitel 3) ist. Durch die Nutzung von Synchronisationskanten ist es folglich möglich, für bestimmte unstrukturierte Prozessmodelle ein blockstrukturiertes äquivalentes Modell zu finden, das lediglich durch Synchronisationskanten erweitert wurde. Synchronisationskanten erweitern also die Ausdrucksmächtigkeit einer blockstrukturierten Notation.

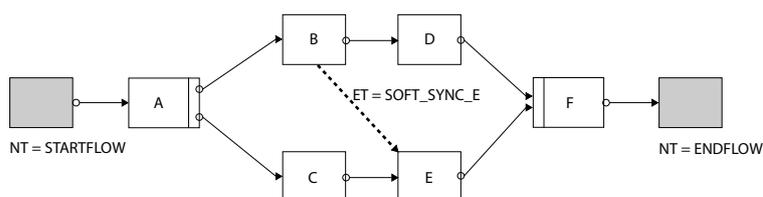


Schaubild 4.13: ADEPT – Beispiel für Synchronisationskanten

4.3.1.3 Kontrollelemente

Im Gegensatz zur BPMN besitzen Elemente zum Aufteilen oder Synchronisieren des Kontrollflusses in ADEPT keine freistehende Symbole, sondern werden an die Aktivitäten angeheftet. Das Split-Element an die vorhergehende Aktivität, das Join-Element an die nachfolgende Aktivität. Es existieren AND- und OR-Splits, sowie AND- und OR-Joins. Im Vergleich zur blockstrukturierten Metanotation entspricht ein OR-Split einem Split-Auswahl, ein OR-Join einem Join-Auswahl, ein AND-Split einem Split-Parallel und ein AND-Join einem Join-Parallel. Die Semantik eines OR-Joins in ADEPT ist die eines Diskriminators (siehe Abschnitt 3.1.1.1).

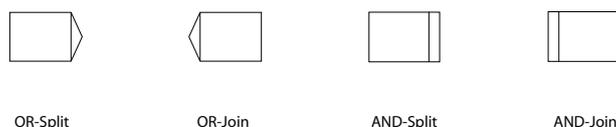


Schaubild 4.14: ADEPT – Kontrollelemente

4.3.1.4 Daten

Die Eingabe- und Ausgabe-Daten einer Aktivität und der Fluss der Daten zwischen Aktivitäten repräsentieren einen bedeutenden Aspekt innerhalb eines WfMS. In ADEPT

wird durch verschiedene Maßnahmen [23] die Konsistenz der Daten gewährleistet. Annotiert werden die Daten, wie in Schaubild 4.15 gezeigt, ähnlich zur BPMN durch Annotationskanten und einem Datensymbol.

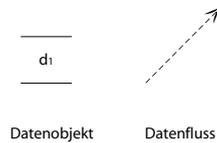


Schaubild 4.15: ADEPT – Daten

4.3.2 Grundlegende Modelle in ADEPT

Analog zu Abschnitt 4.2.2 während der Vorstellung von BPMN sollen an dieser Stelle grundlegende Konstrukte gezeigt werden, die in ADEPT modelliert wurden. Gleichzeitig stellen diese Konstrukte auch die Basis der blockstrukturierten Notation ADEPT dar. Die Modelle in ADEPT bestehen prinzipiell, wie die in Kapitel 3 vorgestellten blockstrukturierten Workflows, rein aus der Komposition der hier vorgestellten Grundkonstrukte. Natürlich sind die Split- und Join-Elemente nicht auf zwei aus- bzw. eingehende Kanten beschränkt.

In den Schaubildern 4.16, 4.17, 4.18 und 4.19 sind die 4 grundlegenden Konstrukte Sequenz, Auswahlstruktur, parallele Struktur und Schleife illustriert.

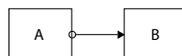


Schaubild 4.16: ADEPT – grundl. Modelle – Sequenz

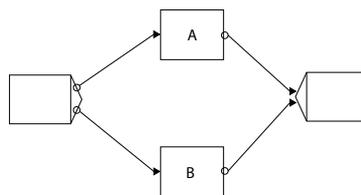


Schaubild 4.17: ADEPT – grundl. Modelle – Auswahlstruktur

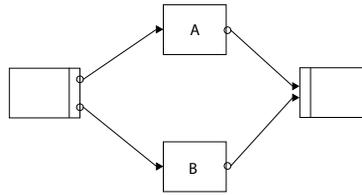


Schaubild 4.18: ADEPT – grundl. Modelle – parallele Struktur

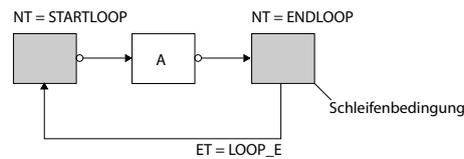


Schaubild 4.19: ADEPT – grundl. Modelle – Schleife

ADEPT erweitert die Basis der Grundkonstrukte im Gegensatz zu den blockstrukturierter Workflows aus Kapitel 3 um das Konstrukt „parallele Struktur mit finaler Auswahl“. In diesem Konstrukt werden zwei oder mehr Pfade parallel ausgeführt. Der OR-Join aktiviert nach dem erfolgreichen Beenden eines Pfades den nachfolgenden Pfad und behandelt die restlichen Aktivitäten auf den noch nicht beendeten Pfaden gemäß ihr Status, indem sie abgebrochen oder rückgängig gemacht werden [23].

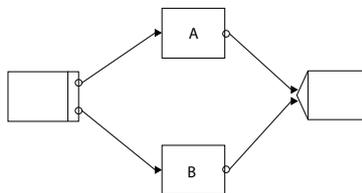


Schaubild 4.20: ADEPT – grundl. Modelle – parallele Struktur mit finaler Auswahl

Um einen Einblick in die Modellierung durch ADEPT zu geben, wurde der Beispielworkflow in Schaubild 4.21 ausgewählt. Die Einbeziehung einer Synchronisationskante ist ebenfalls dargestellt.

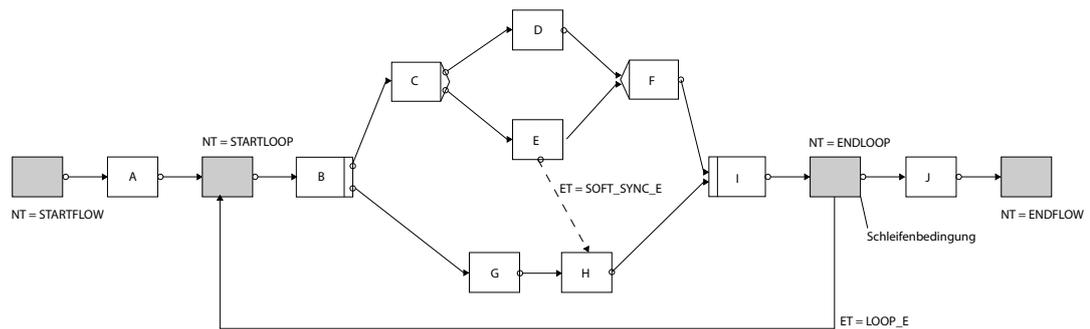


Schaubild 4.21: ADEPT – grundl. Modelle – Beispielworkflow

4.3.3 Einordnung

Die Modellierung in ADEPT erfolgt in blockstrukturierter Weise, wie es bereits in Kapitel 3 vorgestellt wurde. Dies bedeutet, es existieren in ADEPT keine überlappende Blöcke, sondern nur korrekt verschachtelte Modelle. Die Grundkonstrukte bilden die Modelle, die in Abschnitt 4.3.2 vorgestellt werden. Die Sequenz, das parallele Konstrukt und das bedingte Konstrukt sind äquivalent zu den Modelle aus Kaptel 3. Das Schleifenkonstrukt kann keine Aktivitäten auf der rückwärts gerichteten Kante besitzen. Dies stellt aber keine Einschränkung dar, da hierfür eine Transformation existiert. Die *parallele Struktur mit finaler Auswahl* ist als Erweiterung zu sehen, und produziert, da der Join-Auswahl als Diskriminator wirkt, keine multiplen Instanzen nachfolgender Aktivitäten. Die Intention dieses Konstruktes ist es, Aktivitäten in den Zweigen parallel auszuführen. Nachdem alle Aktivitäten eines einzelnen Zweiges vollständig ausgeführt wurden, aktiviert die Join-Auswahl die nachfolgenden Aktivitäten und ignoriert das erfolgreiche Beenden weiterer Zweige.

Neben dem Konstrukt *parallele Struktur mit finaler Auswahl* wurde die zweite Erweiterung des blockstrukturierten Ansatzes durch ADEPT, namentlich die Synchronisationkanten bereits in Abschnitt 4.3.1.2 vorgestellt. ADEPT besitzt also, im Vergleich zu einer klassischen blockstrukturierten Notation zwei Erweiterungen, die aber erwiesenermaßen die Sicherheit vor Deadlocks und multiplen Instanzen einer Aktivität nicht beeinträchtigen [23].

Kapitel 5

Abbildbarkeit der BPMN auf ADEPT

Der gute Redner wird Vergleiche anwenden und Beispiele vorbringen.

Marcus Tullius Cicero (106-43 v. Chr.)
röm. Redner u. Schriftsteller

Nach der Schaffung von, für die Untersuchung der Abbildbarkeit bedeutenden Grundlagen durch die vorigen Kapitel, soll im Folgenden die Abbildbarkeit unstrukturierte Prozessmodelle auf blockstrukturierte Workflows anhand der Notationen BPMN und ADEPT untersucht werden. Hierzu werden die Ergebnisse bereits publizierter Arbeiten aufgegriffen und angepasst. Anpassungen sind unter anderem durch die vorgestellten Erweiterungen der blockstrukturierten Notation ADEPT durch zum Beispiel Synchronisationskanten nötig.

Um die Abbildbarkeit zu evaluieren, wird zuvor in Abschnitt 5.1 das Prinzip des Nachweises der Äquivalenz zweier Modelle angeführt. Anschließend wird die Abbildbarkeit von BPMN und ADEPT anhand einer Fallunterscheidung erörtert. Abschließend soll die Automatisierbarkeit der Abbildungen betrachtet und eine Möglichkeit zur kombinierten Analyse und Transformation vorgestellt werden.

5.1 Äquivalenz

Um Abbildungen auf ihre Richtigkeit zu verifizieren, ist es von Nöten eine Definition zu geben, die festlegt, wann zwei Workflows unter den Gesichtspunkten des Kontrollflusses äquivalent sind. Ähnlich zu Kapitel 3 ist hier unter einem Workflow die zeitlich koordinierte Abfolge von Aktivitäten zu verstehen und ermöglicht somit die Übertragung der Ergebnisse auf die Äquivalenz von Prozessmodellen mit Modellen von Workflows.

Eine Möglichkeit die Äquivalenz von zwei Workflows zu zeigen ist das Prinzip der „bisimulation games“, das u.a. in [16] vorgestellt wird. Das Konzept der „bisimulation games“ beruht prinzipiell auf der Nachahmung des Kontrollflusses eines Workflows durch den anderen Workflow. Dazu eine Definition:

Definition 5.1 *Workflow A und Workflow B sind **äquivalent**, wenn Workflow A jeden Schritt*

(z.B. das Starten des Workflows, eine Aktivität starten/beenden, etc.) von Workflow B imitieren kann und Workflow B jeden Schritt von Workflow A imitieren kann.

Um die Schritte besser vergleichbar zu machen, können Modelle zweier Notationen beispielsweise auf einen Formalismus wie Petri-Netze abgebildet werden [16]. Aufgrund der Vorstellung des Prinzips dieses Vorgehens sollen aber diese Details außen vor bleiben.

Es soll weiterhin festgehalten werden, dass es neben äquivalenten Workflows noch sogenannte quasi-äquivalente Workflows gibt.

Definition 5.2 Ein Abbildung von Workflow A auf Workflow B ist **quasi-äquivalent**, wenn Workflow A Workflow B simulieren kann, aber Workflow B nicht Workflow A simulieren kann.

Um das Verständnis für einen quasi-äquivalenten Workflow zu fördern, soll in Schaubild 5.1 ein in der BPMN notiertes Beispiel gezeigt werden. Hier stellt Workflow A gegenüber Workflow B eine quasi-äquivalente Abbildung dar.

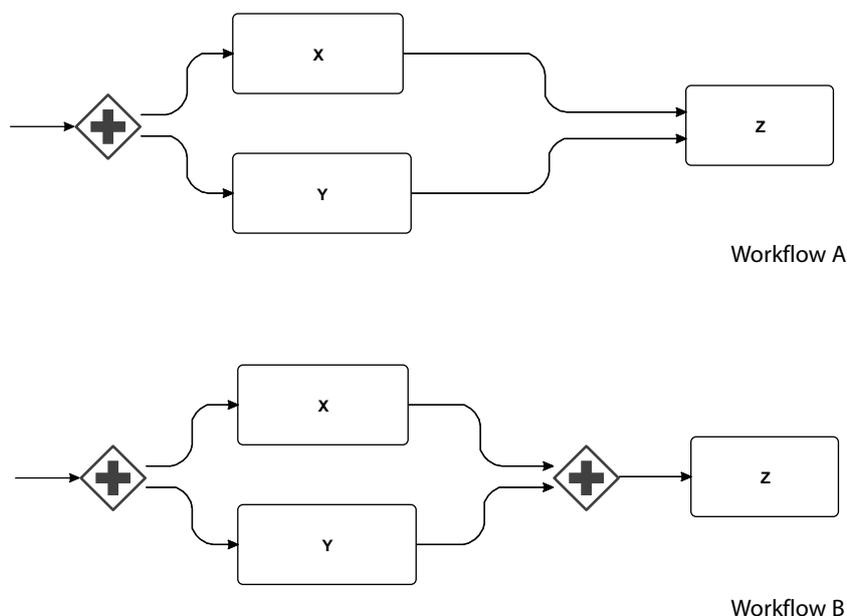


Schaubild 5.1: Äquivalenz – Quasi-Äquivalent Beispiel

Aus dem Schaubild geht hervor, dass Workflow A ausdrucksstärker als Workflow B ist. Mögliche Ausführungspfade¹ der Aktivitäten für Workflow A sind XYZ, YXZ, XZY und YZX. Workflow B besitzt nur die Ausführungspfade XYZ und YXZ. Deshalb kann Workflow A die Ausführungsschritte von Workflow B simulieren. Der inverse Fall trifft nicht zu.

¹Ausführungspfad = Reihenfolge für die Beendigung unter der Annahme einer Serialisierung

5.2 Grundlagen der Abbildbarkeit

Die Untersuchung bezüglich der Abbildbarkeit legt ihren Schwerpunkt auf den Kontrollfluss. Es wurde bereits erwähnt, dass auf die Abbildung von Details wie zum Beispiel von Rollen in Form von Swimlanes und Pools, speziellen Ereignissen oder die Behandlung von Fehlern durch Kompensationsprozesse aus zeitlichen Gründen verzichtet wird.

5.2.1 Abbildungen der Elemente

Um die Abbildbarkeit zu gewährleisten, müssen als Basis die Abbildungen der einzelnen Elemente aus BPMN auf entsprechende Elemente in ADEPT aufgeführt werden. Dies erfolgt in den folgenden Abschnitten.

5.2.1.1 Aktivitäten

Die Aktivitäten in BPMN haben ihr äquivalentes Pendant in den Knoten in ADEPT. Schaubild 5.2 illustriert diesen Zusammenhang.



Schaubild 5.2: Abbildungen der Elemente – Aktivitäten

5.2.1.2 Ereignisse

Die Abbildung des Start- und End-Ereignisses aus Modellen in der BPMN erfolgt intuitiv auf den Startknoten und Endknoten in ADEPT. Zwischenereignisse sollen, wie bereits bei ihrer Vorstellung erwähnt, auf eine Aktivität in ADEPT abgebildet werden. So soll je nach Bedeutung des Ereignisses, das Senden und Empfangen eines Signals durch eine für diesen Zweck erschaffene Aktivität behandelt werden. Eine detaillierte Behandlung der vielfältigen Ereignisse aus der BPMN ist aus zeitlichen Gründen nicht möglich.

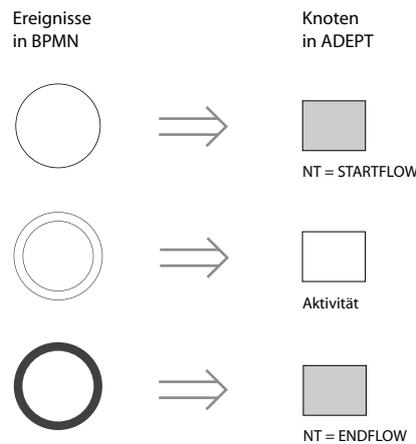


Schaubild 5.3: Abbildungen der Elemente – Ereignisse

5.2.1.3 Gateways

Die in der BPMN vorhandenen exklusiven Gateways (XOR) und die parallelen Gateways besitzen je nach ihrer Funktion als Split- oder Join-Element ein Äquivalent in ADEPT. Dies ist im folgenden Schaubild verdeutlicht dargestellt.

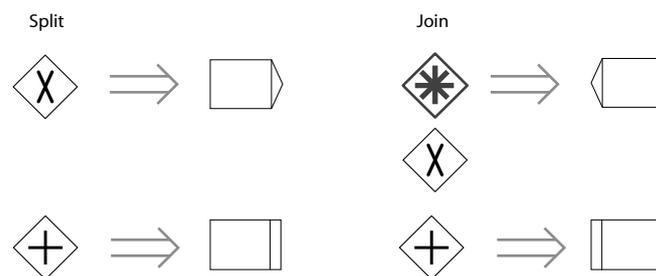


Schaubild 5.4: Abbildungen der Elemente – Gateways

Das komplexe Gateway soll, wenn es als Join-Element mit der Semantik des Diskriminators fungiert, auf das entsprechende Element in ADEPT abgebildet werden. Zu inklusiven Gateways ist anzuführen, dass das inklusive Gateway, wenn es als Split-Element dient, durch ein paralleles Gateway und exklusive Gateways simuliert werden kann. Ein Join durch ein inklusives Gateway ist ebenfalls rekonstruierbar.

Das Ereignis-basierte Gateway, das an sich nur als Split-Element eingesetzt wird, kann durch ein Hilfskonstrukt (Schaubild 5.5) simuliert werden. Aktivität A wartet dabei, bis ein entsprechendes Ereignis eingetreten ist und speichert dies bei Eintreten dementsprechend in ein Datenobjekt. Aktivität B wählt anhand des Datenobjekts den gewünschten Pfad aus.

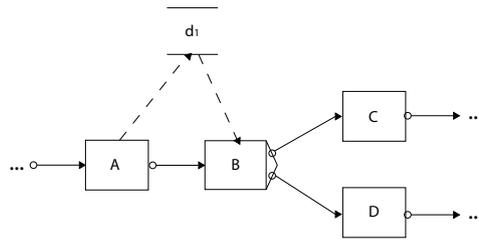


Schaubild 5.5: Abbildungen der Elemente – Hilfskonstrukt Ereignis-basiertes Gateway

5.2.1.4 Daten

Falls eine Aktivität A in BPMN auf ein Datenobjekt zugreift und eine Aktivität B das Objekt liest, sollte in ADEPT der selbe Zugriff auch nach einer Abbildung gewährleistet sein. Da ADEPT komplexe Regeln für den korrekten Zugriff auf Datenobjekte voraussetzt, ist die korrekte Abbildung der Datenobjekte kein Trivialität und soll in dieser Arbeit ausgeblendet werden.

5.2.2 Abbildungen von blockstrukturierten Grundmustern

Es sollen in den folgenden Schaubildern 5.6, 5.7, 5.8, 5.9 und 5.10 die Abbildungen von in der BPMN modellierten blockstrukturierten Konstrukten auf die äquivalente Konstrukte in ADEPT illustriert werden. Diese Abbildungen stellen den Fall dar, dass innerhalb eines Modells in der BPMN die blockstrukturierten Basiskonstrukte identifiziert werden können. Diese Konstrukte innerhalb eines Modells in BPMN könnten durch einzelne pseudo Aktivitäten ersetzt werden, und das Modell weiter analysiert werden. Das Modell würde sich, falls es blockstrukturiert wäre, neben dem Start- und dem Endknoten auf eine einzelne (pseudo) Aktivität reduzieren.

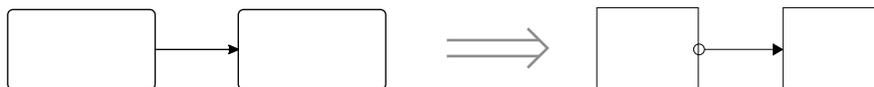


Schaubild 5.6: Abbildungen Grundmuster – Sequenz

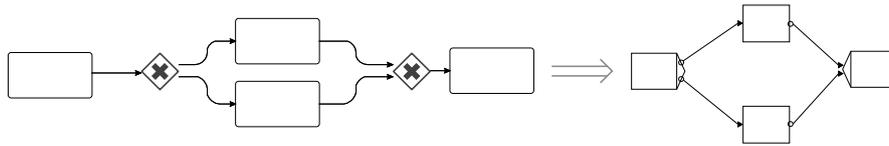


Schaubild 5.7: Abbildungen Grundmuster – Auswahlstruktur

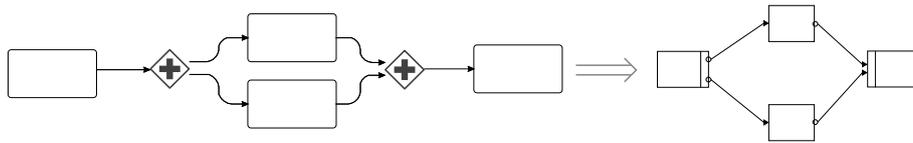


Schaubild 5.8: Abbildungen Grundmuster – Parallele Struktur

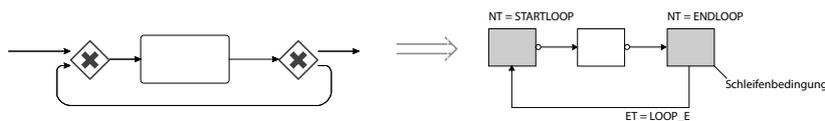


Schaubild 5.9: Abbildungen Grundmuster – Schleife

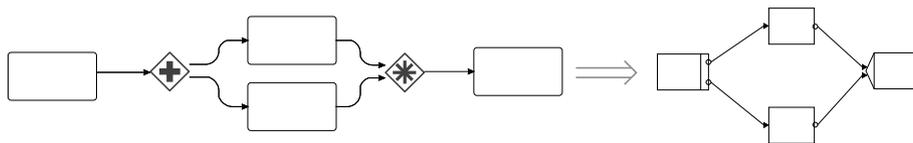


Schaubild 5.10: Abbildungen Grundmuster – Parallele Struktur mit finaler Auswahl

5.3 Abbildbarkeit von Modellen mit strukturellen Mängeln

Durch die Reduzierung eines Prozessmodells in BPMN um die abbildbaren Grundmuster, werden, falls das Modell nicht blockstrukturiert ist, die Konstrukte, die eine Abbildung des Modells verhindern könnten, sichtbar. Die Reduktion eines Prozessmodells kann also zu Identifizierung der „Problemstellen“ einer Abbildung innerhalb eines Prozessmodells dienen. Dieses Prinzip der kombinierten Analyse und Identifikation von

unstrukturierten Konstrukten wird in [34] im Detail erläutert. Auf die Behandlung von strukturellen Mängeln innerhalb eines Modells soll im folgenden Abschnitt eingegangen werden.

5.3.1 Formen struktureller Mängel

Um eine Untersuchung von Konstrukten mit strukturellen Mängeln in angemessener Form zu ermöglichen, werden an dieser Stelle zwei Annahmen getroffen.

Es soll zum Einen eine möglichst weitgehende Transformation der inklusiven, komplexen und Ereignis-basierten Gateways in Konstrukte aus exklusiven und parallelen Gateways stattfinden. Es wurde bewusst der Term „möglichst weitgehend“ gewählt, denn ein Transformation der exklusiven Gateways und inklusiven Gateways, die als Join-Elemente dienen, ist nicht unter allen Umständen möglich. Das Einbeziehen dieser beiden Gateways soll in späteren Ergebnissen dieser Arbeit ebenfalls noch möglich sein. Ein kurzes Beispiel soll auf diesen Sachverhalt eingehen. Schaubild 5.11 zeigt ein Prozessmodell, das ein inklusives Gateways als Split-Element über zwei Aktivitäten mit einem komplexen Gateway (als Diskriminator) verbindet. Eine Transformation des inklusiven Gateways in zwei exklusive und ein paralleles Gateway ist möglich. Um jedoch die Semantik des Diskriminators aufrechtzuerhalten, wird das komplexe Gateway weiterhin benötigt (Schaubild 5.12).

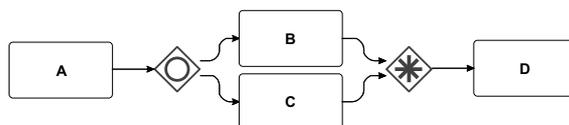


Schaubild 5.11: Beispiel Diskriminator in BPMN

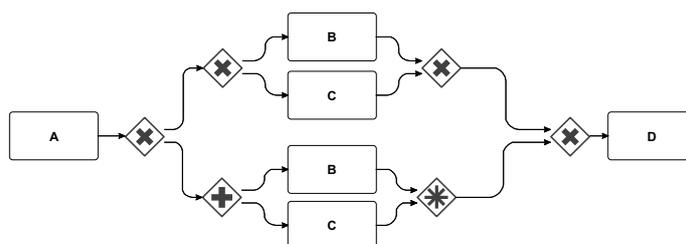


Schaubild 5.12: Beispiel Diskriminator in BPMN – Transformation

Für die folgenden Untersuchungen soll außerdem vorausgesetzt werden, dass jedes Split-Gateway mit mehr als zwei ausgehenden Pfaden in zwei oder mehrere Gateways mit jeweils zwei ausgehenden Pfaden aufteilt werden soll. Jedes Join-Gateway, das mehr als zwei eingehende Pfade besitzt, soll analog in zwei oder mehrere Gateways mit jeweils zwei eingehenden Pfaden aufteilt werden.

Aufgrund dieser Voraussetzungen kann folgende Definition nach [15] eingeführt werden und für die folgenden Abschnitte verwendet werden.

Definition 5.3 Ein Split-Gateway s *korrespondiert* zu einem JOIN-Gateway j , wenn zwei minimale Pfade, die aus den zwei verschiedenen Ästen von s ausgehen, als erstes in j münden. Das korrespondierende Paar wird mit (s,j) bezeichnet.

Mit dem Wort „minimal“ soll zum Ausdruck gebracht werden, dass kein anderes Gateway auf einem Teil des Pfades die korrespondierende Eigenschaft mit s besitzt. Zusätzlich wird in [15] erläutert, dass jedes Split-Element mindestens ein korrespondierendes Join-Element besitzen muss. Dadurch können in der Folge Paare von Gateways identifiziert werden, die zu multiplen Instanzen oder Deadlocks führen können (Schaubild 5.13).

Definition 5.4 Ein Paar korrespondierender Gateways (s,j) ist *falsch zugeordnet*, wenn s ein exklusives und j ein paralleles Gateway (Deadlock) oder s ein paralleles Gateway und s ein exklusives Gateway (multiple Instanzen) ist.

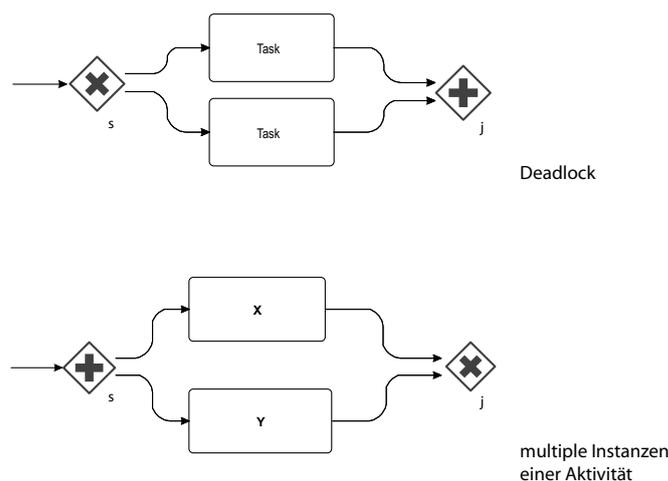


Schaubild 5.13: Falsch zugeordnete Paare in BPMN

Es ergeben sich also, wie bereits in Kapitel 3 vorgestellt, die Probleme der Deadlocks und multiplen Instanzen einer Aktivität durch missbräuchliche Gegenüberstellung von bestimmten Gateways. Um die Ergebnisse aus Kapitel 3 vollständig aufzugreifen, ergibt sich in BPMN neben falsch zugeordneten Paaren auch die unzulässige Verschachtelung als das zweite Problem bei einer Abbildung von Prozessmodellen in der BPMN auf Workflows in ADEPT. Die blockstrukturierte Notation ADEPT erlaubt es, wie der Name es schon sagt, keine „Überlappungen“ der von den Gateways gebildeten Blöcke. Schaubild 5.14 zeigt den Fall einer solchen Überlappung in der BPMN.

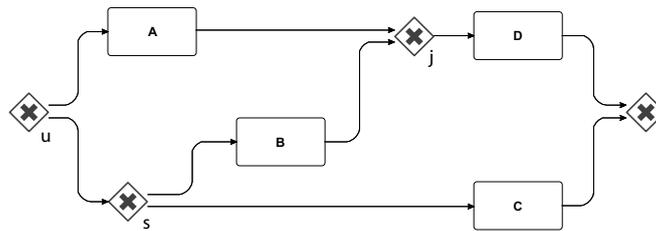


Schaubild 5.14: Beispiel einer unzulässigen Verschachtelung

Es folgt für diese Art der Überlappungen eine Definition nach [15].

Definition 5.5 Ein Paar korrespondierender Gateways (s,j) ist **unzulässig verschachtelt** mit einem anderen Paar korrespondierender Gateways (u,v) , wenn s (oder j) in einem (bestimmten) Pfad von u nach v ist, aber j (oder s) nicht in diesem Pfad ist.

Der Pfad von Gateway u , über das Gateway s , weiter zur Aktivität C und schließlich nach Gateway v zeigt, dass das Modell aus Schaubild 5.14 unzulässig verschachtelte Paare an Gateways besitzt.

5.3.2 Azyklisch unstrukturierte Modelle

Falls ein Prozessmodell eine oder mehrere unzulässige Verschachtelungen vorweist, müssen diese analysiert werden, ob es möglich ist, die Verschachtelungen in äquivalente blockstrukturierte Konstrukte zu transformieren.

Für ein unzulässig verschachteltes Konstrukt, das nur aus exklusiven Gateways besteht und somit keine Parallelität vorweisen, ist eine vollständige Transformation in ein blockstrukturiertes Modell möglich. Das Vorgehen wird in [15] wie folgt beschrieben (angepasst):

Die Aktivitäten, die zwischen zwei benachbarten exklusiven Join-Gateways liegen, werden dupliziert und über das vordere Gateway geschoben und anschließend werden die zwei exklusiven Join-Gateways vertauscht.

Anmerkung: Mit dem „vorderen Gateway“ ist das Gateway gemeint, das sich näher am Anfang des Konstruktes befindet.

Angewandt auf das Prozessmodell aus Schaubild 5.14 ergibt sich ein äquivalentes blockstrukturiertes Prozessmodell illustriert in Schaubild 5.15. Die Abbildung dieses dann blockstrukturierten Prozessmodells nach ADEPT erfolgt durch die in Abschnitt 5.2.2 bereits vorgestellten Abbildungen der blockstrukturierten Grundmuster.

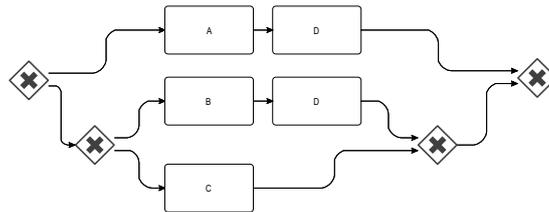


Schaubild 5.15: Unzulässige Verschachtelung aus exklusiven Gateways

Es wurde bereits in Abschnitt 4.3.1.2 während der Vorstellung der Synchronisationskanten gezeigt, dass das unzulässig verschachtelte Modell in Schaubild 5.16 eine äquivalente Form in ADEPT besitzt, jedoch kein rein blockstrukturiertes Pendant.

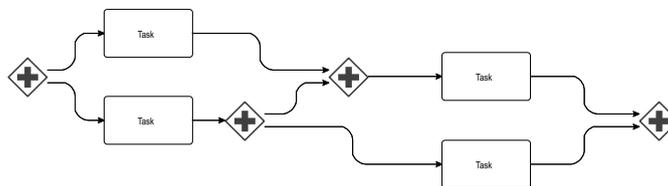


Schaubild 5.16: Unzulässige Verschachtelung aus parallelen Gateways

Es wäre mühsam und nicht sachgerecht, jede einzelnen Möglichkeit der unzulässigen Verschachtelung gesondert mit einem Beispiel anzuführen. Deshalb soll anhand der Prozessmodelle pm1 und pm2 aus den Schaubildern 5.17 und 5.18 und der Tabelle 5.1 eine Übersicht über die Möglichkeiten der Abbildungen dieser Konstrukte gegeben werden. Die Tabelle lehnt sich an Ergebnisse aus [15] an und wurde entsprechend für die Abbildung der BPMN auf ADEPT modifiziert.

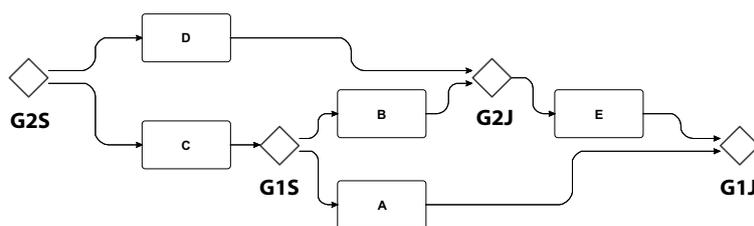


Schaubild 5.17: Muster unzulässiger Verschachtelungen - pm1

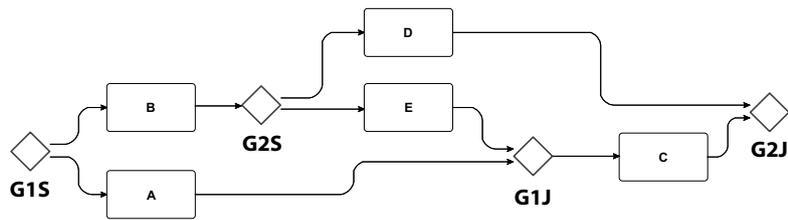


Schaubild 5.18: Muster unzulässiger Verschachtelungen - pm2

Die Split-Gateways G1S und G2S können entweder ein exklusives Gateway oder ein paralleles Gateway sein. Die Join-Gateways können entweder ein paralleles Gateway, ein exklusives Gateway, ein inklusives Gateway oder auch ein komplexes Gateway (als Diskriminator) sein.

Es wurde deshalb bei den Eigenschaften und der Aussage über die Abbildbarkeit der Modelle der Typen 3, 7, 13, 15 und 16 eine Markierung in Form eines Sterns (*) angefügt. Dies soll bedeuten, dass wenn das „OR“ oder die „OR“s in einer Tabellenzeile durch inklusive oder komplexe Gateways ersetzt werden, ist das unstrukturierte Konstrukt well-behaved und es existiert eine äquivalente Abbildung nach ADEPT, die meistens zusätzlich durch Synchronisationskanten realisiert werden.

Falls aber in einem Konstrukt eines der Join-Gateways in Form eines exklusiven Gateways realisiert wird, wird das Konstrukt zu multiplen Instanzen einer Aktivität führen und es existiert nur eine quasi-äquivalente Abbildung.

Typ	G1S	G1J	G2S	G2J	Eigenschaften	Abb. ADEPT
1	OR	OR	OR	OR	Well-behaved	Ja
2	OR	OR	OR	AND	Deadlock	Nein
3	OR	OR	AND	OR	* mult. Inst.	* qu.-äquiv.
4	OR	OR	AND	AND	Deadlock	Nein
5	AND	AND	OR	OR	Deadlock	Nein
6	AND	AND	OR	AND	Deadlock	Nein
7	AND	AND	AND	OR	* mult. Inst.	* qu.-äquiv.
8	AND	AND	AND	AND	Well-behaved	Ja
9	OR	AND	OR	OR	Deadlock	Nein
10	OR	AND	OR	AND	Deadlock	Nein
11	OR	AND	AND	OR	Deadlock	Nein
12	OR	AND	AND	AND	Deadlock	Nein
13	AND	OR	OR	OR	* mult. Inst.	* qu.-äquiv.
14	AND	OR	OR	AND	* mult. Inst.	* qu.-äquiv.
15	AND	OR	AND	OR	Deadlock	Nein
16	AND	OR	AND	AND	* mult. Inst.	* qu.-äquiv.

Tabelle 5.1: Abbildbarkeit unzulässig verschachtelter Konstrukte

Die Modelle pm1 und pm2 sind generischer Natur, da die Aktivitäten auch als reduzierte Teilprozesse angesehen werden können. Trotzdem ergeben sich neben der unzulässigen Überlappung von zwei korrespondierenden Gateways auch Situationen, in denen mehrfache Überlappungen auftreten können. Einige können nicht durch die Abbildungen aus Tabelle 5.1 aufgelöst werden.

Ein Problem ist das Auftreten des Sachverhalts innerhalb eines Modelles, der in Schaubild 5.19 dargestellt ist.

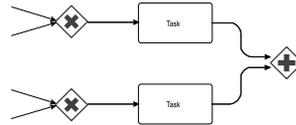


Schaubild 5.19: Unzulässige Verschachtelung – Erweiterung

Falls sich dies als das „Ende“ einer sogenannten überlappenden Struktur (engl. overlapped structure, Schaubild 5.20) herausstellt, existiert eine äquivalente blockstrukturierte Abbildung, die in Schaubild 5.21 durch BPMN illustriert wird. Diese nach ADEPT abzubilden, sollte durch ihre blockstrukturierte Form keine Umstände bereiten.

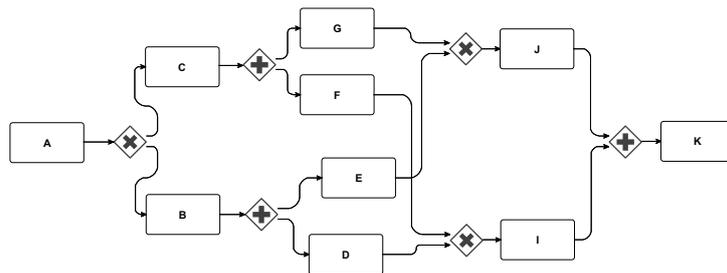


Schaubild 5.20: Unzulässige Verschachtelung – Überlappende Struktur

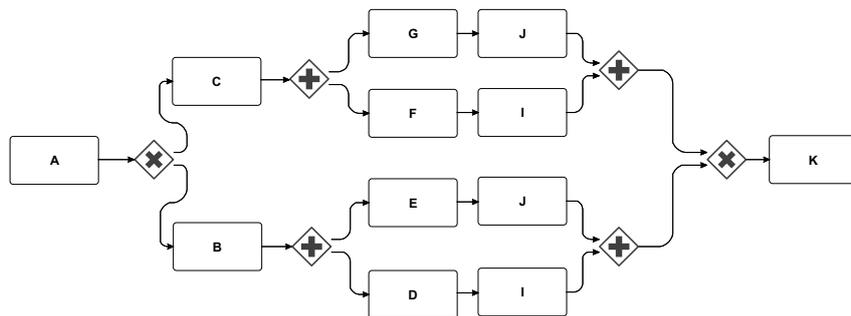


Schaubild 5.21: Unzulässige Verschachtelung - Blockstr. Abbildung überlappende Struktur

5.3.3 Zyklische unstrukturierte Modelle

Bisher wurden nur azyklische Modelle in der Untersuchung berücksichtigt. Jedoch wurde in Kapitel 3 bereits angemerkt, dass neben falsch zugeordneten Paaren und der unzulässigen Verschachtelung, auch Schleifen eine Quelle von Deadlocks und multiplen Instanzen sein können. Obwohl an dieser Stelle auf eine Nachmodellierung der Schleifen-Konstrukte aus Abschnitt 3.2.1, die zu Deadlocks oder multiplen Instanzen führen, verzichtet wird, sei festgehalten, dass diese Konstrukte, in der BPMN modelliert, natürlich nicht nach ADEPT abbildbar sind.

Weiterhin wurde bereits in Abschnitt 5.2.2 gezeigt, dass die Abbildung eines blockstrukturierten Schleifenkonstruktes aus der BPMN nach ADEPT möglich ist.

Neben der „falschen Konstruktion“ einer Schleife existieren aber leider zyklische Konstrukte, die in ihrer Form ähnlich zu den unzulässigen Verschachtelungen in azyklischen Modellen sind. So bedarf es für den Eintritt sowie des Austritts aus einer Schleifenkonstruktion einer weitergehenden Untersuchung. Die Schaubilder 5.22 und 5.23 illustrieren die Muster des unzulässigen Eintritts in eine Schleife sowie des unzulässigen Austritts aus einer Schleife.

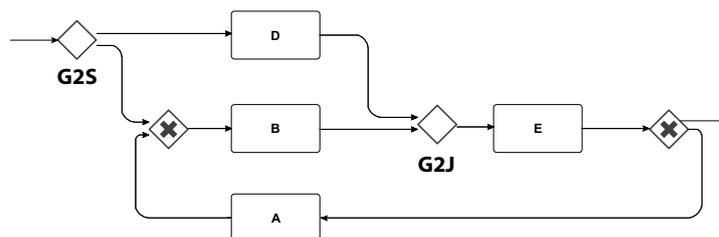


Schaubild 5.22: Unzulässiger Eintritt in eine Schleife

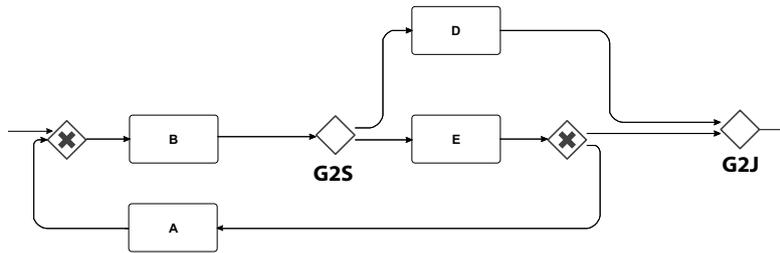


Schaubild 5.23: Unzulässiger Austritt aus einer Schleife

Es gibt ebenfalls Aufstellungen über die Abbildbarkeit dieser Konstrukte aus [15] und sie wurden wie zuvor für die Abbildung der BPMN auf ADEPT modifiziert. Die Tabellen 5.22 und 5.23 geben damit Aufschluss über die Abbildbarkeit der Konstrukte des unzulässigen Eintritts in eine Schleife und des unzulässigen Austritts aus einer Schleife. Analog zu Abschnitt 5.3.2 wurde eine Kennzeichnung (*) verwendet, falls das Modell well-behaved ist, wenn das Join-Gateway G2J durch ein inklusives Gateway oder durch ein komplexes Gateway mit der Wirkung eines Diskriminators anstatt eines exklusiven Gateways ersetzt wird.

Die Möglichkeiten zur Ersetzung der Gateways G2S und G2J sind ebenfalls analog nach Abschnitt 5.3.2 gegeben.

Typ	G2S	G2J	Eigenschaften	Abb. ADEPT
1N	AND	OR	* multiple Instanzen	* quasi-äquivalent
2N	OR	AND	Deadlock	Nein
3N	OR	OR	Well-behaved	Ja
4N	AND	AND	Deadlock	Nein

Tabelle 5.2: Abbildbarkeit unzulässiger Eintritt in eine Schleife

Typ	G2S	G2J	Eigenschaften	Abb. ADEPT
1A	AND	OR	* multiple Instanzen	Nein
2A	OR	AND	Deadlock	Nein
3A	OR	OR	Well-behaved	Ja
4A	AND	AND	Well-behaved	Nein

Tabelle 5.3: Abbildbarkeit unzulässiger Austritt aus einer Schleife

Es soll auf drei Ergebnisse gesondert eingegangen werden, da diese Ergebnisse für eine Untersuchung der Abbildung von Relevanz sind. Zum Einen soll auf die Form der blockstrukturierten Abbildung des Konstrukts 3A eingegangen werden und zum Anderen auf den Umstand, weshalb die Konstrukte 1A und 4A keine äquivalente blockstrukturierten Pendanten besitzen.

Ein Konstrukt von Typ 3A besitzt eine äquivalente Abbildung nach ADEPT, die aller-

dings nur durch das Einbeziehen von Hilfsvariablen konstruierbar ist. Hilfsvariablen sind in diesem Kontext Datenobjekte, die dazu dienen, den gewählten Kontrollfluss festzuhalten und diese Information zu einem späteren Zeitpunkt wieder verfügbar zu machen. Beispielsweise kann in einem Datenobjekt der gewählte Pfad an einem exklusives Split-Gateway gespeichert werden, um später anhand dieser Information den Pfad bei einer weiteren Verzweigung auswählen zu könne. Für einen Beweis, dass das Konstrukt 3A nur durch das Einbeziehen von Hilfsvariablen in einer blockstrukturierten Form konstruierbar ist, sei auf [15] verwiesen. Aufgrund der Größe des Schaubilds sei an dieser Stelle auf eine Illustration verzichtet. Es sei jedoch festgehalten, dass unter der Einführung von Hilfsvariablen die Nachvollziehbarkeit des Kontrollflusses eines Prozessmodells bzw. eines Workflows erheblich leidet.

Es soll weiterhin der Beweis erbracht werden, dass ein unzulässige Austritt aus einer Schleife vom Typ 1A nicht in eine blockstrukturierte Form gebracht werden kann (nach [15]). Der Beweis erfolgt durch Widerspruch.

Beweis 2 *Unter der Annahme, dass eine blockstrukturierte Abbildung des Konstruktes vom Typ 1A existiert, müsste diese Abbildung aus einer parallelen Struktur und einem Schleifen-Konstrukt bestehen. Die parallele Struktur würde Aktivität E und D enthalten. Wenn die parallele Struktur innerhalb der Schleife wäre, stellt dies ein Widerspruch dar, da D sich außerhalb der Schleife befindet. Wäre die parallele Struktur außerhalb der Schleife, wäre dies ebenfalls ein Widerspruch, denn E liegt innerhalb der Schleife. Es folgt, dass die Annahme einen Widerspruch darstellt.*

Ein Nachweis, dass für ein Konstrukt vom Typ 4A ebenfalls keine Abbildung existiert, würde analog erfolgen. Damit ist festzuhalten, dass das Konstrukt vom Typ 4A, obwohl es well-behaved ist, keine Abbildung nach ADEPT besitzt.

5.4 Automatisierbarkeit

Durch die vorigen Abschnitte wurde bereits aufgezeigt, dass es für Konstrukte innerhalb eines Prozessmodells äquivalente, quasi-äquivalente und auch keine Abbildungen gibt.

Die Abbildungen von Konstrukten, die ein blockstrukturiertes Pendant besitzen, können automatisiert, ohne das Mitwirken einer Fachkraft, erfolgen. Bei Konstrukten, für die nur eine quasi-äquivalente Abbildung existiert, könnte sich der Umstand ergeben, dass eine Abbildung in dieser Form vom Modellierer nicht gewünscht wird. Es sollte bei einer Abbildung eine Abfrage, ob eine Abbildung in ein quasi-äquivalentes Modell durchgeführt werden soll, erfolgen. Eine Software könnte die Abbildung begleiten bzw. auch durchführen, und diese Form der Rücksprache realisieren. Auch die Behandlung von erkannten Deadlocksituation bei unzulässig verschachtelten Gateway-Paaren könnte in dieser Form realisiert werden, wenngleich die Erkennung eines Vorschlags für die Abbildung des Deadlock sich weitaus schwieriger gestalten kann.

Ein Beispiel für solch einen Ansatz könnte die Behandlung des Deadlocks des unzuläs-

sig verschachtelten Konstrukts vom Typ 4 sein (siehe Tabelle 5.1). Ein Vorschlag für die Behandlung des Konstrukts ist in Schaubild 5.24 zu sehen.

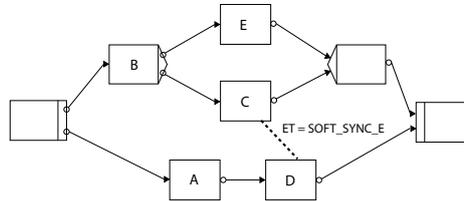


Schaubild 5.24: Vorschlag einer Abbildung der unzulässigen Verschachtelung Typ 4

Natürlich existieren Konstrukte, die Deadlocks verursachen, für die sich ein, bezüglich der Ausführungsemantik möglichst naheliegender Vorschlag, nicht finden lässt. Ein Deadlock vom Typ 2 aus den unzulässigen Verschachtelungen ist hierfür eventuell ein Beispiel.

5.4.1 Kombinierte Analyse und Transformation

Die Forscher Hauser et al. stellen in ihrer Arbeit *Combining Analysis of Unstructured Workflows with Transformation to Structured Workflows* [17] einen Algorithmus vor, mit dem es möglich sein soll, die Analyse und eine Abbildung zu automatisieren.

Ein vollständige Vorstellung der Details der Arbeit würde dem zeitlichen Rahmen der Arbeit nicht gerecht. Es soll aber an dieser Stelle auf die Prinzipien dieser Arbeit eingegangen werden.

5.4.1.1 Algorithmus

Das Ziel des Algorithmus ist es, einen sogenannten Regionenbaum eines Modells aufzubauen. Dieser Baum wird durch den Algorithmus über Regeln aufgebaut. Prinzipiell wird jedem Element eines Modells eine einzelne, eindeutige Region zugewiesen. In der Arbeit wurden Regeln identifiziert, die es erlauben, zwei Regionen zu einer neuen zusammenzufassen, wenn diese den Regeln entsprechen. Die Regeln sind so gewählt, dass wenn die zwei Regionen konform zu einer Blockstruktur sind, es die neue Region ebenfalls ist. Im Modell ersetzt die neue Region die zwei vorigen. Durch das wiederholte Ausführen dieses Vorgangs lässt sich ein Binärbaum von den Blättern zur Wurzel aufbauen. Gleichzeitig reduziert der Algorithmus, falls das Modell blockstrukturiert ist, das Modell auf eine einzelne Region. Ein Beispiel für einen Regionenbaum und das dazugehörige Modell wird in den Schaubildern 5.25 und 5.26 dargestellt.

Kapitel 6

Schlussbetrachtung

Wer recht erkennen will, muß zuvor in richtiger Weise gezweifelt haben.

Aristoteles (384-322 v. Chr.)
griech. Philosoph

Nach der Untersuchung der Abbildbarkeit der Notation BPMN auf ADEPT soll abschließend eine Zusammenfassung der Ergebnisse gegeben werden. Dazu bietet es sich an, die in der Einleitung aufgeführten Fragen zu beantworten, um damit die Resultate dieser Arbeit zu erörtern.

Ist jedes Prozessmodell modelliert in der BPMN auf ADEPT abbildbar?

Die Antwort dieser Frage ist leider „Nein“. Es wurde in Kapitel 5 gezeigt, dass es eine Vielzahl an Mustern innerhalb eines Prozessmodells in BPMN gibt, die kein Äquivalent in ADEPT besitzen. Die Gründe hierfür liegen in der Konzeption der Notation BPMN und ADEPT. BPMN wurde klar mit dem Grundsatz der leichten Verständlichkeit für die Modellierer geschaffen. Die damit implizierten Folgen in Form des Mangels an strukturellen Vorschriften sind das Kernproblem einer Abbildung auf ADEPT. Des Weiteren steigert die hohe Vielfalt an Elementen in BPMN die Komplexität eines Algorithmus, der eine Abbildung inklusive einer Behandlung der strukturellen Mängel durchführen könnte.

Existiert eine Teilmenge an Elementen in BPMN, die eine vollständige Abbildung nach ADEPT ermöglichen?

Es wurde bereits erwähnt, dass es für jedes Modell ohne Parallelität (ohne parallele Gateways) eine blockstrukturierte Abbildung existiert. Natürlich ist das Ausschließen der parallelen Gateways nicht realitätsnah. Mit dem Einbinden der parallelen Gateways treten aber die Probleme auf, die im vorigen Kapitel bereits erläutert wurden. Zu diesen zählen eben Laufzeitfehler wie Deadlocks und multiple Instanzen einer Aktivität, die in einem Workflow nicht auftreten dürfen.

Kann jedes Prozessmodell, das erfolgreich beendet werden kann (well-behaved), in einen in ADEPT modellierten Workflow umgewandelt werden?

Nein. In Abschnitt 5.3.3 wurde gezeigt, dass ein Austritt aus einer Schleife vom Typ 4A zwar well-behaved ist, aber keine äquivalente blockstrukturierte Form besitzt. Es gibt somit Konstrukte, die einerseits in ADEPT nicht darstellbar sind und andererseits keine Laufzeitfehler, wie Deadlocks oder multiple Instanzen einer Aktivität, verursachen.

Gibt es bereits existente Abbildungen der BPMN auf andere blockstrukturierte Notationen als ADEPT?

Es existieren diverse Untersuchungen [25, 35, 26, 27] der BPMN auf die blockstrukturierte BPEL. In der BPEL existieren wie in ADEPT blockstrukturierte Grundkonstrukte und die Vorschrift, Schleifen nur durch ein entsprechendes Grundkonstrukt modellieren zu können. Jedoch bietet BPEL Erweiterungen wie die Elemente „flow“ und „link“, die unstrukturierte parallele Konstrukte ermöglichen, sowie den sogenannten „event handler“ [35]. Viele dieser Arbeiten orientieren sich sehr stark an der Überwindung der Abbildungsbarrieren anhand dieser Sonderkonstrukte. Das Ziel dieser Arbeit ist es, die Abbildbarkeitsbarrieren zwischen BPMN und ADEPT zu erläutern.

In welcher Form wirken sich Erweiterungen der blockstrukturierten Notation ADEPT auf die Abbildbarkeit aus?

Ein interessantes Resultat ist die Erweiterung der Abbildungsmöglichkeiten einer unstrukturierten Notation auf eine blockstrukturierte Notation durch Synchronisationen. Diese erlauben eine Abbildung unzulässig verschachtelter Konstrukte, die nur aus parallelen Gateways bestehen, in eine äquivalente blockstrukturierte Form.

Könnte die blockstrukturierte Notation ADEPT um weitere Elemente bzw. Konstrukte erweitert werden, um die Abbildbarkeit zu erhöhen?

Eine Erweiterung könnte das Einfügen eines Elementes wie das inklusive Gateway sein. Die Semantik des inklusiven Gateway als Join-Element, bei dem das Gateway auf alle aktivierten Pfade wartet und erst nach erfolgreicher Ausführung aller aktivierten Pfade den nachfolgenden Pfad aktiviert, fehlt in dieser Form als einzelnes Element in ADEPT. Ein blockstrukturiertes Konstrukt zur Modellierung dieses Umstands ist natürlich durch exklusive und parallele Gateways simulierbar, wäre aber subjektiv über ein eigenes Element intuitiver modellierbar.

Nach der Beantwortung ausgewählter Fragen zur Abbildbarkeit der BPMN auf ADEPT bleibt insgesamt festzustellen, dass die mangelnde vollständige Abbildbarkeit unstrukturierter Prozessmodelle auf Workflows auf die Konzeption der Notationen zurückzuführen ist. Der Wunsch der strukturell uneingeschränkten Modellierung ist nicht mit den unabdingbaren Ansprüchen der Software, die die Prozesse bzw. Workflows ausführen sollen, nach Sicherheit vor Ausführungsfehlern zu vereinbaren. Es sollte aber möglich sein, durch angepasste Algorithmen eine weitestgehende Abbildung der unstrukturierter Prozessmodelle durchzuführen, um die Workflows nicht komplett nachmodellieren zu müssen.

Literaturverzeichnis

- [1] F. Nordsieck, *Die schaubildliche Erfassung und Untersuchung der Betriebsorganisation*, 6. Auflage. C.E. Poeschel Verlag, Stuttgart, 1962.
- [2] Object Management Group, "Unified modelling language," Specification Version 2.1.2, November 2007.
- [3] O. M. Group, "Business Process Modeling Notation, v1.1," <http://www.bpmn.org/Documents/BPMN%201-1%20Specification.pdf>, Tech. Rep., 2008.
- [4] J. Recker, "BPMN modeling: Who, where, how and why," *BPTrends*, vol. 5, no. 3, 2008.
- [5] F. Nordsieck, *Betriebsorganisation. Lehre und Technik, Textband*. 2. Auflage. C.E. Poeschel Verlag, Stuttgart, 1972.
- [6] C. R. von Hagen and W. Stucky, *Business-Process- und Workflow-Management: Prozessverbesserung durch Prozess-Management*, Teubner, 2004.
- [7] M. Hammer and J. Champy, *Reengineering the corporation : A manifesto for business revolution*, 1st ed., Harper Business, 1993.
- [8] M. Weske, *Business Process Management: Concepts, Languages, Architectures*. Springer, 2007.
- [9] T. Davenport, *Process innovation: Reengineering work through information technology*, Harvard Business School Press, 1993.
- [10] L. Fischer, Ed., *BPM and Workflow Handbook 2007*. Workflow Management Coalition, 2007.
- [11] D. Georgakopoulos, M. F. Hornick, and A. P. Sheth, "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure," *Distributed and Parallel Databases*, vol. 3, no. 2, pp. 119–153, 1995.
- [12] N. Josuttis, *SOA in der Praxis: System-Design für verteilte Geschäftsprozesse*, 1. Auflage. Dpunkt Verlag, Heidelberg, 2008.
- [13] Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Bartek Kiepuszewski and Alistair P. Barros, "Workflow Patterns," *Distributed and Parallel Databases*, vol. 14, no. 1.

- [14] W. M. P. van der Aalst, A. H. M. ter Hofstede, and M. Weske, "Business Process Management: A Survey," in *Business Process Management*, 2003, pp. 1–12.
- [15] R. Liu and A. Kumar, "An Analysis and Taxonomy of Unstructured Workflows," in *Business Process Management*, W. M. P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, Eds., vol. 3649, 2005, pp. 268–284.
- [16] B. Kiepuszewski, A. H. M. ter Hofstede, and C. Bussler, "On Structured Workflow Modelling," in *CAiSE*, ser. Lecture Notes in Computer Science, B. Wangler and L. Bergman, Eds., vol. 1789. Springer, 2000, pp. 431–445.
- [17] R. Hauser, M. Friess, J. M. Kuster, and J. Vanhatalo, "Combining Analysis of Unstructured Workflows with Transformation to Structured Workflows," in *EDOC '06: Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 129–140.
- [18] B. Kiepuszewski, "Expressiveness and Suitability of Languages for Control Flow Modelling in Workflows," Ph.D. dissertation, Queensland University of Technology, Brisbane, Australia, 2002.
- [19] N. Russell, A.H.M. ter Hofstede, W.M.P. van der Aalst and N. Mulyar, "Workflow Control-Flow Patterns - A Revised View," <http://www.workflowpatterns.com/documentation/documents/BPM-06-22.pdf>, 2006.
- [20] Wil M.P. van der Aalst and Arthur H.M. ter Hofstede, "Verification of workflow task structures: A petri-net-based approach," *Inf. Syst.*, vol. 25, no. 1, pp. 43–69, 2000.
- [21] Manfred Reichert and Peter Dadam, "A Framework for Dynamic Changes in Workflow Management Systems," in *DEXA Workshop*, 1997, pp. 42–48.
- [22] Manfred Reichert, Stefanie Rinderle and Peter Dadam, "ADEPT Workflow Management System: Flexible Support for Enterprise-Wide Business Processes – Tool Presentation –," in *W.M.P van der Aalst et al.: BPM 2003, LNCS 2678, Springer-Verlag Berlin Heidelberg 2003*, 2003, pp. pp. 370–379.
- [23] Manfred Reichert and Peter Dadam, "ADEPT flex – Supporting Dynamic Changes of Workflows Without Losing Control," *Journal of Intelligent Information Systems*, vol. 10, no. 2, pp. 93–129, 1998.
- [24] S. White, "Introduction to BPMN," <http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf>, Object Management Group, Tech. Rep., 2004.
- [25] Chun Ouyang, Marlon Dumas, Arthur H.M. ter Hofstede and Wil M.P. van der Aalst, "Pattern-based translation of BPMN process models to BPEL web services," *International Journal of Web Services Research (JWSR)*, vol. 5, no. 1, pp. 42–62, 2007. [Online]. Available: <http://eprints.qut.edu.au/archive/00006810/>

- [26] Chun Ouyang, Wil M.P. van der Aalst, Marlon Dumas and Arthur H.M. ter Hofstede, "Translating BPMN to BPEL," <http://eprints.qut.edu.au/archive/00003615/>, 2006.
- [27] J. Recker and J. Mendling, "On the Translation between BPMN and BPEL: Conceptual Mismatch between Process Modeling Languages," in *CAiSE 2006 Workshop Proceedings - Eleventh International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD 2006)*, T. Latour and M. Petit, Eds., June 2006, pp. 521–532.
- [28] A. Alves, A. Arkin, S. Askary, B. Bloch, F. Curbera, Y. Golland, N. Kartha, Sterling, D. König, V. Mehta, S. Thatte, D. van der Rijn, P. Yendluri, and A. Yiu, "Web services business process execution language version 2.0," OASIS Committee Draft, May 2006.
- [29] J. Recker, M. Indulska, M. Rosemann and P. Green, "How good is BPMN really? Insights from theory and practice," in *Proceedings of the 14th European Conference on Information Systems. Goeteborg, Sweden, 2006*, pp. 1582–1593. [Online]. Available: <http://eprints.qut.edu.au/archive/00004636/01/4636.pdf>
- [30] Petia Wohed, Wil M. P. van der Aalst, Marlon Dumas, Arthur H. M. ter Hofstede and Nick Russell, "Pattern-based analysis of BPMN - an extensive evaluation of the control-flow, the data and the resource perspective (revised version)," <http://www.workflowpatterns.com/documentation/documents/BPM-06-17.pdf>, 2006.
- [31] Kevin Göser, Martin Jurisch, Hilmar Acker, Ulrich Kreher, Markus Lauer, Stefanie Rinderle, Manfred Reichert and Peter Dadam, "Next-generation Process Management with ADEPT2," in *BPM (Demos)*, ser. CEUR Workshop Proceedings, M. Adams and S. W. Sadiq, Eds., vol. 272. CEUR-WS.org, 2007.
- [32] T. Bauer, "Effiziente Realisierung unternehmensweiter Workflow-Management-Systeme," Ph.D. dissertation, Universität Ulm, 2001.
- [33] Stefanie Rinderle, Manfred Reichert and Peter Dadam, "Flexible Support of Team Processes by Adaptive Workflow Systems," *Distributed and Parallel Databases*, vol. 16, no. 1, pp. 91–116, 2004.
- [34] Wasim Sadiq and Maria E. Orłowska, "Applying Graph Reduction Techniques for Identifying Structural Conflicts in Process Models," *Lecture Notes in Computer Science*, vol. 1626, p. 195ff, 1999.
- [35] Chun Ouyang, Marlon Dumas, Wil M.P. van der Aalst and Arthur H.M. ter Hofstede, "From Business Process Models to process-oriented Software Systems: The BPMN to BPEL Way," <http://eprints.qut.edu.au/archive/00005266/>, 2006.