

Peter Loos, Markus Nüttgens,
Klaus Turowski, Dirk Werth (Hrsg.)

MobIS 2008

**Modellierung betrieblicher Informationssysteme –
Online-Proceedings, Workshop-Papers
including EPK2008, KobAS2008 and ModKollGP2008**

**27.–28. November 2008
Saarbrücken, Germany**

Organized by

Institut für Wirtschaftsinformatik (IWi)
im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI)
Universität des Saarlandes
Campus D32, Stuhlsatzenhausweg 3, D-66123 Saarbrücken

This Online-Proceedings complement the Lecture Notes in Informatics (LNI)
Proceedings of the Conference MobIS2008, Volume P-141

The conference was organized by the Special Interest Group on Modelling Business Information Systems (SIGMoBIS) within the German Informatics Society (GI), in cooperation with the Working Committee for ‚Geschäftsprozessmanagement mit Ereignis-gesteuerten Prozessketten‘ (WI-EPK) and ‚Komponentenorientierte betriebliche Anwendungssysteme (WI-KobAS) as well as the ‚Institut für Wirtschaftsinformatik‘ (IWi) im ‚Deutschen Forschungszentrum für Künstliche Intelligenz‘ (DFKI).

Volume Editors

Prof. Dr. Peter Loos
Institute for Information Systems (IWi) at DFKI,
Campus D 3 2,
D-66123 Saarbruecken, Germany
Email: peter.loos@iwi.dfki.de

Prof. Dr. Markus Nüttgens
Universität Hamburg
WISO Fakultät, Wirtschaftsinformatik
Von-Melle-Park 9, 20146 Hamburg, Germany
Email: markus.nuettgens@wiso.uni-hamburg.de

Prof. Dr. Klaus Turowski
Universität Augsburg
Wirtschaftsinformatik und Systems Engineering
Universitätsstr. 16, 86159 Augsburg, Germany
Email: klaus.turowski@wiwi.uni-augsburg.de

Dr. Dirk Werth
Institute for Information Systems (IWi) at DFKI,
Campus D 3 2,
D-66123 Saarbruecken, Germany
Email: dirk.werth@iwi.dfki.de

Preface

Modellierung von EPKs im Web mit Oryx.....	7
<i>Stefan Krumnow, Gero Decker und Mathias Weske</i>	
Zur automatischen Ermittlung von Testszenarien aus EPK-Schemata	18
<i>Oliver Skroch</i>	
Methoden zur adaptiven Anpassung von EPKs an individuelle Anforderungen vor der Abarbeitung	31
<i>Jens Brüning, Peter Forbrig</i>	
Zur Weiterentwicklung der Spezifikation betrieblicher Softwarekomponenten	44
<i>Jörg Ackermann</i>	
Formale Kontrolle kollaborativer B2B-Geschäftsprozesse.....	64
<i>Janina Fengel, Michael Rebstock, Carlo Simon</i>	
Process Modeling for Network Organizations – The Impact of the Process Environment	80
<i>Stefan Große Böckmann, Jan vom Brocke, Kai Riemer, Daniel Richter</i>	
Generische Einschränkung der Modellierungsfreiheit in fachkonzeptuellen Modellen	95
<i>Martin Juhrisch, Gunnar Dietz, Werner Esswein</i>	
Ein konfiguratives Metamodellierungswerkzeug.....	109
<i>Patrick Delfmann, Sebastian Herwig, Milan Karow, Łukasz Lis</i>	

SEMAT – Ein Werkzeug zur ontologiebasierten Analyse und zum Vergleich von Prozessmodellen	128
<i>Martin Kluth, Frederik Ahlemann, Frank Teuteberg</i>	
Ein Vorschlag zur Messung der Nutzung von Referenzprozessmodellen – Konzept und Anwendung am Beispiel des SCOR-Modells.....	148
<i>Peter Fettke</i>	
Integration der Finanz- und Produktionsdomäne nach ANSI/ISA-95 in einem REA-System.....	161
<i>Walter S.A. Schwaiger, Robert Ranzi</i>	
Semantic Business Process Management: An Empirical Case Study	165
<i>Sebastian Stein, Christian Stamber, Marwane El Kharbili, Pawel Rubach</i>	
Policy-Based Semantic Compliance Checking for Business Process Management	178
<i>Marwane El Kharbili, Sebastian Stein, Elke Pulvermüller</i>	

Modellierung von EPKs im Web mit Oryx

Stefan Krumnow, Gero Decker und Mathias Weske
Hasso-Plattner-Institut an der Universität Potsdam
stefan.krumnow@student.hpi.uni-potsdam.de
(gero.decker,mathias.weske)@hpi.uni-potsdam.de

Abstract: Ereignisgesteuerte Prozessketten werden häufig zur Prozessmodellierung in betrieblichen Anwendungen eingesetzt. Besonders hilfreich sind Prozessmodelle dann, wenn die Prozessbeteiligten auf die Modelle einfach zugreifen und sie bei Bedarf sogar verändern können. In diesem Beitrag stellen wir Oryx vor, eine webbasierte Plattform zur Erstellung, Speicherung und gemeinsamen Bearbeitung von Prozessmodellen. Oryx ist erweiterbar, so dass projektspezifische Erweiterungen einfach realisierbar sind. Auf diese Weise können auch neue Forschungsergebnisse mit vergleichsweise geringem Aufwand implementiert und evaluiert werden.

1 Einleitung

Prozessmodellierung erlaubt es, die Abläufe in einer Organisation zu verstehen, Veränderungsbedarf zu identifizieren, neue Abläufe zu kommunizieren und Anforderungen an Informationssysteme abzuleiten. Prozessmodelle dokumentieren, welche Aktivitäten anfallen, wer daran beteiligt ist, welche Artefakte benötigt und erstellt werden und welche Abhängigkeiten zwischen den einzelnen Aktivitäten bestehen.

Ereignisgesteuerte Prozessketten (EPKs [KNS92]) haben sich als Modellierungstechnik etabliert. Durch ihre Syntax und Semantik ermöglichen sie eine effiziente Kommunikation zwischen den Prozessbeteiligten. Verschiedene Personen müssen dabei Zugriff auf ein Prozessmodell haben, zum einen als Ersteller des Modells als auch als Leser des Modells. Während in manchen Fällen alle Beteiligten der gleichen Organisation angehören, gibt es zahlreiche Szenarien, in denen Beteiligte aus verschiedenen Organisationen zusammenarbeiten müssen. Gerade im Falle von Outsourcing von Prozessen oder Prozessteilen ist es wichtig, ein gemeinsames Prozessverständnis zu erreichen. Auch im öffentlichen Sektor, wo Prozesse verschiedene Behörden überspannen, ist dies ein zentrales Thema.

Die entsprechenden Modellierungswerkzeuge müssen also einen einfachen Zugriff verschiedenster Beteiligter auf die Modelle ermöglichen. Oft wird dies aber durch den Einsatz verschiedener Softwareversionen, fehlender Softwarelizenzen oder inkompatibler Benutzerverwaltungen erschwert.

Außerdem besteht Bedarf nach Modellierungswerkzeugen in der akademischen Community. Solche Werkzeuge sollen es erlauben, mit Spracherweiterungen zu experimentieren, Verifikations- und Validierungstechniken umzusetzen, Prozessmetriken zu testen und verschiedenste andere Algorithmen zu realisieren. Durch eine Plattform, die bereits viele

der üblichen Funktionen bereitstellt, kann unnötige Reimplementierung vermieden werden und die Integration verschiedener Prototypen erreicht werden.

Dieses Papier stellt Oryx vor, eine offene Plattform, die durch seine Erweiterbarkeit die schnelle Entwicklung von akademischen Prototypen ermöglicht. Darüber hinaus erlaubt Oryx einfachen Zugriff auf Modelle und vermeidet Versionskonflikte, da es sich dabei um eine webbasierte Lösung handelt.

Das Papier gliedert sich in folgende Abschnitte: Zunächst wird die Architektur von Oryx erläutert, bevor Abschnitt 3 die EPK-spezifischen Teile von Oryx einführt. Abschnitt 4 gibt einen Überblick über verwandte Arbeiten und Abschnitt 5 schließt das Papier mit Zusammenfassung und Ausblick ab.

2 Oryx-Systemarchitektur

Oryx ist ein webbasiertes Prozessmodellierungstool. Abbildung 1 zeigt einen Screenshot des Systems. Zu sehen ist, dass Oryx in einem Standardwebbrowser läuft und das Modell durch eine URL eindeutig identifiziert wird. Dieser Link kann an andere Modellierer verschickt werden, sodass diese einfachen Zugriff auf das Modell haben, ohne Software auf dem eigenen Rechner installieren zu müssen. Voraussetzung ist natürlich, dass entsprechende Zugriffsrechte für das Modell vergeben wurden. Diese sind nötig, da nicht jeder ein Prozessmodell einsehen können soll. Andere Modellierer können, wieder entsprechende

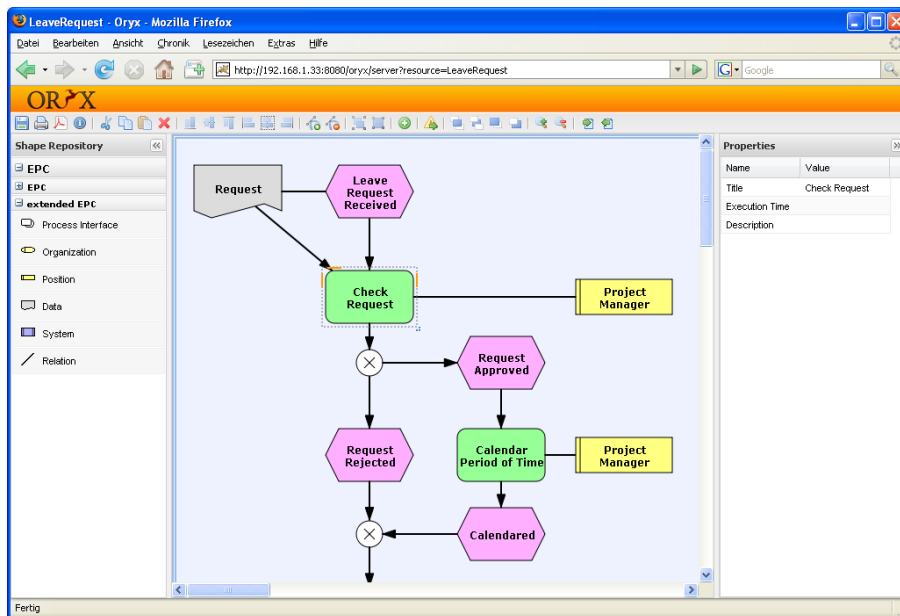


Abbildung 1: Oryx zur Modellierung einer EPK

Zugriffsrechte vorausgesetzt, das Modell modifizieren. Kollaboratives Modellieren über das Web ist somit möglich.

Oryx ist gerade für Szenarien geeignet, in denen Modellierer aus verschiedenen Organisationen auf die gleichen Modelle zugreifen müssen. Zum einen umgeht man mit Oryx das Problem, dass unterschiedliche (und möglicherweise inkompatible) Versionen von Software in den verschiedenen Organisationen installiert sind. Zum anderen besteht beim verteilten Zugriff nicht das übliche Problem, dass verschiedene Organisationen unterschiedliche Benutzerverwaltungen verwenden, oder dass man ein separates Benutzerkonto eigens für das Prozessmodellierungswerkzeug anlegen und pflegen muss. Dies ist möglich, da die Authentifizierung über OpenID [ope07] realisiert ist. Bestehende Benutzerkonten können somit wieder verwendet werden.

Oryx ist darauf ausgelegt, einfach sowohl um neue Notationen als auch um zusätzliche Funktionalität erweitert zu werden. Da er als Open-Source Projekt zur Verfügung steht, kann jede an Prozessmodellierung interessierte Organisation Weiterentwicklungen vornehmen. Hierbei profitieren vor allem Forscher, die für neue Modellierungsverfahren nicht mehr eigene Insel-Prototypen erstellen müssen, sondern sie innerhalb einer existierenden Lösung integrieren und erproben können.

Auf Grund seiner guten Erweiterbarkeit unterstützt Oryx neben EPKs auch weitere Modellierungssprachen wie die *Business Process Modeling Notation* (BPMN) in den Versionen 1.0 [bpm06] und 1.1 [bpm08], sowie Petri- und Workflow-Netze [vdAvH02]. Neben Standardmodellierungsfunktionen wie Copy&Paste, dem Ausrichten von Modellelementen oder dem Layouting von Kanten, werden auch notationsabhängige Funktionen angeboten. So können BPMN-Diagramme in Petri-Netze transformiert oder EPKs im- und exportiert werden. Diagramme einer jeden Notation lassen sich zudem auch als PDF oder PNG exportieren.

2.1 Genutzte Technologien

Um Oryx als Webseite in einem Browser anzeigen lassen zu können, ist die Verwendung verschiedener Webtechnologien nötig. Bevor jedoch auf diese eingegangen werden kann, soll zunächst die Architektur von Oryx vorgestellt werden (siehe Abbildung 2). Oryx besteht aus einem in den Browser geladenen Client, in dem das Modell editiert wird, und einem Backend, das eine Menge von Prozessmodellen vorhält.

Zur Anzeige des Editor-Clients werden vor allem JavaScript-Routinen verwendet, die als Teil eines Dokuments in einen Browser geladen werden. Dieses über eine URL adressierbare Dokument enthält ein Prozessmodell, das im Embedded RDF (eRDF [Dav06]) Format vorliegt. eRDF ermöglicht es, Metadaten direkt in ein HTML Dokument zu schreiben und daraus RDF zu extrahieren. Die Benutzung von RDF macht Oryx-Modelle einfach portierbar.

Die mit dem Prozessmodell mitgelieferten JavaScript Routinen sind dafür verantwortlich, die grafische Benutzerschnittstelle zu erzeugen und Modellierungsfunktionalität bereitzustellen. Modellelemente des aktuellen Diagramms finden sich als JavaScript-Objekte im

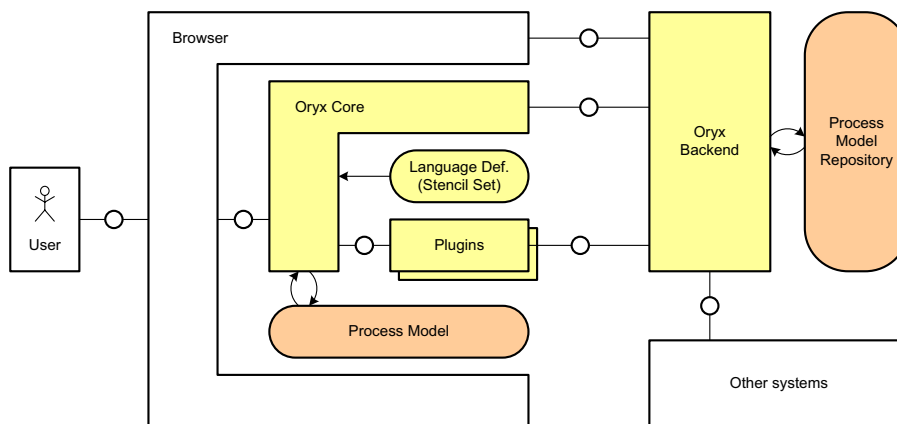


Abbildung 2: Oryx-Systemarchitektur

Browser wieder und können so manipuliert werden. Beim Speichern eines Modells werden die Elemente im eRDF Dokument asynchron an das Backend geschickt und dort in einer Datenbank gesichert.

Die Benutzerschnittstelle, die neben der Zeichenfläche aus Toolbar, Shape Repository und Property Pane besteht, ist mit Hilfe des ExtJS-Frameworks¹ erstellt. ExtJS ist eine JavaScript-UI-Bibliothek, die Widget-Typen für Rich Internet Application bereitstellt. Die Zeichenfläche enthält Modellelemente, die als Scalable Vector Graphics (SVG) in den Browser geladen werden.

Wie bereits beschrieben ist jedes Prozessmodell im Oryx durch eine URL eindeutig adressiert. Dadurch können Modelle durch bloßes Weiterreichen der URL ausgetauscht werden. Darüber hinaus können Modellressourcen unterschiedliche Repräsentationen aufweisen. So kann z.B. neben dem im Oryx editierbaren Dokument auch eine PDF-Repräsentation eines Prozesses angefordert werden.

Durch die OpenID-basierten Authentifizierung können Änderungen an Prozessmodellen – ähnlich wie in Wikis – nachverfolgt werden. Auf kritischen Ressourcen können zudem die Rechte, die ein Nutzer hat, eingeschränkt werden. Das System kennt hierbei die drei Rollen *Owner*, *Contributor* und *Viewer*, wobei ein Contributor ein Modell verändern und ein Viewer das Modell lediglich ansehen darf.

2.2 Stencil Set Konzept

Die Erweiterbarkeit ist eine der wichtigsten Eigenschaften von Oryx. Um verschiedene Notationen in Modellierungsprojekten zu unterstützen, wurde Oryx mit einem Mechanismus zum Auflösen beliebiger aus Knoten- und Kantentypen bestehender Modellierungs-

¹Siehe <http://extjs.com>

sprachen versehen. Dabei wird eine solche Notation im Oryx durch ein so genanntes *Stencil Set* repräsentiert. Stencil Sets werden beim Öffnen eines Prozessmodells generisch in den Editor geladen und erzeugen dort z.B. Einträge im Shape Repository und in der Property Pane.

Kernstück eines Stencil Sets ist eine in der JavaScript Object Notation (JSON) verfassten Datei, die das Metamodell der Notation enthält. In ihr werden die Typen aller Modellelemente definiert. Jeder Elementtyp ist entweder Knoten oder Kante und verfügt über eine ID, einen Namen und eine Beschreibung. Zur grafischen Repräsentation existiert für jeden Typ eine SVG Datei, die als Vorlage für die modellierten Elemente genutzt wird.

Darüber hinaus verfügt ein Typ über eine Menge von Properties, die bei der Modellierung gesetzt werden können. Beispiel hierfür ist der Name eines EPK Ereignisses oder die URL eines durch eine Prozess-Schnittstelle referenzierten Subprozesses. Jede Property hat einen Datentypen und einen Standardwert. Properties können an XML-Knoten innerhalb der SVG Datei des gleichen Elementtyps gebunden werden. Dadurch erhalten die Propertywerte während der Modellierung Einfluss auf die grafische Erscheinung des Elements. Durch diesen Mechanismus wird zum Beispiel der Name einer EPK Funktion innerhalb des sie repräsentierenden abgerundeten Rechtecks angezeigt.

Um die Beziehungen, die zwischen Modellelementen bestehen dürfen, auszudrücken, können im Stencil Set Kompositionsregeln angegeben werden. Dabei gibt es Regeln, die festlegen, ob Elemente zweier Typen durch Knoten oder Kanten eines bestimmten Typs verbunden werden dürfen. So kann beispielsweise ausgedrückt werden, dass ein Ereignis mit einer Funktion, nicht aber mit einem anderen Ereignis verbunden werden darf. Außerdem kann geregelt werden, ob Elemente eines Typs in Elementen eines anderen Typs enthalten sein dürfen. Auf Grundlage dieser Bedingungen wird beim Modellieren geprüft, ob neue Beziehungen korrekt sind. Darüber hinaus werden aus den Verbindungsregeln Vorschläge für nachfolgende Elemente erzeugt, die dann durch Benutzung eines Kontextmenüs einfach zu erstellen sind.

Die Erweiterung von Oryx um eine zusätzliche Notation ist sehr einfach: Es muss nur ein neuer Unterordner im Stencil Set Verzeichnis angelegt werden. In diesem Ordner werden die JSON sowie die Icon- und SVG-Grafikdateien der neuen Notation gespeichert. Anschließend kann sofort damit begonnen werden, Modelle der neuen Notation zu erstellen.

2.3 Plugin Konzept

Neben der Erweiterbarkeit durch Stencil Sets kann Oryx auch funktional durch Plugins erweitert werden. Ein Plugin wird beim Laden von Oryx initialisiert und registriert dabei seine Funktionalität im Nutzerinterface. Durch Nutzerinteraktion kann die Funktionalität des Plugins angestoßen werden. Dabei hat es Zugriff auf die JavaScript-Objekte des aktuellen Prozessmodells sowie auf alle anderen geladenen Ressourcen.

Der Großteil der Modellierungsfunktionalität von Oryx ist durch Plugins implementiert: Während der Oryx Core aus Abbildung 2 für den Aufbau des Editor Layouts verantwortlich ist, werden Funktionen wie das Ausrichten von Modellelementen oder auch das Spei-

chern der Modelle durch austauschbare Plugins realisiert. Zudem sind auch das Shape Repository, die Property Pane und selbst die Toolbar, in der die meisten anderen Plugins vertreten sind, als Plugin implementiert.

Plugins werden in JavaScript programmiert. Außerdem muss ein Plugin in der *plugins.xml*-Datei eingetragen werden. Dabei kann auch spezifiziert werden, dass ein Plugin nur für Modelle eines bestimmten Stencil Sets geladen werden soll.

In den vergangenen Monaten wurden schon zahlreiche Plugins entwickelt: So existieren Im- und Exporter für Petri-Netze und ein Transformator, der BPMN in ausführbare Petri-Netze übersetzt. Auch im Bereich der EPKs werden Plugins genutzt: Ein Syntax-Check kann genutzt werden, um die modellierten EPKs zu überprüfen. Außerdem können diese in EPML exportiert und aus EPML importiert werden.

3 Modellierung von EPKs mit Oryx

Ereignisgesteuerte Prozessketten sind eine leicht verständliche, semi-formale Notation, die in den frühen 1990er Jahren entwickelt wurde [KNS92]. Sie können gut dazu genutzt werden, Unternehmensabläufe zu modellieren. Dabei nutzen EPKs bipartite Abfolgen von Ereignissen und Funktionen, die durch Konnektoren verzweigt und zusammen geführt werden können.

Im Folgenden wird gezeigt, welche Modellelementtypen existieren und in Oryx unterstützt werden. Außerdem werden spezielle Modellierungsfunktionalitäten für Syntax Checks sowie Im- und Exports vorgestellt.

3.1 EPK Stencil Set

EPKs bestehen in erster Linie aus Ereignissen und Funktionen. Funktionen stellen aktive Elemente dar, die Aufgaben ausführen und dabei Daten erstellen oder verändern. Ereignisse sind passive Komponenten, die den erreichten Zustand innerhalb der aktuellen Prozesskette repräsentieren. Sie werden durch Funktionen erzeugt und verursachen ihrerseits die Ausführung weiterer Funktionen.

Außerdem existieren Konnektoren, die verwendet werden können, um Abläufe zu verzweigen oder wieder zusammenzuführen. Verzweigende Konnektoren haben einen Eingangs- und zwei Ausgangsflüsse, während zusammenführende Konnektoren zwei Eingänge und einen Ausgang haben. Dabei gibt es drei Typen: disjunktive (XOR), adjunktive (OR), und konjunktive (AND) Konnektoren.

Verzweigende ANDs modellieren, dass zur Laufzeit beide ausgehende Pfade abgearbeitet werden. XORs hingegen drücken aus, dass nur einer der beiden ausgehenden Pfade aktiviert wird. Die Entscheidung welcher der beiden Pfade das ist, wird durch das Auftreten eines der beiden nachfolgenden disjunktiven Ereignisse getroffen. Aus diesem Grund darf direkt nach einem verzweigenden XOR keine Funktion folgen. Das gleiche gilt für

verzweigende ORs. Hierbei ist es allerdings möglich, dass beide Ereignisse auftreten und daher auch beide Pfade abgearbeitet werden. Zusammenführende Konnektoren haben eine entsprechende Semantik: ANDs führen zwei ausgeführte Pfade zusammen, XORs vereinen zwei Pfade, von denen genau einer ausgeführt wurde und ORs tun dies mit zwei Pfaden, von denen wenigstens einer ausgeführt wurde.

All diese Standardkonstrukte [KNS92, Wes07] werden von Oryx unterstützt, wie Abbildung 3 zeigt. Dabei ist Oryx darauf ausgelegt, Diagramme zur Dokumentation existierender und neuer Prozesse zu ermöglichen und damit zum besseren Verständnis dieser beizutragen. Automatisch ausführbare Modelle sind hierbei nicht vorgesehen.

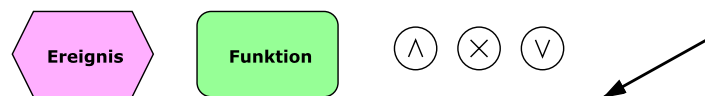


Abbildung 3: Übersicht der EPK Stencils

Ereignisse und Funktionen verfügen jeweils über eine Namen- und eine Beschreibungs-Property, die es dem Modellierer ermöglichen, den Modellelementen eine Bedeutung zuzuordnen. Im Zuge von EPK Modellierungsprojekten, die zusammen mit Industriepartnern durchgeführt wurden, ergab sich der Bedarf nach zusätzlichen Attributen. Diese Properties sind Werte, die vor allem aus bisherigen Ausführungen von Prozessen gewonnen werden konnten. So kann im Oryx angegeben werden, wie häufig ein bestimmtes Ereignis auftritt. Funktionen verfügen über eine Property, die angibt, wie lang die durchschnittliche Ausführungszeit ist und an Kanten können Ausführungswahrscheinlichkeiten annotiert werden. Dies ist vor allem für Kanten, die einer XOR Verzweigung entspringen, interessant und kann in Algorithmen genutzt werden, die die Komplexität von EPK verringern [PSW08b].

Neben den Standard-EPKs existieren noch erweiterte Ereignisgesteuerte Prozessketten (eEPKs). Diese zeichnen sich dadurch aus, dass in die normalen Abläufen auch Zuständigkeiten sowie datenbasierte Informationen integriert werden können. Während über die Menge an EPK Elementen Konsens herrscht, sind die Anzahl und Ausprägung von eEPK Elementen nicht festgeschrieben.



Abbildung 4: Übersicht der eEPK Stencils

Um das Modellieren einfach und übersichtlich zu halten, ist die Menge der unterstützten Elementtypen möglichst klein gehalten, wie Abbildung 4 zeigt. Dabei dient vor allem die *EPC Markup Language* (EPML [MN05]) als Orientierung.

Um verschiedene EPKs miteinander in Verbindung zu setzen, können Prozessschnittstellen genutzt werden. Diese können innerhalb oder am Ende einer EPK mit Hilfe von Sequenzfluss eingebunden werden und verweisen durch eine URL auf eine andere EPK, die

den Sub- oder Nachfolgeprozess modelliert. Zuständigkeiten lassen sich mit Hilfe von Stellen- oder im Falle von Prozessen zwischen verschiedenen Organisationen durch Organisationsknoten modellieren. Diese können mit Hilfe einer Relationsbeziehung an eine Funktion angebunden werden.

Auf die gleiche Art kann auch die Unterstützung oder Ausführung einer Funktion durch ein EDV-System modelliert werden. Letztendlich können Daten modelliert werden. Diese können sowohl mit Ereignissen als auch mit Funktionen in gerichteten oder ungerichteten Relationen auftauchen.

3.2 Syntax Checks

Neben der Auswahl von Modellelementen ist vor allem deren korrekte Kombination während der Modellierung von Bedeutung. Hierfür bietet Oryx eine zweistufige Überprüfung der Modelle während der Erstellung und zu einem späteren manuell gewählten Zeitpunkt als Unterstützung an.

Ein Stencil Set enthält, wie bereits erwähnt, neben Element-Definitionen auch Verknüpfungsregeln. So ist im EPK Stencil Set zum Beispiel festgelegt, dass auf ein Ereignis kein Ereignis folgen kann. Diese Regeln werden bei jeder Erstellung einer neuen Beziehung überprüft. Außerdem werden sie immer ausgewertet, wenn potenziell zu verbindende Elementtypen zu bestimmen sind (siehe Abbildung 5). Da diese Prüfung während des Modellierens vorgenommen wird, kann sie aus Performance-Gründen nur die lokale Umgebung einer Verbindung betrachten.

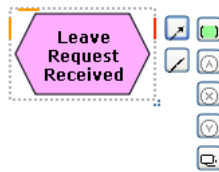


Abbildung 5: Vorschläge für nachfolgende Elemente im Oryx

Wie bereits in [MN03] ausgeführt, reicht eine solche lokale Betrachtung nicht aus um ein syntaktisch korrektes Modell garantieren zu können. Als Beispiel wollen wir die Überprüfung einer Kante von einem XOR Knoten zu einer Funktion anführen:

Ein Konnektor-Knoten kann in Abhängigkeit von der Anzahl ein- und ausgehender Kanten sowohl als zusammenführender als auch als verzweigender Konnektor genutzt werden. Während auf einen zusammenführenden XOR sowohl eine Funktion als auch ein Ereignis folgen darf, müssen auf einen verzweigenden XOR zwei Ereignisse folgen. Soll geprüft werden, ob der Konnektor mit einer nachfolgenden Funktion verknüpft werden darf, ist also das Wissen darüber von Nöten, ob der Konnektor verzweigend oder zusammenführend ist. Da dieses Wissen aber von anderen Verbindungsbeziehungen abhängt, die nicht mehr zur lokalen Umgebung der zu prüfenden Relation gehören, kann die Frage zu diesem Zeitpunkt nicht beantwortet werden.

Vollständige Syntax-Checks können zu einem späteren Zeitpunkt manuell oder zum Beispiel beim Speichern ausgelöst werden. Oryx bietet hierfür ein Framework an, das durch Nutzerinteraktion angestoßen einen Syntax-Check veranlasst und fehlerhafte Modellelemente markiert.

Dabei wird der Plugin-Mechanismus von Oryx genutzt, um den Syntax Check einzubinden. Das Plugin bietet dabei einen Button in der Toolbar an, an den eine Methode gebunden ist, die das in RDF serialisierte Prozessmodell an das Backend weiterreicht. Im Backend ist nun ein Java Servlet dafür verantwortlich, das RDF in eine Java Objektstruktur umzuwandeln. Anschließend kann die Objektstruktur umfassend auf syntaktische Fehler überprüft werden. Anders als in [GL06] passiert dies durch eine Java-Methode, die über alle Kanten und Knoten des Modells iteriert und dabei die gesamte Umgebung des Elements für dessen Syntax-Check einbezieht. Neben dem Syntax-Checker für EPKs gibt es auch eine Implementierung BPMN Diagramme.

Die Benutzung einer Umwandlung in Java Objekte hat mehrere Vorteile: Das verwendete Metamodell und der Umwandler stellen wiederverwendbare Komponenten dar. So können sie auch für weitergehende semantische Überprüfungen - wie Soundness-Checks [Men07] - oder für die Implementierung von Transformationen in andere Notationen, z.B. in Petri-Netze, verwendet werden. Darüber hinaus ist die Performance eines deployten Servlets besser als die von lokal im Browser laufenden JavaScript Routinen.

Das Ergebnis der Überprüfung wird zurück an das JavaScript Plugin im Oryx geschickt, welches fehlerhafte Elemente markiert. Dabei wird neben der ID eines Elements auch ein beschreibender Text übermittelt, der als Tooltip mit in die Markierung übernommen wird.

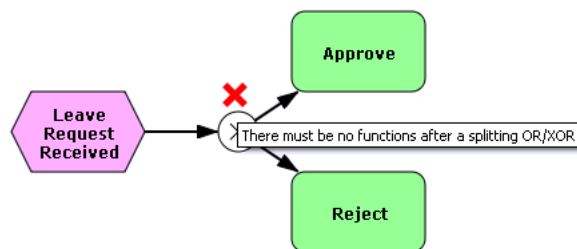


Abbildung 6: Markierung eines fehlerhaften Konnektors in Oryx

Abbildung 6 zeigt, wie der oben beschriebene Fall in einer nicht lokalen Syntax-Prüfung als Fehler erkannt und markiert werden kann. Durch einen weiteren Klick auf den Button in der Toolbar können die Markierungen wieder entfernt werden.

3.3 Im- und Exports

Damit mit Oryx modellierte EPKs auch in anderen Umgebungen genutzt werden können und vor allem auch um Ergebnisse bisheriger Arbeiten mit anderen Tools im Oryx fortzuführen, bietet Oryx Im- und Exportmöglichkeiten an.

Mit Hilfe eines weiteren Plugins wurden Funktionen in Oryx hinzugefügt, die einen Export nach EPML, bzw. einen Import aus EPML erlauben. EPML ist ein XML basiertes toolunabhängiges Austauschformat für EPKs [MN05]. Es kann unter Zuhilfenahme von XSL-Transformationen auch in AML, das ARIS-Austauschformat, transformiert werden [MN04].

Der Export eines Prozessmodells nach EPML wird durch den Nutzer per Button-Klick eingeleitet. Anschließend werden zwei Transformationen durchgeführt: Zunächst wird das im eRDF Format gespeicherte Modell in RDF transformiert. Hierfür kann ein gegebenes XSLT Stylesheet² verwendet werden. In einem zweiten Schritt muss dann das RDF in EPML umgewandelt werden, was wiederum durch ein XSLT Stylesheet realisiert wird. Die Umwandlung kann direkt Client-seitig im Plugin geschehen, da moderne Browser eigene XSLT-Engines anbieten. Die resultierende EPML Datei wird dem Benutzer in einem neuen Browsertab zur Verfügung gestellt.

Um eine EPK importieren zu können, muss sich der Nutzer ebenfalls im Oryx Editor befinden. Durch einen Button-Klick öffnet er einen Upload-Dialog, mit dem er eine EPML-Datei an ein Servlet im Backend schicken kann. Innerhalb des Servlets wird das EPML wieder mittels XSLT in eRDF umgewandelt. Das Ergebnis kann zurück an den Client gesendet werden, wo das Prozessmodell zur Anzeige kommt und evtl. durch den Nutzer gespeichert werden kann.

4 Verwandte Arbeiten

Marktführer auf dem Feld der EPK-Modellierungswerkzeuge ist das *ARIS Toolset* der IDS Scheer AG³. Dieses beruht auf dem Konzept der Architektur integrierter Informationssysteme (ARIS [Sch00]), das die Modellierung betriebswirtschaftlicher Abläufe unter den unterschiedlichen Gesichtspunkten der Organisation, der Daten, der Funktionen und der Steuerung vorsieht. Dabei werden diese auch auf verschiedenen Abstraktionsebenen zwischen Konzeptionierung und Implementation betrachtet.

Dementsprechend bietet das ARIS Toolset neben der Modellierung von EPKs auch weitere Notationen von Wertschöpfungsketten bis Organigrammen an. Außerdem existieren verschiedene Module, die auf die unterschiedlicher Benutzer zugeschnitten sind. So können Prozesse auf verschiedenen Abstraktionen modelliert und simuliert, aber auch implementiert und gewartet werden.

Durch eine Client-Server Architektur, können verschiedene Nutzer kollaborativ Modelle verwalten. Hierfür muss jedoch, anders als bei Oryx, die Client-Software auf jedem Rechner installiert werden, von dem ein Nutzer auf die Modelle zugreifen möchte.

Neben ARIS existieren aber auch weitere Tools, die im kommerziellen oder im wissenschaftlichen Bereich eingesetzt werden können, um EPKs zu modellieren. Mit *bflow*⁴ entsteht zur Zeit eine Eclipse-basierte Geschäftsprozessmodellierungsumgebung, die, ähnlich

²Siehe <http://research.talis.com/2005/erdf/extract-rdf.xsl>

³Siehe <http://www.ids-scheer.com>

⁴Siehe <http://www.bflow.org/>

wie Oryx, als Open-Source-Projekt vor allem Forschern die Möglichkeit der Integration eigener Konzepte geben soll.

Ebenfalls auf Eclipse setzt das Werkzeug *EPC Tools*⁵ auf. Dabei unterstützt es die Standard-EPK-Modellelemente, bietet aber auch Simulations- und Analysefunktionen an. Anders als Oryx und ARIS sind bflow* und EPC Tools Standalone Anwendungen, in denen man Modelle nur durch das Verschicken von Dateien oder die Integration von Versionsverwaltungs-Systemen austauschen kann.

Außer frei verfügbaren Shapes für das Modellierungstool *Visio*⁶, die das Erstellen von EPKs und eEPKs unterstützen, existiert mit *SemTalk* ein kommerziell vertriebener Visio-Aufsatz [FW01]⁷. Dieser bietet ähnlich wie Oryx eine Vielzahl von Notationen und dabei auch das Modellieren, den Austausch und die Analyse von EPKs an. SemTalk ist als Standalone Anwendung konzipiert, die aber auch Dokumente mit einem Repository abgleichen kann.

Auch wenn bereits Open-Source Modellierungstools in der EPK Community existieren, zeichnet sich Oryx vor allem durch seine Erweiterbarkeit um Notationen und Funktionalitäten sowie seinen kollaborativen Ansatz aus. Dadurch, dass er einfach als Webseite in einem Browser aufgerufen werden kann, ermöglicht Oryx einen im Vergleich zu kommerziellen Lösungen unkomplizierten und direkten Zugriff auf zentral hinterlegte Modelle.

5 Zusammenfassung und Ausblick

Dieses Papier hat EPK-Modellierung mit Oryx vorgestellt. Oryx ist eine technische Plattform, in der typische Funktionen eines grafischen Modellierungstools enthalten sind und die Erweiterungen einfach zulässt. Diese Erweiterungen können sowohl Spracherweiterungen als auch funktionale Erweiterungen sein. Syntax-Checks sind vorhanden und die Integration weiterer Verifikationsmechanismen ist geplant.

Als nächster Schritt steht die Integration von EPC-Soundness-Checks [Men07] an. Dies ermöglicht die Identifikation von unerreichbaren Ereignissen und Funktionen, sowie die Prüfung auf Abwesenheit von Deadlocks in den Modellen. Darüber hinaus ist die Integration von Prozessabstraktionsmechanismen geplant, die es ermöglichen, große Prozessmodelle auf Basis von Ausführungswahrscheinlichkeiten und durchschnittliche Ausführungszeiten automatisch zu reduzieren [PSW08a]. Außerdem wird zur Zeit an der Möglichkeit des eigenständigen Im- und Exports von ARIS Modellen, die im proprietären Format der AML ausgetauscht werden, gearbeitet.

Oryx steht allen Forschungsgruppen und Unternehmen offen, eigene Erweiterungen umzusetzen oder Funktionalität zu integrieren. In Kollaborationen mit anderen Universitäten hat sich bereits gezeigt, dass dies tatsächlich mit geringem Aufwand möglich ist und dass so neue Funktionalität schnell einem größeren Publikum zugänglich gemacht werden kann.

⁵Siehe <http://wwwcs.uni-paderborn.de/cs/kindler/research/EPCTools/>

⁶Siehe <http://office.microsoft.com/visio/>

⁷Siehe <http://www.semtalk.com/>

Danksagungen. Wir möchten uns beim Oryx-Team für die umfangreichen Entwicklungen bedanken.

Literatur

- [bpm06] Business Process Modeling Notation (BPMN) Specification, Final Adopted Specification. Bericht, Object Management Group (OMG), February 2006.
- [bpm08] Business Process Modeling Notation, V1.1. Bericht, Object Management Group (OMG), Jan 2008. <http://www.omg.org/spec/BPMN/1.1/PDF/>.
- [Dav06] Ian Davis. RDF in HTML. Bericht, Talis Information Limited, 2006. <http://research.talis.com/2005/erdf/wiki/Main/RdfInHtml>.
- [FW01] Christian Fillies und Frauke Weichhardt. SemTalk: A RDFS Editor for Visio 2000. Bericht, SC4, July 2001. <http://www.semtalk.com/pub/swws.htm>.
- [GL06] Volker Gruhn und Ralf Laue. Validierung syntaktischer und anderer EPK-Eigenschaften mit PROLOG. In Markus Nüttgens, Frank J. Rump und Jan Mendling, Hrsg., *Proc. of the 5th GI Workshop on Event-Driven Process Chains (EPK 2006)*, Seiten 69–84. GI-Arbeitskreis Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, 2006.
- [KNS92] Gerhard Keller, Markus Nüttgens und August-Wilhelm Scheer. Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)". Bericht Heft 89, Institut für Wirtschaftsinformatik Universität Saarbrücken, 1992.
- [Men07] Jan Mendling. *Detection and Prediction of Errors in EPC Business Process Models*. Dissertation, Vienna University of Economics and Business Administration, 2007.
- [MN03] Jan Mendling und Markus Nüttgens. EPC Syntax Validation with XML Schema Languages. In Markus Nüttgens und Frank J. Rump, Hrsg., *Proc. of the 2nd GI Workshop on Event-Driven Process Chains (EPK 2003)*, Seiten 19–30. GI-Arbeitskreis Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, 2003.
- [MN04] Jan Mendling und Markus Nüttgens. Transformation of ARIS Markup Language to EPML. In M. Nüttgens und F.J. Rump, Hrsg., *Proc. of the 3rd GI Workshop on Event-Driven Process Chains (EPK 2004)*, Seiten 27–38. GI-Arbeitskreis Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, 2004.
- [MN05] Jan Mendling und Markus Nüttgens. EPC Markup Language (EPML): An XML-based Interchange Format for Event-Driven Process Chains (EPC). Bericht, Vienna University of Economics and Business Administration, July 2005.
- [ope07] OpenID Authentication 2.0 - Final. Bericht, OpenID Foundation, December 2007. http://openid.net/specs/openid-authentication-2_0.html.
- [PSW08a] Artem Polyvyanyy, Sergey Smirnov und Mathias Weske. Process Model Abstraction: A Slider Approach. In *EDOC 2008: Proceedings of the 12th IEEE International Enterprise Distributed Object Computing Conference*. IEEE Computer Society, 2008.
- [PSW08b] Artem Polyvyanyy, Sergey Smirnov und Mathias Weske. Reducing the Complexity of Large EPCs. Bericht 22, Hasso-Plattner-Institute, 2008.

- [Sch00] August-Wilhelm Scheer. *ARIS Business Process Modeling*. Springer Verlag, 2000.
- [vdAvH02] Wil van der Aalst und Kees van Hee. *Workflow Management: Models, Methods, and Systems (Cooperative Information Systems)*. The MIT Press, January 2002.
- [Wes07] Mathias Weske. *Business Process Management*. Springer, 2007.

Zur automatischen Ermittlung von Testscenarien aus EPK-Schemata

Oliver Skroch

Wirtschaftsinformatik und Systems Engineering
Universität Augsburg
Universitätsstr. 16, D-86159 Augsburg
oliver.skroch@wiwi.uni-augsburg.de

Abstract: Für funktionale Systemtests großer, komponenten- und serviceorientierter betrieblicher Anwendungssoftware sind durchgängige Testscenarien sinnvoll, die für die wichtigsten, geschäftskritischen betrieblichen Abläufe des Anwenders vorzubereiten und durchzuführen sind. Zur Unterstützung der Ermittlung solcher „higher order“ Testfälle wird eine systematische Methode der Extraktion verzweigungsfreier Wege durch EPK-Schemata vorgeschlagen. Um die vorgestellte Methode praktikabel zu machen werden optionale Annahmen zur kombinatorischen Relaxation getroffen. Die Methode ist als Erweiterung der Open-Source Initiative „EPC-Tools“ implementiert.

1 Einleitung und Motivation

Im komponenten- und serviceorientierten Entwicklungsparadigma betrieblicher Anwendungssysteme [SGM02; Tu03] kann man fragen, wie fremdbezogene Komponenten bzw. Dienste möglichst früh und effizient überprüft werden [Bo81; We98]. Man geht dabei für die Seite der Komponentenhersteller bzw. Dienstelieferanten davon aus, dass sie die Außensicht auf ihre Komponenten und Dienste spezifizieren, um sie am Markt anzubieten [Tu03]. Man kann weiter annehmen, dass ein Anwender solche Spezifikationen auf ihre Eignung für seine zu automatisierenden betrieblichen Abläufe überprüft, bevor er die entsprechenden Komponenten und Dienste einsetzt bzw. bevor er sie auch nur bezieht. Jedoch können im allgemeinen nicht alle möglichen Abläufe einer zu automatisierenden Domäne gegen alle Eigenschaften der sie unterstützenden Software getestet werden. Das dürfte auch, zumindest bei nicht sicherheitskritischen Systemen, in den allerwenigsten Fällen wirtschaftlich sinnvoll sein.

Damit geht es für den nachfragenden Anwender zunächst darum, die für ihn wichtigsten, geschäftskritischen Abläufe unabhängig von den angebotenen Komponenten bzw. Diensten systematisch zu identifizieren. Sind diese kritischen Abläufe gefunden, dann eignen sie sich gut als Grundlage zur weiteren Ausarbeitung der tatsächlichen Prüfschritte, dabei v.a. zur Definition verzweigungsfreier Szenarien für Positivtests mit unabhängigen Orakeln („sunshine paths“ [Sk07, ST07]). Die Frage, die dieser Beitrag zu beantworten versucht ist damit, ob und wie aus einem betrieblichen Ablaufschema

verzweigungsfreie Wege als Grundlage für die Ausarbeitung von Testszenarien durch dieses Ablaufschema extrahiert werden können.

Der weitere Beitrag ist folgendermaßen aufgebaut: in Kapitel 2 werden weitere Arbeiten diskutiert, die mit dem hier vorgestellten Lösungsansatz verwandt sind oder für den Hintergrund und die verwendeten Begriffe eine Rolle spielen; in Kapitel 3 wird die Methode im Detail vorgestellt; Kapitel 4 stellt die dazu entwickelte Werkzeugenerweiterung vor; Kapitel 5 schließt den Beitrag mit einer Zusammenfassung und einem Ausblick.

2 Hintergrund, Begriffe, und verwandte Arbeiten

Für die Modellierung von Geschäftsprozessen gibt es heute viele kommerzielle Werkzeuge („workflow management systems“), die verschiedenen methodischen Ansätzen und Notationsgrammatiken folgen (z.B. [AH02; Mu02]). Dieser Beitrag geht von der Ereignisgesteuerten Prozesskette (EPK) [KNS92] aus, weil sie eine besonders intuitive und speziell im deutschsprachigen Raum weit verbreitete, semiformale Notation für die Modellierung betriebliche Abläufe ist, die schon seit einiger Zeit ergiebig erforscht und auch erfolgreich in der Praxis eingesetzt wird (z.B. [Sc02], [DJ05]).

Szenarien können als (Teil-)Modelle betrachtet werden und sind wichtige Bausteine der Softwareentwicklung, sie helfen bei der Erfassung von funktionalen Anforderungen und beeinflussen Design, Implementierung und funktionalen Test [Br00; Su03; CR05]. Bei der funktionalen Prüfung von Software auf Anwenderseite (d.h. Black-Box Systemtest [Be95]) sind szenario-orientierte Vorgehensweisen gut bekannt und werden z.B. bei der Testfallermittlung auf Grundlage von UML Anwendungsfällen („use cases“) eingesetzt [OA99; BL02; SZ04].

Prüfprozesse in der Softwareentwicklung werden allgemein etwa in [My79; Ri97; ISO01] diskutiert, für Test und Qualitätssicherung im komponentenorientierten Paradigma etwa in [We98; GTW03; Me03; Vi+03]. Für die Entwicklung – und dabei besonders für den funktionalen Systemtest – von flexiblen, komponenten- und serviceorientierten Anwendungssystemen werden wichtige Vorteile durch die Szenario-Orientierung diskutiert, cf. etwa [EDF96; SZ04].

Mit vollständig formalen Spezifikationen und „kleinen“ betrachteten Domänen sind szenario-orientierte Prüfmethode gut bekannt, z.B. in der technischen Informatik als Spuren („traces“) im Rahmen des „model checking“, vgl. etwa [CH03; Br+05] oder vertiefend [Ku+05]. Diese Ansätze lassen sich aber nur eingeschränkt auf semiformal und i.a. unvollständig beschriebene, funktional im Vergleich wesentlich größere Modelle betrieblicher Abläufe übertragen. Solche Modelle haben, wie z.B. in [DJ05] gezeigt wurde, eigene Besonderheiten, zumal es letztlich um „higher order“ Prüfschritte [My79, S. 103ff] geht, die über rein syntaktische Kriterien hinaus reichen.

Um trotzdem klar abgegrenzte Szenarien als Ausgangspunkt für die Definition weiterer konkreter Prüfschritte in großen und semiformalen EPK-Modellen zu finden, erweist

sich die Idee der „Workflow Patterns“ [Aa+03] (Ablauf- oder Kontrollfluss-Muster) als nützlich. Betrachtet man das Schema der betrieblichen Abläufe analog zum Kontrollfluss eines deterministischen Programms, dann kann man den gerichteten Graph des Ablaufschemas mittels weniger grundlegender Ablaufmuster aus [Aa+03] auf enthaltene verzweigungsfreie Szenarien analysieren. Ein lineares Szenario hat (in Anlehnung an die Spuren der linearen temporalen Logik) von jedem Knoten maximal eine ausgehende Kante in einen Folgeknoten, und das komplexe betriebliche Ablaufschema wird über die Ablaufmuster an den „Verzweigungsstellen“ in einzelne solche linearen Durchläufe zerlegt.

In [Sc00] wird ein Ansatz vorgeschlagen, der eine ähnliche Linearisierungsidee, die dort an Konzepte aus der Netzplantechnik angelehnt wird, nicht erst für die Gewinnung von Testszenerien sondern schon für die Konstruktion der Ablaufmodelle selbst verfolgt. In [MNN05] wird eine (hier nicht erforderliche) Erweiterung der EPK zur Unterstützung aller „Workflow Patterns“ aus [Aa+03] diskutiert.

Für die automatische Ermittlung dieser Szenarien aus den als EPK modellierten Ablaufschemata ist eine vollständig formale Syntax und Semantik für EPK notwendig, wobei es wesentlich um die Formalisierung der logischen Konnektoren (der „Verzweigungsstellen“ im Schema) geht. Dieser Beitrag baut dabei auf den Vorschlägen [Aa99; NR02; GL06] auf, und zusätzlich auf [Ri99] für die Wohlgeformtheit von Split-Join-Paaren im Ablaufschema, sowie auf [Kin06] für die Minimierung der sog. nicht-lokalen EPK-Semantik.

3 Beschreibung der Methode

3.1 Formalisierung von Notation, Syntax und Semantik

Zur graphischen Darstellung der EPK gibt es als Notationssymbole einen gerichteten Kantentyp und fünf unterschiedliche Knotentypen (Abbildung 1). Kantentyp ist der *Kontrollfluss*. Die fünf Knotentypen sind *Ereignis*, *Funktion*, und logischer Konnektor vom konjunktiven Typ („a und b“, *AND*), disjunktiven Typ („entweder a oder b“, *XOR*), und adjunktiven Typ („a oder auch b“, *OR*).



Abbildung 1: EPK Notationssymbole

Ein syntaktisch korrektes EPK-Schema A sei ein Tupel $A = (E, F, C, T, S_0)$ mit den folgenden Eigenschaften:

- E ist eine nichtleere Menge von Ereignisknoten.
- F ist eine nichtleere Menge von Funktionsknoten.
- C ist eine Menge von logischen Konnektoren, bestehend aus paarweise disjunkten Teilmengen C_{AND}, C_{XOR}, C_{OR} .
- E, F, C sind paarweise disjunkt.

- Der Kontrollfluss $T : (N \times N)$ ist eine Übergangsfunktion mit $N = E \cup F \cup C$.
- S_0 ist eine Menge von Startereignissen.

Ein syntaktisch korrektes EPK-Schema sei mit den folgenden Eigenschaften auch semantisch korrekt:

- 1) $G = (N, T)$ ist ein gerichteter, zusammenhängender und endlicher Graph.
- 2) Zwischen zwei Knoten existiert maximal eine Kontrollflusskante.
- 3) Funktionsknoten haben genau eine eingehende und genau eine ausgehende Kontrollflusskante.
- 4) Ereignisknoten haben maximal eine eingehende und maximal eine ausgehende Kontrollflusskante.
- 5) Konnektoren haben entweder genau eine eingehende und mehrere ausgehende Kontrollflusskanten (und heißen dann Split), oder mehrere eingehende und genau eine ausgehende Kontrollflusskante (und heißen dann Join).
- 6) Ereignisknoten sind nur mit Funktionsknoten verbunden, möglicherweise über Konnektoren.
- 7) Funktionsknoten sind nur mit Ereignisknoten verbunden, möglicherweise über Konnektoren.
- 8) Es gibt keinen Kreis, der nur aus Konnektoren besteht.
- 9) Es gibt mindestens ein Startereignis (ohne eingehende Kontrollflusskante) und mindestens ein Endereignis (ohne ausgehende Kontrollflusskante).
- 10) Nach Ereignisknoten folgt kein XOR-Split und kein OR-Split im Kontrollfluss (vgl. [KNS92, S. 15]).
- 11) Jeder Knoten ist von einem Startereignis erreichbar.
- 12) Von jedem Knoten ist ein Endereignis erreichbar.

Zusätzlich seien semantisch wohlstrukturierte Split-Join-Paare gefordert, so dass der öffnende Split und der schließende Join eines Paares jeweils vom gleichen Typ sind (vgl. vertiefend [Ri99, S. 2]).

3.2 Behandlung der grundlegenden Ablaufmuster

Die musterbasierte Analyse der EPK-Schemata zur Ermittlung verzweigungsfreier Wege erfolgt anhand der drei fundamentalen Modellierungsprimitive Sequenz, Selektion und Iteration, die auch den „Workflow Patterns“ aus [Aa+03] zugrunde liegen.

Sequenz

Die Sequenz ist die fundamentale Idee des Ablaufs von diskreten Einzelschritten nach einander (Abbildung 2). Die gängige semantische Interpretation ist Implikation bzw. Kausalität.

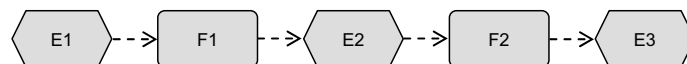


Abbildung 2: Lineare Sequenz

Lineare Sequenzen sind eben die verzweigungsfreie Wege, die das angestrebte Ergebnis der Linearisierung darstellen. Die Linearisierung eines EPK-Schema erzeugt alle möglichen solchen Wege vom Startereignis zum Endereignis. Ein EPK-Schema, das wie in Abbildung 2 nur aus einem verzweigungsfreien Weg besteht, ist bereits linear.

Selektion – Split-Konnektor

Split-Konnektoren besitzen genau eine eingehende und mehr als eine ausgehende Kante (vgl. 3.1, Nr. 9). Die Selektion mit Splits stellt die fundamentale Idee eines sich in der Zukunft verzweigenden Ablaufs dar. Abbildung 3 zeigt die zwei Muster, wobei der Split nach einem Ereignis (linke Seite der Abbildung 3) nur vom Typ *AND*, nicht aber vom Typ *OR* und auch nicht vom Typ *XOR* sein kann (vgl. 3.1, Nr. 10).

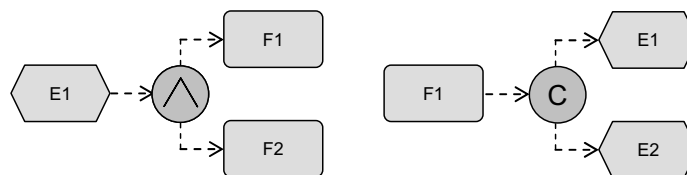


Abbildung 3: Selektion mit Split-Konnektor

Tabelle 1 zeigt die Ergebnisse der Linearisierung dieser Muster aus Abbildung 3, abhängig vom Konnektortyp. Die Muster werden in alle möglichen Abläufe expandiert.

Typ	Linearisierung nach Ereignis	Linearisierung nach Funktion
<i>AND</i>	(E1, F1, F2), (E1, F2, F1)	(F1, E1, E2), (F1, E2, E1)
<i>XOR</i>	nicht definiert	(F1, E1), (F1, E2)
<i>OR</i>	nicht definiert	(F1, E1), (F1, E2), (F1, E1, E2), (F1, E2, E1)

Tabelle 1: Linearisierung für Split-Konnektor

Gängige semantische Interpretationen für den *AND*-Split ist die Parallelbearbeitung, für den *XOR*-Split die Entscheidung. Der *OR*-Split ist nur schwierig intuitiv zu interpretieren. Er kann als Kombination aus *AND*- und *XOR*-Split behandelt werden, da die logische Semantik eine Kombination von *AND* und *XOR* darstellt.

Selektion – Join-Konnektor

Join-Konnektoren besitzen mehr als eine eingehende und genau eine ausgehende Kante (vgl. 3.1, Nr. 9). Die Selektion mit Joins stellt die fundamentale Idee sich vereinigender Abläufe dar. Daraus folgt, dass durch das jeweilige Schema in der Vergangenheit vor dem Konnektor mehr als ein Weg möglich ist. Abbildung 4 zeigt die zwei Muster.

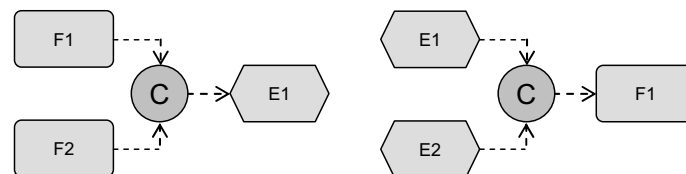


Abbildung 4: Selektion mit Join-Konnektor

Tabelle 2 zeigt die Ergebnisse der Linearisierung dieser Muster aus Abbildung 4, abhängig vom Konnektortyp. Die Muster werden in alle möglichen Abläufe expandiert.

Gängige semantische Interpretationen für den *AND*-Join ist die Synchronisierung, für den *XOR*-Join der Auslöser („trigger“). Der *OR*-Join ist ähnlich wie sein Split-Pendant nur schwierig intuitiv zu interpretieren, er kann aber ebenfalls als Kombination aus *AND*- und *XOR*-Join behandelt werden, da seine logische Semantik eine Kombination von *AND* und *XOR* darstellt.

<i>Typ</i>	Linearisierung vor Ereignis	Linearisierung vor Funktion
<i>AND</i>	(F1, F2, E1), (F2, F1, E1)	(E1, E2, F1), (E2, E1, F1)
<i>XOR</i>	(F1, E1), (F2, E1)	(E1, F1), (E2, F1)
<i>OR</i>	(F1, E1), (F2, E1), (F1, F2, E1), (F2, F1, E1)	(E1, F1), (E1, F2), (E1, E2, F1), (E2, E1, F1)

Tabelle 2: Linearisierung für Join-Konnektor

Es geht bei der Linearisierung der Join-Muster darum, den nach dem vereinigenden Join nächsten Schritt zu finden. Zwar ist, unabhängig von Typ des Join, der Folgeschritt nach dem Join immer eindeutig (vgl. 3.1, Nr. 5). Das Schema hat vor dem Join aber Verzweigungen, diese sind bis zum Join schon in die möglichen verzweigungsfreien Wege aufgespalten. Betrachtet man also einen einzelnen verzweigungsfreien Weg, dann wird jeder Join auf diesem Weg verzweigungsfrei erreicht. Die Behandlung der Join-Muster zur Linearisierung wäre soweit trivial. Es stellt sich aber die Frage, wie die miteinander zu kombinierenden Split-Join-Sequenzen bestimmt werden, da sie nicht unbedingt eindeutig sind.

Bestimmt werden muss zu jedem Join der entsprechend passende Split, und damit der genaue Nachbereich aller im Schema vorhandenen Splits. Die Forderung nach semantisch wohlstrukturierten Split-Join-Paaren (vgl. 3.1) verlangt, dass im Schema jedem Split ein Join vom gleichen Typ entspricht. Der Nachbereich beginnt mit dem Split-Konnektor selbst und wird entweder auf einem entsprechenden Join-Konnektor geschlossen oder endet mit einem Endereignis. Für einen Split ist der entsprechend passende Join: der unter allen Kindknoten (d.h. allen von Split aus erreichbaren Knoten), mittels Breitensuche (breadth first search), erste erreichbare Join vom gleichen Typ nach mindestens einem Ereignis- oder Funktionsknoten.

Die Linearisierung benötigt hier zur praktischen Handhabbarkeit zusätzlich noch sinnvolle Annahmen zur Einschränkung der zu prüfenden Kombinationen. Es kann für die Bildung von funktionalen Testszenarien von einer ausdrücklichen Betrachtung der unterschiedlichen Ausführungsreihenfolgen abgesehen werden, da angenommen werden kann, dass diese auch nicht modelliert sind. Zusätzlich können, in Anlehnung an die gängige Vorgehensweise des Testens an Randbereichen („boundary value“), bei *OR* mit

$n > 2$ Alternativen nur die „Randfälle“ $\binom{n}{1}$ und $\binom{n}{n}$ betrachtet werden.

Iteration

Die Iteration ist die fundamentale Idee eines sich gleichartig wiederholenden Ablaufs (Abbildung 5). Die gängige semantische Interpretation ist die Scheife.

Iterationen verwenden nur *XOR*-Konnektoren: die Joins der Iteration müssen offensichtlich disjunktiv sein, und für die entsprechenden Splits ist semantisch der gleiche Typ gefordert (vgl. 3.1).

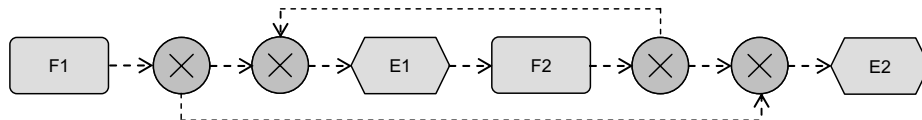


Abbildung 5: Iteration

Tabelle 3 zeigt die Ergebnisse der Linearisierung des Iterationsmusters aus Abbildung 5. Die Iteration wird in alle möglichen Abläufe expandiert.

Durchläufe	Linearisierung vor Ereignis
0	(F1, E2) – „oben geprüfte Schleife“
1	(F1, E1, F2, E2)
2	(F1, E1, F2, E1, F2, E2)
...	(F1, E1, F2, E1, F2, ..., E2)

Tabelle 3: Linearisierung der Iteration

Die Linearisierung der Iteration ist also einfach, benötigt aber zur praktischen Handhabbarkeit zusätzlich sinnvolle Annahmen zur Einschränkung der Anzahl der zu prüfenden Durchläufe. In Anlehnung z.B. an [Ri97] erfolgt die funktionale Prüfung mit keinem („oben geprüfte Schleife“), einem, oder $n > 1$ Durchläufen, wobei $n = 2$ gesetzt werden kann.

3.3 Komplexität und Relaxationsannahmen

Zur Bestimmung aller verzweigungsfreien Wege wird eine Erweiterung des BFS Standardalgorithmus („Breadth First Search“, Breitensuche) aus der Graphentheorie verwendet. Dabei ist BFS für die Auffindung aller Wege erweitert (Standard-BFS findet kürzeste Wege), und für die Behandlung der EPK-Konnektoren sowie der Iterationen angepasst.

Für die Zeitkomplexität des Algorithmus gilt $O(N + T)$, da alle möglichen Wege betrachtet werden, wofür jede Kante und jeder Knoten besucht wird und jeder Schritt eine konstante Zeit benötigt (mit Standard-BFS ist das der ungünstigste Fall, hier gilt es immer). Für die Platzkomplexität gilt ebenfalls $O(N + T)$, da alle bisher besuchten Knoten gespeichert werden bis die nächsttiefer Ebene des Graphen erreicht ist.

Die Methode ist damit für größere Probleme nur geeignet, wenn in den betrachteten EPK-Schemata die Anzahl der Knoten N (Ereignisse, Funktionen und Konnektoren) plus die Anzahl der Kanten T (Kontrollflüsse) mit zunehmender Tiefe des Schemas nicht

exponentiell wächst. Allgemein graphentheoretisch ist natürlich von exponentiellem Wachstum auszugehen. Wie aber aus den „No Free Lunch“ Theoremen [WM95; WM97] gefolgert werden kann, ist ein algorithmisches Such- oder Optimierungsverfahren möglichst gut an die untersuchte Problemklasse anzupassen, indem sinnvolle Annahmen zur Problemstellung getroffen werden.

Betrachtet man EPK-Schemata echter betrieblicher Abläufe in der Praxis, sind Splits (einschließlich Iterationen) oft nicht tief verschachtelt, sondern werden häufig ohne viele weitere Verschachtelungen durch entsprechende Joins geschlossen bzw. treffen auf Endereignisse. Stellt man die rein praktische Überlegung an, dass die hier betrachteten betriebliche Abläufe letztlich von Menschen verstanden, geplant, (zumindest teilweise) umgesetzt, verwaltet und verantwortet werden müssen, so ist diese Eigenart nachvollziehbar, und es ist daher auch anzunehmen, dass reale betriebliche Abläufe nicht beliebig komplex werden. Aus dieser Überlegung kann man auf eine nicht exponentielle Zunahme von $N+T$ bei grossen betrieblichen EPK-Schemata ausgehen. Weiterhin kann auch durch die mehrfach vorgeschlagene hierarchische Modellierung betrieblicher Abläufe (z.B. [KM94; Sc00]), bzw. durch die nachträgliche hierarchische Interpretation der Abläufe (z.B. mittels „blocking“ [ST07]), dem Problem der tief verschachtelten Modellteile und damit des exponentiell wachsenden Prüfaufwands begegnet werden. Unter diesen Annahmen für reale EPK-Schemata wäre die Standard-BFS als Grundlage der vorgeschlagenen Methode praktikabel.

Zusätzlich muss die Komplexität der vorgeschlagenen Methode noch hinsichtlich ihrer BFS Erweiterungen, der Behandlung von Iterationen und (nicht-disjunktiven) Konnektoren, eingeschränkt werden – soweit das ihrem letztlichen Ziel, der Erzeugung von Testszenarien für funktionale Systemtests auf Anwenderseite, nicht schädlich ist. Die folgende Analyse hinsichtlich der szenario-orientierten Prüfungsmöglichkeiten in EPK-Schemata führt zu solchen begründeten Relaxationsannahmen.

Sei $n > 1$ die Anzahl der ausgehenden Kontrollflusskanten eines Split, dann vermehrt (nur) ein Split die Anzahl der möglichen verzweigungsfreien Abläufe auf einem Weg im Schema. Bei einem Join wird zunächst der entsprechende Split bestimmt (wobei u.a. gefordert ist, dass Split und Join vom selben Konnektortyp sind), dann werden alle linearen Szenarien für das Split-Join-Paar gebildet. Deren Anzahl hängt vom Konnektortyp ab, wie Tabellen 4 und 5 zeigen.

<i>Split-Join-Typ</i>	Anzahl der Abläufe im Split-Join-Paar	
	vollständig	relaxiert
<i>AND</i>	$\#L(n, AND) = n!$	$\#LR(n, AND) = 1$
<i>XOR</i>	$\#L(n, XOR) = n$	$\#LR = \#L$

Tabelle 4: Anzahl der Szenarien in Split-Join-Paaren *AND* und *XOR*

Die linke Spalte in Tabelle 4 nennt den Typ des Split-Join-Paares, die linke Seite der rechten Spalte die Anzahl der Abläufe bei vollständiger linearer Expansion, und die rechte Seite der rechten Spalte die Anzahl der Abläufe bei relaxierter linearer Expansion. Für *AND* bedeutet die Relaxation die Nichtberücksichtigung der unterschiedlichen

Ausführungsreihenfolgen (Permutationen). Für *XOR* erscheint eine Relaxation für die spätere Bildung von Testszenarien nicht sinnvoll.

Für *OR* wird zunächst festgestellt, dass es gleichbedeutend mittels *XOR* gefolgt von *AND* modelliert werden kann. Damit ergibt sich eine Abhängigkeit der *OR* Relaxation von der Behandlung des *AND*. Die Anzahl der Szenarien ist in Tabelle 5 dargestellt.

Bei vollständig expandiertem *AND* ergeben sich mit ebenso vollständig expandiertem *OR* alle Variationen der hypergeometrischen Verteilung; mit relaxiertem *OR* werden in der vorgeschlagenen Methode nur die beiden „Randfälle“ $\binom{n}{1}$ und $\binom{n}{n}$ behandelt. Bei relaxiertem *AND* ergeben sich mit vollständig expandiertem *OR* $\binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n} = 2^n - 1$ Szenarien, und mit ebenfalls relaxiertem *OR* entsprechend

der vorgeschlagenen Methode nur $\binom{n}{1} + \binom{n}{n} = n + 1$ Szenarien.

<i>Split-Join OR</i>	Anzahl der Abläufe im Split-Join-Paar	
	<i>OR</i> vollständig	<i>OR</i> relaxiert
<i>AND</i> vollständig	$\#L(n, OR) = \sum_{k=1}^n k! \binom{n}{k}$	$\#LR(n, OR) = n! + 1$
<i>AND</i> relaxiert	$\#L(n, OR) = 2^n - 1$	$\#LR(n, OR) = n + 1$

Tabelle 5: Anzahl der Szenarien in Split-Join-Paaren *OR*

Für Iterationen wird davon ausgegangen, dass die funktionale Prüfung mit keinem, einem oder $n > 1$ Durchläufen erfolgt, wobei $n = 2$ gesetzt werden kann (in Anlehnung z.B. an [Ri97]). Die Iteration wird also bei „oben geprüfter Schleife“ nicht berücksichtigt und expandiert ansonsten in zwei verzweigungsfreie Szenarien.

Bei gleichzeitiger Relaxation von *AND* und *OR* und bei der Relaxation von Iterationen scheint also, unter den getroffenen Annahmen zur Ermittlung von Testszenarien, die Komplexität der vorgeschlagenen Methode für reale EPK-Schemata praktikabel.

4 Erweiterung der „EPC Tools“

Die hier vorgeschlagene Methode ist als Erweiterung der Open-Source Initiative „EPC Tools“ [CK05] auf der Eclipse Java IDE implementiert. Der Algorithmus liest in seiner ersten Phase eine EPK ein und konvertiert sie in eine Datenstruktur für Graphen; es wird dabei das EPK Austauschformat EPML [MN06] („EPC Markup Language“) unterstützt. In seiner zweiten Phase durchläuft der Algorithmus diese Datenstruktur und expandiert alle Verzweigungen und Iterationen entsprechend der beschriebenen Methode in einzelne verzweigungsfreie Wege.

Die Implementierung ist in der Lage, zwischen gewähltem Start- und Endknoten korrekte EPKs zu linearisieren. Die implementierten Relaxations-Parameter für die Behandlung von Schleifen, *AND* und *OR* Konnektoren erlauben es, das Verhalten des Algorithmus einzustellen. Die folgenden drei Parameter sind implementiert:

Iterativer Relaxations-Parameter: über diesen Parameter kann die Anzahl der zu testenden Iterationen in den Zyklen des EPK-Schemas eingestellt werden. Sinnvolle Werte sind 0 (nur bei „oben geprüften Schleifen“), 1 und 2.

Konjunktiver Relaxations-Parameter: ist dieser Parameter gesetzt, so werden Split-Join-Paare vom Typ *AND* zu einem einzigen Szenario, mit einer beliebigen Ablaufreihenfolge (Permutation) der parallel zu bearbeitenden Abläufe, expandiert; andernfalls werden $n!$ Szenarien expandiert, jeweils ein Szenario für jede mögliche Reihenfolge.

Adjunktiver Relaxations-Parameter: ist dieser Parameter gesetzt, so werden für $n > 2$ nur wie beschrieben die beiden „Randsituationen“ expandiert, andernfalls werden alle $2^n - 1$ möglichen Kombinationen expandiert. Dabei hängt die Behandlung der konjunktiven Teilkombinationen jeweils vom konjunktiven Relaxations-Parameter ab, es wird also je nachdem entweder nur ein Szenario mit beliebiger Reihenfolge gebildet, oder faktoriell alle kombinatorisch möglichen Szenarien.

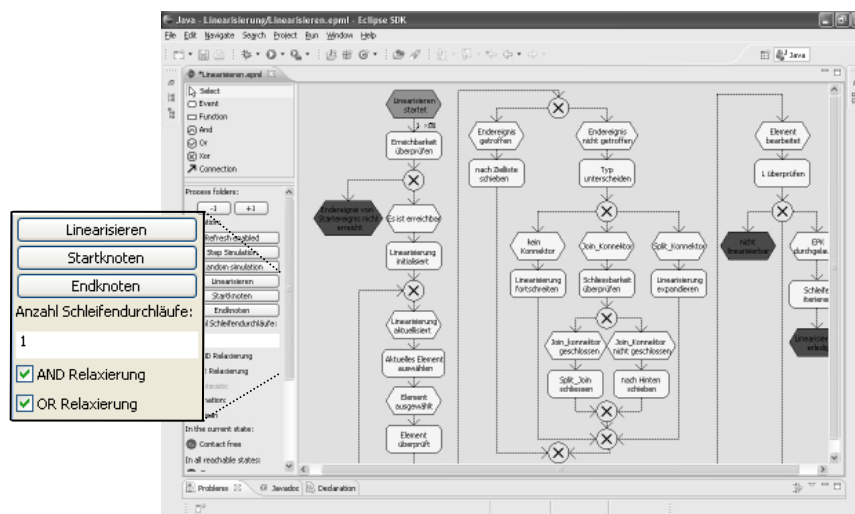


Abbildung 6: Erweiterung der Open-Source Initiative „EPC-Tools“ [CK05]

Die Implementierung ist weder instrumentiert noch auf Laufzeit oder Speicherbedarf optimiert. Unter Verwendung der Relaxations-Parameter wurden mit dieser Implementierung – neben kleineren funktionsprüfenden Beispielen – verschiedene EPK-Schemata aus betrieblichen Referenzmodellen auf einem Standard-PC ohne wahrnehmbare Laufzeit linearisiert.

Da Start- und Endknoten eines Linearisierungslaufs im Schema frei gewählt werden können, können auch verzweigungsfreie Szenarien unterschiedlicher Auflösung in

unterschiedlichen Teilen des Schemas durch entsprechende Anpassung der Parameter-Einstellungen für den jeweiligen Lauf erzeugt werden. Beispielsweise kann ein kleiner Teilprozess, in dem die Ausführungsreihenfolge der parallel modellierten Schritte als kritisch für den späteren Test angesehen wird, mit einem Linearisierungslauf unter eingeschalteter *AND* Kombinatorik analysiert werden.

5 Zusammenfassung und Ausblick

In diesem Beitrag wurde eine Methode vorgeschlagen, mit der unter bestimmten Annahmen zur praktischen Handhabbarkeit alle verzweigungsfreien Wege durch ein korrektes EPK-Schema betrieblicher Abläufe extrahiert werden können. Zur Methode wurde ein entsprechendes Werkzeug als Eclipse-Plugin Erweiterung der Open-Source Initiative „EPC-Tools“ implementiert.

Diese verzweigungsfreien Wege können nach bestimmten Kriterien priorisiert werden um so die wichtigsten, geschäftskritischen Abläufe zu identifizieren. Diese können dann – mit unabhängigen Orakeln versehen – als Grundlage für lineare Positivtest-Szenarien („sunshine paths“) zum funktionalen Black-Box Systemtest im komponenten- und serviceorientierten Entwicklungsparadigma verwendet werden.

Vorteile ergeben sich erstens durch die Möglichkeit, alle zu testenden Abläufe zu kennen und sie damit bewusst priorisieren zu können, sowie zweitens durch die Möglichkeit, Testorakel aufzustellen, die vom Entwicklungsprozess auf Herstellerseite unabhängig sind. Dadurch wird die Definition aussagefähiger Testmetriken auch bei „higher order“ Tests auf Domänenebene möglich.

Literaturverzeichnis

- [Aa99] van der Aalst, W. M. P.: Formalization and Verification of Event-driven Process Chains. *Information and Software Technology* 41 (1999) 639-650
- [Aa+03] van der Aalst, W. M. P., ter Hofstede, A. H. M., Kiepuszewski, B., Barros, A. P.: Workflow Patterns. *Distributed and Parallel Databases* 14 (2003) 5-51
- [AH02] van der Aalst, W. M. P., van Hee, K. M.: *Workflow Management: Models, Methods, and Systems*. The MIT Press, Cambridge, Ma, USA (2002)
- [Be95] Beizer, B.: *Black-Box Testing: Techniques for Functional Testing of Software and Systems*. John Wiley & Sons, New York, NY, USA (1995)
- [BL02] Briand, L., Labiche, Y.: A UML-Based Approach to System Testing. *Journal of Software and Systems Modeling* 1 (2002) 10-42
- [Bo81] Boehm, B. W.: *Software Engineering Economics*. Prentice-Hall, Englewood Cliffs, NJ, USA (1981)
- [Br00] de Bruin, H.: Scenario-Based Analysis of Component Compositions. In: *Generative and Component-Based Software Engineering: Second International Symposium (GCSE2000): Revised Papers*. 2000/10/09-12, Erfurt (2000) 129-146
- [Br+05] Broy, M., Jonsson, B., Katoen, J.-P., Leucker, M., Pretschner, A.: *Model-Based Testing of Reactive Systems: Advanced Lectures*. Springer, Berlin (2005)

- [CH03] Choi, Y., Heimdahl, M.: Model Checking Software Requirement Specifications using Domain Reduction Abstraction. In: 18th IEEE International Conference on Automated Software Engineering (ASE 2003). 2003/10/06-10, Montreal, Canada (2003) 314-317
- [CK05] Cuntz, N., Kindler, E.: On the semantics of EPCs: Efficient calculation and simulation. In: van der Aalst, W., Benatallah, B., Casati, F., Curbera, F. (Hrsg.): Business Process Management. LNCS 3649, Springer, Berlin (2005) 398-493
- [CR05] Carroll, J. M., Rosson, M. B.: Cases as Minimalist Information. In: 38th Hawaii International Conference on System Sciences (HICSS2005) Track 1. 2005/01/03-06, Big Island, Hi, USA (2005) 40b-
- [DJ05] van Dongen, B. F., Jansen-Vullers, M. H.: Verification of SAP Reference Models. In: van der Aalst, W. M. P., Benatallah, B., Casati, F., Curbera, F. (Hrsg.): Business Process Management. LNCS 3649, Springer, Berlin (2005) 464-469
- [EDF96] Ecklund, E., Delcambre, L., Freiling, M.: Change cases: use cases that identify future requirements. In: 11th ACM SIGPLAN Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA96). 1996/10/06-10, San Jose, Ca, USA (1996) 342-358
- [GL06] Gruhn, V., Laue, R.: Validierung syntaktischer und anderer EPK-Eigenschaften mit PROLOG. In: 5. Workshop Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (EPK2006). 2006/11/30-2006/12/01, Wien, Österreich (2006) 69-84
- [GTW03] Gao, J. Z., Tsao, H.-S. J., Wu, Y.: Testing and quality assurance for component-based software. Artech House, Boston, Ma, USA (2003)
- [ISO01] International Organization for Standardization: International Standard ISO/IEC 9126-1: Software engineering - Product quality - Part 1: Quality model. ISO/IEC, Genf, Schweiz (2001)
- [Kin06] Kindler, E.: On the semantics of EPCs: A framework for resolving the vicious circle. *Data & Knowledge Engineering* 56 (2006) 23-40
- [KM94] Keller, G., Meinhardt, S.: SAP R/3 Analyzer: Optimierung von Geschäftsprozessen auf Basis des R/3-Referenzmodells. SAP AG, Walldorf (1994)
- [KNS92] Keller, G., Nüttgens, M., Scheer, A.-W.: Sematische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)". In: Scheer, A.-W. (Hrsg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 89. Universität des Saarlandes, Saarbrücken (1992) 2-30
- [Ku+05] Kugler, H., Harel, D., Pnueli, A., Lu, Y., Bontemps, Y.: Temporal Logic for Scenario-Based Specifications. In: Tools and Algorithms for the Construction and Analysis of Systems (TACAS05). 2005/04/04-08, Edinburgh, UK (2005) 445-460
- [Me03] Meyer, B.: The Grand Challenge of Trusted Components. In: 25th International Conference on Software Engineering (ICSE2003). 2003/05/03-10, Portland, Or, USA (2003) 660-667
- [MN06] Mendling, J., Nüttgens, M.: EPC markup language (EPML): an XML-based interchange format for event-driven process chains (EPC). *Information Systems and e-Business Management* 4 (2006) 245-263
- [MNN05] Mendling, J., Neumann, G., Nüttgens, M.: Yet Another Event-Driven Process Chain. In: van der Aalst, W. M. P., Benatallah, B., Casati, F., Curbera, F. (Hrsg.): Business Process Management. LNCS 3649, Springer, Berlin (2005) 428-433
- [Mu02] zur Muehlen, M.: Workflow-based Process Controlling: Foundation, Design and Application of Workflow-driven Process Information Systems. Logos, Berlin (2002)
- [My79] Myers, G. J.: The Art of Software Testing. John Wiley & Sons, New York, NY, USA (1979)
- [NR02] Nüttgens, M., Rump, F. J.: Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen (Promise2002). 2002/10/09-11, Potsdam (2002) 64-77

- [OA99] Offutt, A. J., Abdurazik, A.: Generating Tests from UML Specifications. In: The Unified Modeling Language: Beyond the Standard: Second International Conference (UML99). 1999/10/28-30, Fort Collins, Co, USA (1999) 416-429
- [Ri97] Riedemann, E. H.: Testmethoden für sequentielle und nebenläufige Software-Systeme. Teubner, Stuttgart (1997)
- [Ri99] Rittgen, P.: Modified EPCs and Their Formal Semantics. In: Frank, U., Hampe, F. (Hrsg.): Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 19. Universität Koblenz-Landau, Koblenz (1999) 1-11
- [Sc00] Schimm, G.: Generic Linear Business Process Modeling. In: Liddle, S. W., Mayr, H. C., Thalheim, B. (Hrsg.): Conceptual Modeling for E-Business and the Web. LNCS 1921, Springer, Berlin (2000) 31-39
- [Sc02] Scheer, A.-W.: ARIS - vom Geschäftsprozess zum Anwendungssystem. Springer, Berlin (2002)
- [SGM02] Szyperski, C., Gruntz, D., Murer, S.: Component Software: Beyond Object-Oriented Programming. 2nd Edition, Addison-Wesley, London, UK (2002)
- [Sk07] Skroch, O.: Validation of Component-based Software with a Customer Centric Domain Level Approach. In: 14th Annual IEEE Conference and Workshop on the Engineering of Computer Based Systems (ECBS07), Doctoral Symposium. 2007/03/26-29, Tucson, Az, USA (2007) 459-466.
- [ST07] Skroch, O., Turowski, K.: Reducing Domain Level Scenarios to Test Component-based Software. Journal of Software 2 (2007) 64-73.
- [Su03] Sutcliffe, A. G.: Scenario-Based Requirements Engineering. In: 11th IEEE International Conference on Requirements Engineering (RE2003). 2003/09/08-12, Monterey, Ca, USA (2003) 320-329
- [SZ04] Strembeck, M., Zdun, U.: Scenario-based Component Testing Using Embedded Metadata. In: First International Workshop on Software Quality (SOQUA2004) and Workshop Testing Component-Based Systems (TECOS2004). 2004/09/27-30, Erfurt (2004) 31-45
- [Tu03] Turowski, K.: Fachkomponenten: Komponentenbasierte betriebliche Anwendungssysteme. Shaker, Aachen (2003)
- [Vi+03] Vincenzi, A. M. R., Maldonado, J. C., Delamaro, M. E., Spoto, E. S., Wong, W. E.: Component-Based Software: An Overview of Testing. In: Cechich, A., Piattini, M., Vallecillo, A. (Hrsg.): Component-Based Software Quality: Methods and Techniques. LNCS 2693, Springer, Berlin (2003) 99-127
- [We98] Weyuker, E. J.: Testing Component-Based Software: A Cautionary Tale. IEEE Software 15 (1998) 54-59
- [WM95] Wolpert, D. H., Macready, W. G.: No Free Lunch Theorems for Search. Working Papers of the Santa Fe Institute # 95-02-010, The Santa Fe Institute, Santa Fe, NM, USA (1995)
- [WM97] Wolpert, D. H., Macready, W. G.: No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation 1 (1997) 67-82

Methoden zur adaptiven Anpassung von EPKs an individuelle Anforderungen vor der Abarbeitung

Jens Brüning
Peter Forbrig

Fakultät für Informatik und Elektrotechnik
Universität Rostock
Albert-Einstein-Str. 21
18059 Rostock

Jens.Brueining@uni-rostock.de
Peter.Forbrig@uni-rostock.de

Abstract: In diesem Artikel werden Methoden vorgestellt, wie Prozessmodelle an individuelle Anforderungen vor der Abarbeitung des Workflows angepasst werden können. Bei vielen Prozessen entscheidet sich dann, welche Teile von der zugehörigen Ereignisgesteuerten Prozesskette (EPK) ausgeführt werden und welche nicht. Beispielsweise entscheidet der Kunde zu diesem Zeitpunkt häufig, welche von den angebotenen Dienstleistungen er in Anspruch nehmen möchte. Außerdem könnten zusätzliche Anforderungen vom Kunden aufgestellt werden, die im bisherigen Prozessmodell noch nicht berücksichtigt wurden. Um Prozessmodelle zu erhalten, die den individuellen Anforderungen genügen, werden im Folgenden Methoden vorgeschlagen und an bestimmten Stellen die Sprachmittel der Ereignisgesteuerten Prozesskette erweitert, um Prozessdefinitionen zu Beginn des Prozesses adaptiver zu machen.

1 Einleitung

Workflowmodelle dienen dem Angestellten als Anleitung, welche Aktivitäten er in welcher Abfolge bei bestimmten Geschäftsprozessen leisten soll, damit ein geregelter Ablauf im Unternehmen garantiert ist. Sie spiegeln des Weiteren Erfahrungen wieder, wie bestimmte Geschäftsvorfälle mit der angegebenen Prozessdefinition zum Erfolg geführt werden. Die Modelle können in unterschiedlichen Sprachen vorliegen. So sind Ereignisgesteuerte Prozessketten (EPKs), UML Aktivitätsdiagramme, Business Process Modelling Notation (BPMN) und Petrinetze zurzeit sehr verbreitet, um Workflows auszudrücken. Im weiteren Verlauf dieses Artikels werden EPKs für Prozessdefinitionen verwendet und insbesondere im Kapitel 3 erweitert.

Konventionelle, nicht adaptive Workflowspezifikationen und deren Anwendungen in Workflow Management Systemen (WfMS) sind teilweise zu starr, so dass auf wechselnde Anforderungen nicht flexibel reagiert werden kann.

Die in diesem Artikel vorgestellten Vorschläge sollen helfen, wechselnde Anforderungen unmittelbar vor Beginn des Prozesses in die Prozessdefinition einfließen zu lassen. Es können dann Prozessteile aus dem Workflow anhand von neuen Operatoren ausgeschlossen oder hinzugefügt werden. Der Idee, zum Anfang des Geschäftsprozesses Teile der Prozessdefinition auszulassen oder hinzuzufügen, liegt die Beobachtung zu Grunde, dass häufig kurz vor Beginn des Prozesses die Bedingungen mit dem Kunden geklärt werden und dieser dann angibt, welche Leistungen er nicht bzw. speziell noch zusätzlich erbracht haben möchte.

Der weitere Verlauf des Artikels ist wie folgt gegliedert: In Kapitel 2 werden die Grundlagen für die betrachteten EPKs kurz erläutert und erklärt, wie die dort vorhandenen Sprachmittel verwendet werden. Außerdem wird ein Fallbeispiel vorgestellt, das im Verlauf des Artikels zur Verdeutlichung der neuen Konzepte dienen soll. Im Kapitel 3 werden die neuen Konzepte eingeführt, mit denen man zur Instanzierungszeit Teile der Prozessdefinition nach den dort vorgestellten Regeln ausschließen oder hinzufügen und ggf. neu strukturieren kann. Kapitel 4 enthält schließlich eine Zusammenfassung und einen Ausblick, wie die neuen Konzepte umgesetzt werden können.

2 Grundlagen

Bei den Ereignisgesteuerten Prozessketten werden Workflows mit Ereignissen und Funktionen modelliert, die alternierend mit gerichteten Kanten verbunden werden. Der Funktionsbegriff wird in EPKs nach [KNS92] synonym zu Aufgaben verwendet. Des Weiteren gibt es Konnektoren, die den Kontrollfluss bzw. die Kanten aufspalten und zusammenführen können. Der AND-Konnektor kann parallele Prozessabläufe bzw. gegenseitig voneinander unabhängige Aufgaben modellieren. OR- und XOR-Konnektoren können alternativ auszuführende Aufgaben im Geschäftsprozess modellieren. Zusätzlich gibt es Hierarchisierungskonzepte durch Funktionsverfeinerungen, um die EPKs lesbarer zu machen. Genauere Einführungen in EPKs sind u.a. in [KNS92, NR02] zu finden.

Ein Beispiel für einen Geschäftsprozess ist in Abbildung 1 zu sehen. Es wurde ein mögliches Prozessmodell für eine Autowerkstatt erstellt, das in Abbildung 1a) angibt, wie mit einem ankommenden Auto zu verfahren ist. Die Funktion „Auto inspizieren und reparieren“ ist in Abbildung 1b) verfeinert, womit hier das Hierarchisierungskonzept angewendet wurde.

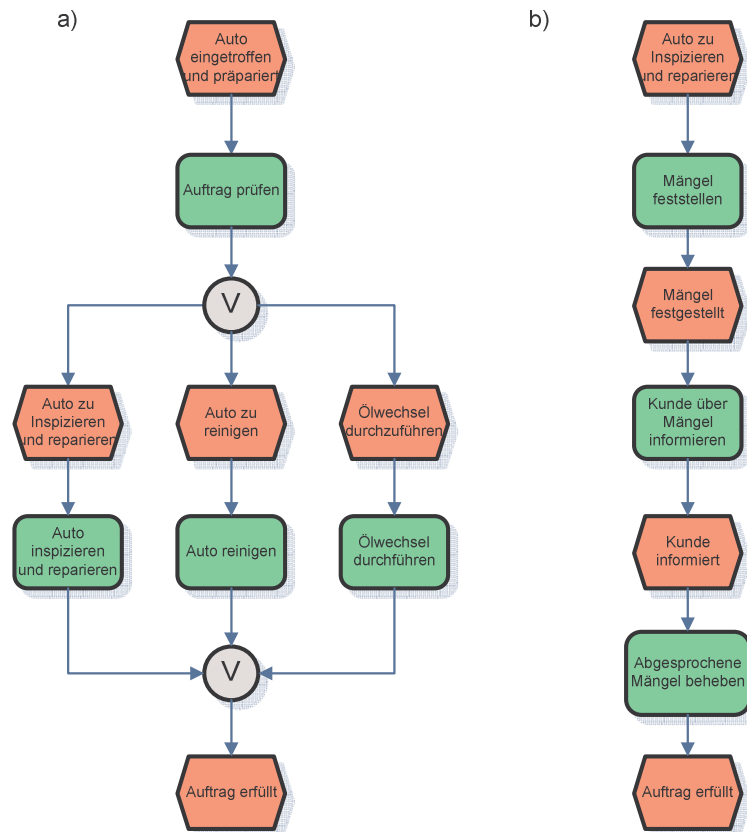


Abbildung 1: a) EPK Prozessdefinition zum Fallbeispiel: Auto inspizieren, reparieren, reinigen und Öl wechseln. b) Prozessverfeinerung für Prozess „Auto inspizieren und reparieren“

Der strukturierte Ablauf des Geschäftsprozesses ist mit den gerichteten Pfeilen modelliert, auf denen zur Ablaufzeit (Runtime) von den Anfangsereignissen ähnlich wie bei Petrinetzen Marken (bei EPKs auch Prozessordner genannt) über die Ereignisse, Funktionen und Kanten wandern. Das EPK-Prozessmodell an sich stellt hierbei noch keinen konkreten im Ablauf befindlichen Geschäftsprozess dar, sondern ist zunächst erst die Schablone dafür. Um einen konkreten Geschäftsprozess zu erhalten, muss dieser instanziiert werden. Erst dann befinden sich die Marken bzw. Prozessordner in der EPK, die jeweils kennzeichnen, welche Funktionen zur entsprechenden Zeit aktiv sind [NR02]. Das Instanziierungsprinzip von Prozessmodellen ist auch bei Workflow Management Systemen (WfMS) vorhanden, um konkrete Prozesse zu erzeugen, die dann vom WfMS gesteuert werden. Man unterscheidet zwischen Buildtime und Runtime [Ho95]. Während der Buildtime wird die Prozessdefinition mit einem Modellierungswerkzeug erstellt. Erst nach der Instanziierung durch das WfMS befindet man sich in der Runtime.

Bei EPKs wird zwischen Buildtime- und Runtime-Modellen unterschieden [ST05]. Buildtime-Modelle können im Gegensatz zu Runtime-Modellen noch nicht instanziiert werden. Sie enthalten Buildtime-Operatoren (OR_B und XOR_B), die sich von der Logik analog zu den üblichen Runtime-OR bzw. -XOR verhalten, nur mit dem Unterschied, dass sie während der Buildtime für das Schema und nicht während der Runtime, also während des Ablaufs der Prozessinstanz, entschieden werden. Erst nach Auflösung dieser Operatoren sind die Modelle Runtime-Modelle geworden, die dann instanziiert werden können.

Eine syntaktische Besonderheit besteht bei der Verwendung der Buildtime-Operatoren im Gegensatz zu den Runtime-Operatoren. In EPKs müssen unmittelbar vor den Runtime OR- und XOR-Operatoren Funktionen vorhanden sein, die die Entscheidungsfindung modellieren. Es dürfen keine Ereignisse unmittelbar vor diesen Konnektoren stehen [KNS92]. Diese Regel gilt nicht für Buildtime-Operatoren, da die dazugehörigen Entscheidungen außerhalb des Modells während der Buildtime getroffen werden. Sie können jedoch auch in die Runtime verlagert werden, indem der Buildtime-Operator durch einen Runtime-Operator mit der gleichen Logik ersetzt und eventuell eine nötige Entscheidungsfunktion unmittelbar davor hinzugefügt wird. Es gelten folgende Auflösungsregeln für die Buildtime-Operatoren nach [Sch98, ST05].

OR_B hat drei Möglichkeiten der Auflösung:

1. Auswahl der gewünschten Pfade zur Buildtime und Ersetzung des OR_B Operators durch ein Runtime AND Konnektor
2. Verlagerung in der OR-Entscheidung in die Runtime durch Hinzufügen einer Entscheidungsfunktion und eines Runtime OR-Konnektors
3. Verlagerung der Entscheidung in die Runtime durch Hinzufügen einer Entscheidungsfunktion und eines Runtime XOR-Konnektors

XOR_B hat zwei Möglichkeiten der Auflösung:

1. Auswahl eines gewünschten Pfades zur Buildtime
2. Verlagerung der XOR-Entscheidung in die Runtime durch Hinzufügen einer Entscheidungsfunktion und eines Runtime XOR-Konnektors

Die Buildtime-Operatoren werden in [Sch98, ST05] so motiviert, dass diese für Referenzmodelle (Buildtime-Modelle) verwendet werden, die für das jeweilige Unternehmen aufgelöst werden müssen. Je nachdem welche Prozessketten für das Unternehmen relevant sind, werden diese dann aus dem Buildtime-Modell ausgewählt und die nichtrelevanten ausgeschlossen. Die dort getroffenen Entscheidungen gelten für alle daraus erzeugten Prozessinstanzen. Die Buildtime-Operatoren werden damit sehr statisch verwendet im Gegensatz zu den neu eingeführten Operatoren in Kapitel 3.1.

Einen ähnlichen Ansatz zur Referenzmodellierung verfolgen die konfigurierbaren EPKs (C-EPCs) [RA05]. Konfigurierbare OR- und XOR-Konnektoren (OR^C und XOR^C) können dort mit den gleichen obigen Regeln aufgelöst werden [RA05]. Des Weiteren wurden dort konfigurierbare Funktionen eingeführt, die mit einer entsprechenden Buildtime-Entscheidung in der Prozesskette verbleibt, ausgelassen oder als optional ausführbare Funktion in die Prozesskette eingefügt wird. Zusätzlich können bei den konfigurierbaren EPKs noch Abhängigkeiten in Form von logischen Formeln an konfigurierbare Konnektoren und Funktionen annotiert werden, so dass beispielsweise bei Anwendung einer der obigen Auflösungsregeln zur Buildtime, eine konfigurierbare Funktion zwangsläufig eingefügt werden muss.

3 Operatoren für Prozessanpassungen zur Instanziierungszeit

In Abbildung 1 wurde ein Fallbeispiel als EPK vorgestellt, wie ein Teil eines Geschäftsprozesses in einer Autowerkstatt aussieht. Diese Prozessdefinitionen sind aber recht starr, so dass der Kunde vor Beginn des Geschäftsprozesses nicht noch individuelle Wünsche festlegen kann. Außerdem ist in der EPK von Abbildung 1a) eine Entscheidungsfunktion „Auftrag prüfen“ enthalten, die eigentlich nicht notwendig ist, wenn man die übliche Vorgehensweise im Unternehmen betrachtet, bei der dem ausführenden Mitarbeiter normalerweise vorher (ggf. per WfMS) mitgeteilt wird, was an dem konkreten Auto zu machen ist. Er muss damit nicht noch eine Entscheidung nach Eintreffen und Präparieren des Autos treffen (s. Entscheidungsfunktion „Auftrag prüfen“ in Abbildung 1a)), was gemacht werden muss. Vorschläge für eine realitätsnähere Modellierung mit Entscheidungsoperatoren, die vorher aufgelöst werden können, werden in diesem Abschnitt vorgestellt.

Die im Folgenden eingeführten Operatoren werden zur Instanziierungszeit aufgelöst. Diese Zeit kann die Verhandlungen zwischen Kunden und Unternehmen widerspiegeln, in der die Modalitäten für den Auftrag geklärt werden. Wir befinden uns hier noch nicht in der Runtime, da zu diesem Zeitpunkt der Auftrag noch nicht erteilt wurde, die EPK für die konkrete Instanz noch angepasst werden muss und noch keine Marken aktive Prozesse in der EPK identifizieren. Beachtet man jedoch, dass das unternehmensspezifische Modell schon in der Buildtime erzeugt wurde und dieses beispielsweise von einem WfMS bereits interpretiert wird, um die Anpassungen für die konkrete Prozessinstanz vorzunehmen, befindet man sich auch nicht mehr in der Buildtime. Man könnte die Zwischenzeit in der die Instantiationtime-Operatoren aufgelöst werden als Instantiationtime bezeichnen, die zeitlich nach der Buildtime und unmittelbar vor der Runtime liegt.

Es ergeben sich damit zwei grundlegende Konfigurationsphasen für EPKs:

1. In der Buildtime-Phase werden aus unternehmensübergreifenden Referenzmodellen durch Auflösung der Buildtime-Operatoren unternehmensspezifische Modelle erzeugt, in dem die Buildtime-Operatoren entsprechend der Regeln (s. Kapitel 2) aufgelöst werden.

2. In der Instantiationtime-Phase wird unmittelbar vor Ausführung der Modellinstanz das zugehörige Modell angepasst, jedoch ausschließlich für die zu erzeugende Prozess-Instanz. Die Instantiationtime-Operatoren werden entsprechend der im Folgenden vorgestellten Regeln aufgelöst.

Es können in Referenzmodellen jetzt also drei Sorten von Operatoren bzw. Konnektoren existieren. Buildtime-, Instantiationtime- und Runtime-Operatoren, die entsprechend ihrer Gattung entweder in einer der zwei Konfigurationsphasen vor Ausführung oder während des Ablaufs entschieden werden. Die Konfigurationsphasen können noch weiter unterteilt werden. Bei C-EPCs wird mit partiell konfigurierbaren EPKs darauf hingewiesen, dass es mehrere Konfigurationsphasen geben kann [RA05]. Eine grundlegende Differenzierung dieser Phasen wurde mit den 2 obigen vorgenommen.

3.1 Entscheidungsoperatoren für Instanziierungszeit

Die hier eingeführten Operatoren sind OR- und XOR-Operatoren, die zur Instanziierungszeit aufgelöst werden. Die Auflösungssemantik entspricht bis auf eine kleine Erweiterung (s.u.) der gleichen wie der beiden in Abschnitt 2 vorgestellten Buildtime-Operatoren. Sie brauchen des Weiteren ebenfalls keine Entscheidungsfunktionen unmittelbar vor den Operatoren, da die dazugehörigen Entscheidungen vorher in der Instanziierungszeit getroffen werden. Jedoch ist die Anwendung der Operatoren durch Auflösung in der Instanziierungszeit, welches für jede Prozessinstanz neu geschieht, abweichend von denen der Buildtime-Operatoren.

In Abbildung 2a) ist der Instantiationtime-OR-Operator (OR_I) abgebildet. Die Notation ist an die des Buildtime-Operators angelehnt. Die Unterscheidung lässt sich hier durch ein kleines I für Instantiationtime erkennen. Für die Auflösung des OR_I -Operators erscheint jedoch für das hier verwendete Fallbeispiel nur die erste der drei OR_B Auflösungsregeln aus Abschnitt 2 sinnvoll, da der Kunde dann festlegen soll, welche Leistungen er verbindlich haben möchte. Die Entscheidung dafür soll also zwingend zur Instantiationtime getroffen werden und darf in diesem Falle nicht in die Runtime verlagert werden. Diese optionale Einschränkung stellt eine Erweiterung zu den Buildtime-Operatoren dar.

Am Beispiel von Abbildung 2b) ist zu sehen, wie eine mögliche Instanz aus dem Modell aus Abbildung 1a) aussehen könnte. Aus dem Instantiationtime-OR ist in diesem Falle ein Runtime-AND geworden und der Pfeil, der zur Funktion „Auto reinigen“ führte ist gelöscht worden, so dass in diesem Fall der Kunde diese Leistung offenbar nicht erbracht haben wollte.

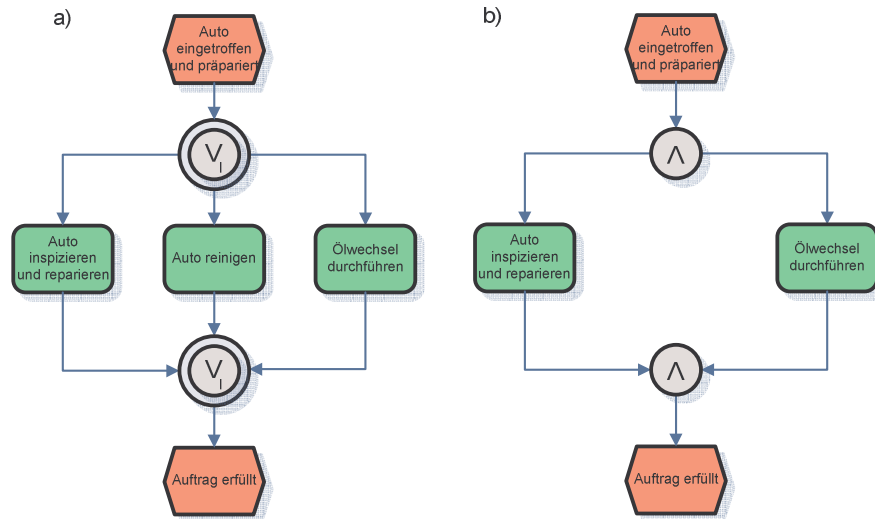


Abbildung 2: a) Instanziierung OR-Operators b) Modellinstanz nach Auflösung des Operators. Prozessmarken wurden weggelassen

Auch der XOR-Operator zur Instanziierungszeit hat sinnvolle Einsatzmöglichkeiten. Nehmen wir das Beispiel aus Abbildung 1b). Es könnten im Kundengespräch folgende 3 Möglichkeiten dem Kunden offeriert werden, wobei er sich zur Instanziierungszeit für eine entscheiden muss. 1.: Er wird in jedem Fall über die Schäden am Wagen benachrichtigt, 2.: Er wird ab einem bestimmten Betrag benachrichtigt, ansonsten wird automatisch repariert, 3.: Er wird nicht benachrichtigt und der Wagen wird in jedem Fall repariert.

Abbildung 3a) zeigt die entsprechende Spezifikation mit dem Instantiationtime-XOR-Operator. Bei der Instanziierung dieser EPK wird je nach Kundenwunsch ein Zweig ausgewählt. Für den Fall, dass der Kunde immer informiert werden möchte, wird bei der Instanziierung der linke Zweig ausgewählt. Die rechte Prozesskette ist für die entsprechende Prozessinstanz nicht mehr vorhanden.

Analog verhält es sich, wenn der Kunde wünscht, nicht über die Mängel informiert zu werden und damit die rechte Prozesskette wählt. Lediglich wenn der Kunde erst informiert werden möchte, wenn die Reparatur einen bestimmten Betrag überschreitet, ist die Entscheidung in die Runtime verlagert und es muss dann eine zusätzliche Entscheidungsfunktion nach XOR_B-Auflösungsregel 2 (s. Abschnitt 2) hinzugefügt werden. Mit der Funktion „Prüfen, ob Betrag überschritten“ wird dann entschieden, ob der Kunde informiert werden muss oder nicht. Dieser Fall ist in Abbildung 3b) zu sehen.

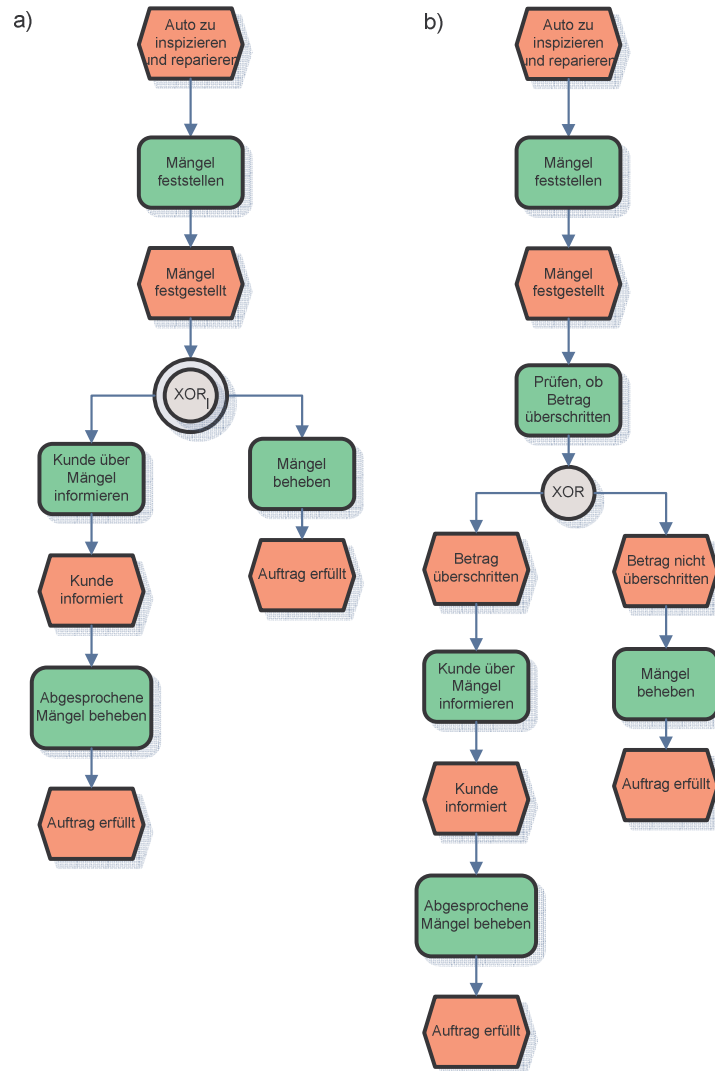


Abbildung 3: a) Instantiationtime-XOR-Operator in der Prozessdefinition. b) EPK nach Auflösung des Operators. Entscheidung ist hier in die Runtime verlegt worden.

Es erscheinen zusätzlich noch Erweiterungen für die Instantiationtime-Operatoren sinnvoll, wenn man bedenkt, dass die Modelle beispielsweise durch ein WfMS interpretiert und zur Erstellung der Prozessinstanz ausgewertet werden. Zunächst müssen die Instantiationtime-Operatoren in ein EPK-Austauschformat z.B. der EPML [MN05] eingeführt werden, damit sie von einem WfMS erkannt und interpretiert werden können. Entsprechendes wurde für die oben erwähnten C-EPCs schon gemacht [MRR05].

Bei der Modellierung in der Buildtime können Fragen mit den neuen Operatoren und Antwortalternativen für die ausgehenden Kanten verbunden werden, die im Instanziierungsprozess vom WfMS abgefragt werden. Im Beispiel von Abbildung 2a) kann z.B. die Frage: „Welche Leistungen wünscht der Kunde?“ mit dem Instantiationtime-OR verbunden werden, die vom WfMS zur Instanzierungszeit angefragt wird und die dazugehörigen Antwortalternativen mit den darauf folgenden Ereignissen oder Funktionen vorgeschlagen werden. Der Nutzer wählt die entsprechend gewünschten Leistungen aus und das WfMS löst damit den Operator auf.

Die Verwendung der neuen Instantiationtime-Operatoren erscheinen aber nicht nur für die Anpassung der EPK an die Anforderungen des Kunden sinnvoll, so wie es im bisherigen Beispiel verdeutlicht wurde. Z.B. entscheidet sich auch bereits bei der Instanzierung des Prozesses, um welche Automarke bzw. -typ es sich bei dem mit dem Prozess verknüpften Auto handelt. Somit kann die EPK auch darauf bereits während der Instanzierungszeit konfiguriert werden. Eine vorzeitige bessere Ressourcenplanung wird dadurch möglich.

3.2 Operator zum Hinzufügen zusätzlicher Funktionen

Bisher wurden Operatoren besprochen, die jeweils Prozessteile zur Instanzierungszeit ausschließen können. Es sind jedoch Situationen denkbar, in denen vom Kunden speziell gewünschte Dienstleistungen noch nicht im Unternehmensangebot und entsprechend noch nicht in der EPK vorgesehen sind, diese aber trotzdem erbracht werden sollen. Um hier reagieren zu können, wird in diesem Abschnitt ein neuer Operator eingeführt, der neu benötigte Funktionen für den jeweilig zu instanzierenden Prozess geregelt hinzufügen kann. Mit Verwendung des Operators werden die Stellen in der EPK gekennzeichnet, in denen neue Prozessteile in der EPK hinzugefügt werden dürfen.

Im Folgenden werden drei Möglichkeiten vorgestellt, wie die neu hinzuzufügenden Aufgaben bzw. Funktionen in die bestehende Prozesskette eingefügt werden können. Sie können erstens unabhängig voneinander (auch parallel), zweitens abhängig voneinander in einer Sequenz und drittens alternativ zueinander angeordnet werden. Da man im voraus keine Aussage treffen kann, wie die neuen Funktionen gegenseitig und zu den bisher in der Prozessdefinition vorhandenen Funktionen in Beziehung stehen, ist es sinnvoll dem Nutzer alle 3 Möglichkeiten des Hinzufügens zur Verfügung zu stellen. Im Zweifelsfall sollte die unabhängige, parallele Modellierungsart verwendet werden. Das Modell erlaubt dann eine flexible Ausführung während der Laufzeit und damit wäre das Modellierungsprinzip „Flexibility by design“ nach [AAH08, SMR07] realisiert. Das WfMS lässt damit viele, u.U. auch nicht sinnvolle Ausführungszeitpunkte der neu hinzugefügten Aufgabe zu. Die ausführende Person hat dann die Aufgabe, den richtigen Zeitpunkt zu bestimmen.

Um das adaptive Hinzufügen der drei oben formulierten Varianten zur Instanziierungszeit zu ermöglichen, wird hier nun ein neuer ADD-Operator vorgeschlagen. Mit dem ADD-Operator werden Nutzer-gesteuert neue Funktionen und Ereignisse für die zu erzeugende Instanz der EPK eingefügt. Die Stelle des Hinzufügens der neuen Funktion wird durch die aktuelle Position des ADD-Operators in der EPK angegeben. Er ist in Abbildung 4 abgebildet und muss ebenfalls wie die Entscheidungsoperatoren aus Kapitel 3.1 zur Instanziierungszeit aufgelöst werden. Die Auflösung kann mit Hilfe eines WfMS Nutzer-gesteuert geschehen, so dass auch dieser Operator in die EPML [MN05] neu eingeführt werden muss, damit das WfMS diesen entsprechend erkennen und interpretieren kann.

Die Auflösung des ADD-Operators geschieht wie folgt. Er dient jeweils als Platzhalter für eine neu hinzuzufügende Funktion oder er kann dupliziert werden. Letztendlich müssen alle ADD-Operatoren durch konkrete Funktionen und die zugehörigen alternierenden Ereignisse ersetzt werden. Beim Duplizieren des ADD-Operators muss entschieden werden, wie der neue Operator zum vorausgehenden in Beziehung stehen soll: sequenziell, parallel/unabhängig oder alternativ. Dieses hängt davon ab, wie die neu hinzuzufügenden Aufgaben in Beziehung stehen. Hiermit wird eine Prozessstruktur während der Instanziierungszeit entstehen, die dann in eine ablauffähige EPK endet und schließlich instanziiert werden kann. Beim Erzeugen der Alternative muss dann jedoch noch eine Entscheidungsfunktion unmittelbar vor den Operator analog zur Transformation in Abbildung 3b) hinzugefügt werden. Des Weiteren muss es für den Fall, dass keine Funktionen in die bestehende Prozesskette eingefügt werden soll, möglich sein, den ADD-Operator zur Instanziierungszeit einfach zu löschen.

In Abbildung 4 ist ein Entstehungsprozess zu sehen. Ausgehend von einem ADD-Operator in a) sind bis d) drei neue Funktionen Nutzer-gesteuert hinzugefügt worden. Bei der Transformation von a) nach b) wurde der ADD-Operator unabhängig zum alten reproduziert, was mit einer vorausgehenden AND-Verknüpfung in der EPK ausgedrückt wurde. Die Beziehung der hinzuzufügenden Funktionen ist damit unabhängig und eine parallele Ausführung ist möglich. Bei der darauf folgenden Transformation nach c) hat dann gleich eine Ersetzung des linken ADD-Operators durch die Funktion „Heckklappe reparieren“ stattgefunden und zusätzlich wurde der rechte ADD-Operator in Sequenz reproduziert. In d) sind dann schließlich die übrigen ADD-Operatoren durch Funktionen ersetzt worden. Es zeigt sich dort, dass die hinzugefügten Funktionen eine Abhängigkeit besitzen, da ja zunächst die Ursache des Ölverlusts festgestellt werden muss, bevor diese behoben werden kann und daher die vorhergehend erzeugte Sequenz sinnvoll ist.

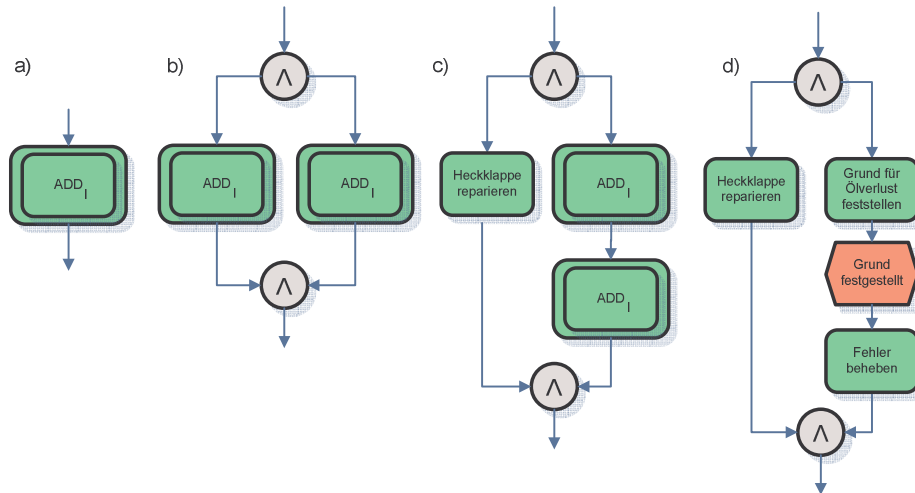


Abbildung 4: Entwicklung der Prozesskette durch den ADD-Operator zur Instanziierungszeit

3.3 Genauere Modellierung zur Instanziierungszeit

Die im Kapitel 3.2 vorgestellte Methode stellt schon viele Möglichkeiten bereit, während der Instanziierungszeit Teilprozesse individuell für die jeweilige Prozessinstanz hinzuzufügen. Es ist jedoch noch nicht möglich, die neu hinzugefügten Funktionen in die vorhandene Prozessdefinition einzuflechten.

Nehmen wir beispielsweise an, dass ein Prozess mit der in Abbildung 2a) vorgestellten EPK instanziiert werden soll und der Kunde wünscht während des Instanziierungsprozesses, insbesondere einen Ölwechsel für sein Auto zu bekommen. Zusätzlich hat dieser auch einen Ölverlust festgestellt, so dass dieser auch behoben werden soll, was dann mit dem in Kapitel 3.2 eingeführten ADD-Operator in die EPK hinzugefügt wird. Nun kann der Ölwechsel unabhängig bzw. parallel zur Reparatur der Ursache des Ölverlustes im Modell nach dem Prinzip „Flexibility by design“ eingefügt werden. Das WfMS wird in diesem Fall auch Geschäftsabläufe zulassen, die aus folgendem Grund nachteilig sind. Es kann dann nämlich der Ölwechsel vor der Reparatur der Ursache des Ölverlustes stattfinden, was zur Folge haben kann, dass das neu hinzugefügte Öl schon wieder verloren geht oder bei der Reparatur das neue Öl wieder abgelassen werden muss.

Beim obigen Geschäftsprozessbeispiel liegt es bei paralleler Modellierung der neu hinzugefügten Aufgabe in der Verantwortung des Mitarbeiters, die sinnvolle Durchführung des Prozesses zu wählen und zu erkennen, dass in dem konkreten Beispiel die Reparatur vor dem Ölwechsel zu vollziehen ist, obwohl das WfMS auch eine andere Ausführung zulässt. Eine genauere Modellierung der jeweiligen Prozessinstanz wird zwar bei korrekter Modellierung einen besseren Ablauf des Geschäftsprozesses garantieren, sie bedeutet jedoch einen bedeutend höheren Aufwand und birgt Risiken. Die editierte Workflow Definition kann Fehler enthalten, läuft dann zum ersten Mal und ist damit ungetestet, so dass im schlimmsten Fall ein Deadlock auftreten kann. Es stellt sich dadurch die Frage, inwiefern es Sinn macht, für jede Prozessinstanz den Prozess einzeln zu modellieren bzw. zu editieren.

Das genauere Modellieren des Geschäftsprozesses für die zu erzeugende Prozessinstanz kann jedenfalls durch zusätzliche Editierfunktionen der Prozessdefinition geschehen. Es können neu hinzugefügte Funktionen an andere Stellen in der Prozessdefinition verschoben werden, wo sie am sinnvollsten erscheinen. Das Verschieben muss Nutzer-gesteuert geschehen, da es in der Regel des Fachwissens bedarf, um die adaptiv neu hinzugefügten Funktionen in den bestehenden Workflow sinnvoll einzugliedern. Damit wird das Prinzip „Flexibility by change“ [AAH08, SMR07] realisiert.

Zusätzlich zum Verschieben sind noch weitere Operationen zur Veränderung bzw. Verfeinerung der Prozessdefinition denkbar. Beispielsweise können durch die Funktionsverfeinerung bei den EPKs neue Funktionen genauer spezifiziert werden. Es können Methoden aus dem Bereich des adaptiven Workflow Managements angewendet werden, in dem die Prozessdefinitionen nach [AH04] mit Hinzufügen, Ersetzen und Umstellen von Prozessen zur Laufzeit verändert werden. Weitere Änderungen sind außerdem noch bei den Change-Patterns ausgearbeitet [WR08]. Da die Mittel zur Laufzeit nach bestimmten Regeln angewendet werden dürfen, ist es natürlich insbesondere auch erlaubt, sie zum Anfang des Workflows, also zur hier untersuchten Instanzierungszeit, zu nutzen.

4 Zusammenfassung und Ausblick

Die hier vorgestellten Operatoren sollen die Prozessmodelle für die daraus zu bildenden Prozessinstanzen flexibler gestalten. Für den jeweiligen konkreten Geschäftsvorfall könnten dann z.B. Spezialwünsche des Kunden zur Instanzierungszeit in den Prozess aufgenommen oder spezielle Teile ausgeschlossen werden. Es wurden Entscheidungsoperatoren vorgestellt, die zur Instanzierungszeit für jeden neuen Geschäftsprozess entschieden werden müssen. Hierfür wurde die Idee der bereits vorhandenen Buildtime-Operatoren [Sch98] bzw. C-EPCs [RA05] verwendet. Um Prozesse des Weiteren um Funktionen zur Instanzierungszeit zu erweitern, wurde der ADD-Operator eingeführt. Die Möglichkeit des kompletten Editierens der Prozessdefinition für jede Prozessinstanz wurde anschließend in Abschnitt 3.3 diskutiert.

Für zukünftige Arbeiten ist die Umsetzung der neuen Operatoren in Verbindung mit einem Workflow Management System denkbar. Die EPKs müssen zunächst mit einem Modellierungstool, das die neuen Operatoren unterstützt definiert werden. Die Prozessmodelle können in einer erweiterten EPML-Codierung dem WfMS übergeben werden, das diese interpretiert. Dabei werden die Instantiationtime-Operatoren als erstes aufgelöst, indem die entsprechenden Entscheidungen bei den Instantiationtime-OR- und -XOR-Operatoren getroffen und ggf. neue Funktionen gemäß der Semantik des ADD-Operators hinzugefügt werden. Die daraus entstandenen Instanzen enthalten dann die gewünschten Funktionen, so dass die Prozessdefinition an den individuellen Anforderungen zur Instanzierungszeit angepasst wurde. Es wird an Werkzeugen zum Editieren und Anpassen von Workflowspezifikationen gearbeitet, in die die hier vorgestellten Vorschläge zu integrieren sind.

Literaturverzeichnis

- [AAH08] van der Aalst, W.; Adams, M.; ter Hofstede, A.H.M.; Pesic, M.; Schonenberg, H.: Flexibility as a Service, 2008; S. 4. <http://www.yawlfoundation.org/documents/BPM-08-09.pdf> (gelesen: 12.10.08)
- [AH04] van der Aalst, W.; van Hee, K.: Workflow Management. The MIT Press, Cambridge, 2004; S. 192 ff.
- [Ho95] Hollingsworth, D.: The Workflow Reference Model, WfMC-TC-1003, 1995. <http://www.wfmc.org/reference-model.html> (gelesen: 12.10.08)
- [KNS92] Keller, G.; Nüttgens, M.; Scheer, A.W.: Semantische Prozessmodellierung auf der Grundlage Ereignisgesteuerter Prozessketten (EPK). 1992. <http://www.iwi.uni-sb.de/Download/iwihefte/heft89.pdf> (gelesen: 12.10.08)
- [MN05] Mendling, J.; Nüttgens, M.: EPC Markup Language (EPML) – An XML-Based Interchange Format for Event-Driven Process Chains (EPC). 2005. <http://wi.wu-wien.ac.at/home/mendling/publications/TR05-EPML.pdf> (gelesen: 12.10.08)
- [MRR05] Mendling, J.; Recker, J.; Rosemann, M.; van der Aalst, W.: Towards the Interchange of Configurable EPCs: An XML-based Approach for Reference Model Configuration. 2005. <http://wi.wu-wien.ac.at/home/mendling/publications/05-EMISA.pdf> (gelesen: 12.10.08)
- [NR02] Nüttgens, M.; Rump, F. J.: Syntax und Semantik Ereignisgesteuerter Prozessketten. 2002. <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-65/06nuettgens.pdf> (gelesen: 12.10.08)
- [RA05] Rosemann, M.; van der Aalst, W.: A Configurable Reference Modelling Language, 2005. <http://is.tm.tue.nl/staff/wvdaalst/publications/p361.pdf> (gelesen: 12.10.08)
- [Sch98] Schütte, R.: Grundsätze ordnungsgemäßer Referenzmodellierung: Konstruktion konfigurations- und anpassungsorientierter Modelle. Gabler, Wiesbaden, 1998, Diss. Univ. Münster; S. 246 ff.
- [SMR07] Schonenberg, M.H.; Mans, R.S.; Russell, N.C.; van der Aalst, W.: Towards a Taxonomy of Process Flexibility, 2007. <http://out.uclv.edu.cu/cei/Daniel%20Rojas/Towards%20a%20Taxonomy%20of%20Process%20Flexibility.pdf> (gelesen: 12.10.08)
- [ST05] Scheer, A.-W.; Thomas, O.: Geschäftsprozessmodellierung mit der Ereignisgesteuerten Prozesskette. 2005. http://www.wiso.uni-hamburg.de/fileadmin/WISO_FS_WI/EPK-Community/Scheer_Thomas_2005_WISU_EPK.pdf (gelesen: 12.10.08)
- [WR08] Weber, B.; Reichert, M.; Rinderle-Ma, S.: Change Patterns and Change Support Features – Enhancing Flexibility in Process-Aware Information Systems, 2008. http://dbis.eprints.uni-ulm.de/335/1/DKE_WRR08.pdf (gelesen: 12.10.08)

Zur Weiterentwicklung der Spezifikation betrieblicher Softwarekomponenten

Jörg Ackermann

Lehrstuhl für Wirtschaftsinformatik und Systems Engineering
Universität Augsburg
Universitätsstraße 16, 86135 Augsburg
joerg.ackermann.hd@t-online.de

Zusammenfassung. Die präzise und eindeutige Spezifikation einer Softwarekomponente ist eine wesentliche Voraussetzung für ihre erfolgreiche Wiederverwendung. Dazu wurde 2002 vom Arbeitskreis WI-KobAS der GI ein Spezifikationsrahmen erstellt, wie die Spezifikation betrieblicher Softwarekomponenten erfolgen soll. Mittlerweile entsprechen die damaligen Empfehlungen allerdings nicht mehr in allen Punkten dem Stand der Forschung und dem Stand der Technik. Dieser Beitrag diskutiert bestehende Defizite und unterbreitet einen umfassenden Vorschlag, wie der Spezifikationsrahmen weiterentwickelt werden sollte.

1 Einleitung

Das grundlegende Paradigma komponentenorientierter Softwareentwicklung besteht darin, Anwendungen aus wiederverwendbaren Komponenten zu bauen – auf diese Weise wird die Erstellung von Komponenten (development for reuse) von der Erstellung ganzer Systeme aus Komponenten (development with reuse) entkoppelt [Br00, Sz02]. Dieser Ansatz verspricht u. a. kürzere Entwicklungszeiten, höhere Flexibilität und geringere Entwicklungskosten [DW98:397f., Gr98:583ff.].

Um erfolgreich wiederverwendet werden zu können, muss eine Komponente alle für ihren Einsatz relevanten Informationen in Form einer Komponentenspezifikation zur Verfügung stellen. Eine präzise und verlässliche Spezifikation unterstützt die Auswahl der passenden Komponente und das Vertrauen in die korrekte Arbeitsweise der Komponente [GG06:100]. Außerdem sind Komponentenspezifikationen Voraussetzung für den Erfolg von Komponentenmärkten [HT02] sowie für eine Kompositionsmethodologie und Werkzeugunterstützung [Ov04:4]. Aus diesen Gründen ist die Spezifikation von Softwarekomponenten ein kritischer Erfolgsfaktor bei der komponentenorientierten Entwicklung betrieblicher Anwendungssysteme.

Für betriebliche Softwarekomponenten steht mit dem Memorandum „Vereinheitlichte Spezifikation von Fachkomponenten“ [Tu02] ein Spezifikationsrahmen zur Verfügung, dessen Ziel in der Schaffung eines methodischen Standards für die Spezifikation besteht.

Mittlerweile entsprechen die Empfehlungen des Memorandums allerdings nicht mehr in allen Punkten dem Stand der Forschung und dem Stand der Technik.

Deshalb wird vorgeschlagen, dass der Arbeitskreis WI-KobAS eine überarbeitete Version 2.0 des Memorandums erstellt. Dieser Beitrag zeigt dazu auf, an welchen Stellen Handlungsbedarf besteht, und unterbreitet Vorschläge, wie sich die Defizite beseitigen lassen. Dazu wird zum einen auf eigene Forschungsergebnisse zurückgegriffen und zum anderen beschrieben, wie sich Ergebnisse anderer Autoren (vor allem [Ov06]) in das Memorandum (in seiner bisherigen Struktur) integrieren lassen. Die Arbeit ist wie folgt aufgebaut: Zunächst erfolgt eine Einführung zur Spezifikation betrieblicher Softwarekomponenten (Kapitel 2) sowie die Vorstellung einer Beispielkomponente (Kapitel 3). Danach erfolgen Vorschläge zu allgemeinen Aspekten der Spezifikation (z. B. Sprachauswahl und Erweiterbarkeit), die alle Ebenen betreffen (Kapitel 4). Anschließend werden für die sieben Spezifikationsebenen des Memorandums bestehende Defizite benannt und Lösungsvorschläge unterbreitet (Kapitel 5). Die Arbeit schließt mit dem zusammenfassenden Kapitel 6. Die Bedeutung der vorliegenden Arbeit besteht darin, dass ein umfassender Vorschlag unterbreitet wird, wie der Spezifikationsrahmen [Tu02] weiterentwickelt werden kann.

2 Spezifikation betrieblicher Softwarekomponenten

Unter *Spezifikation einer Softwarekomponente* wird die vollständige, widerspruchsfreie und eindeutige Beschreibung der Außensicht einer Komponente verstanden – die Spezifikation zeigt auf, welche Dienste eine Komponente in welchem Bedingungsrahmen bereitstellt [Tu02:3].

Trotz der großen Bedeutung von Komponentenspezifikationen gibt es zurzeit keinen Spezifikationsstandard, der allgemein anerkannt ist und alle relevanten Sachverhalte abdeckt. So konzentrieren sich viele der bestehenden Ansätze auf einzelne Entwicklungsphasen wie z. B. Entwurf und Entwicklung [DW98, CD01], Komposition [YS97] oder Auswahl [HL01]. Außerdem streben viele dieser Arbeiten eine Unterstützung von Entwicklung, Anpassung und Komposition aus technischer Sicht an – Suche und Auswahl von Komponenten (insbesondere aus fachlicher Sicht) werden dagegen nur unzureichend unterstützt [GG06]. Bisher existieren nur wenige Ansätze, deren Ziel die umfassende und vollständige Spezifikation von Softwarekomponenten ist – zu nennen sind dabei vor allem [Be99, Tu02, Ov06].

Die vorliegende Arbeit bezieht sich auf den Spezifikationsrahmen „Vereinheitlichte Spezifikation von Fachkomponenten“ (Memorandum) [Tu02]. Der Inhalt des Memorandums ist eine Empfehlung des Arbeitskreises „Komponentenorientierte betriebliche Anwendungssysteme“ (WI-KobAS) der Gesellschaft für Informatik und wurde gemeinsam von Vertretern aus Forschung und Praxis erstellt. Der Spezifikationsrahmen strebt an, einen *methodischen* Standard für die Spezifikation von betrieblichen Fachkomponenten zu schaffen. Dazu wurden die zu spezifizierenden Objekte ermittelt und verschiedenen Beschreibungsebenen zugeordnet und es wurden die auf den einzelnen Ebenen einzusetzenden Notationen festgeschrieben. [Tu02] sieht

insgesamt sieben Spezifikationsebenen vor: Syntax-, Verhaltens-, Abstimmungs-, Qualitäts-, Terminologie-, Aufgaben und Vermarktungsebene. In Kapitel 5 findet sich eine kurze Einführung zu Inhalt und Notation der einzelnen Ebenen.

Das Memorandum [Tu02] hebt sich von früheren Spezifikationsansätzen vor allem dadurch ab, dass es die Spezifikation aller relevanten Aspekte in einem vereinheitlichten Ansatz anstrebt, dazu sowohl technische als auch fachliche Aspekte betrachtet und berücksichtigt, dass die unterstützten betrieblichen Aufgaben sowie die verwendeten betrieblichen Begriffe auf fachlicher Ebene zu spezifizieren sind. Nachteile des Memorandums bestehen darin, dass die Spezifikationsvorschriften auf der Qualitätsebene unpräzise bleiben, dass die zu verwendenden Satzbaupläne auf Terminologie- und Aufgabenebene nicht verbindlich vorgegeben werden und dass der Spezifikationsrahmen von einem eingeschränkten Komponentenmodell (keine Mehrfachschnittstellen, keine benötigten Schnittstellen) ausgeht.

Zu erwähnen bleibt, dass auf den Ideen von [Tu02] aufbauend eine Weiterentwicklung der Komponentenspezifikation durch Arbeiten von Overhage erfolgte, in deren Ergebnis der Spezifikationsrahmen UnSCom [Ov06] entstand. Die vorliegende Arbeit unterscheidet sich von [Ov06] dadurch, dass hier Vorschläge zur Weiterentwicklung unterbreitet werden, die Struktur und Aufbau des Memorandums (weitgehend) erhalten, während sich [Ov06] bzgl. Spezifikationsebenen und Notationstechniken zum Teil deutlich von [Tu02] unterscheidet.

3 Beispielkomponente *Lagerverwaltung*

Zur Illustration der weiteren Ausführungen wird in diesem Abschnitt eine Beispielkomponente *Lagerverwaltung* eingeführt. Die Komponente unterstützt die Abwicklung einer sehr einfachen Lagerhaltung – zur Vereinfachung wurden Funktionalität und Komplexität auf das Nötigste reduziert. Die Komponente ist komplex genug, um als Beispiel in dieser Arbeit zu dienen, aber in dieser Form kaum praxistauglich – so unterstützt die Komponente z. B. nur die manuelle Einlagerung.

Die Komponente *Lagerverwaltung* unterstützt die Lagerabwicklung in genau einem Lager. Die von der Komponente verwalteten Daten werden in Abb. 1 durch ein Datenmodell dargestellt. Das Lager besteht aus einer Anzahl von *Lagerplätzen*, auf denen die Materialien physisch gelagert werden. *Lagerbestand* repräsentiert die an einem Lagerplatz vorhandene Menge eines Lagermaterials (beispielsweise drei Paletten von Lagermaterial ABC-XYZ). *Lagermaterial* enthält die lagerspezifischen Eigenschaften eines Materials. Wird auf einem Lagerplatz ausschließlich ein bestimmtes Lagermaterial gelagert, kann man dem Lagerplatz dieses Lagermaterial zuweisen. Im Beispiel werden keine Lagereinheitentypen verwendet und es wird stattdessen angenommen, dass innerhalb des Lagers jeder Lagerplatz für ein Material geeignet ist.

Die Beispielkomponente unterstützt die Ausführung folgender zentraler Lageraktivitäten: Lagermaterial einlagern, Lagermaterial auslagern, Lagerbestand abfragen (Schnittstelle *IStockManagement*). Darüber hinaus existieren Funktionen, mit

denen sich Lagerplätze und Lagermaterialien verwalten lassen (Schnittstellen *IStorageBin* und *IWarehouseMaterial*). Außerdem ermöglicht die Komponente, bei einer (außerhalb der Komponente liegenden) Materialverwaltungskomponente die Existenz eines Lagermaterials abzufragen (benötigte Schnittstelle *IMaterial*).

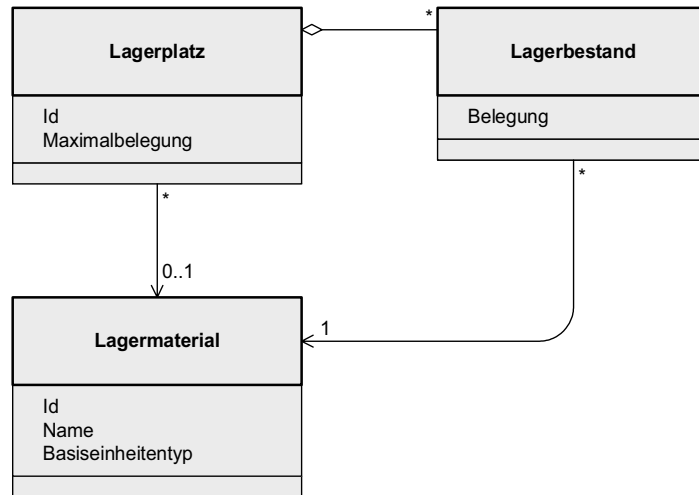


Abb. 1: Von der Komponente *Lagerverwaltung* verwaltete Daten

4 Ebenenübergreifende Aspekte

Dieses Kapitel fasst offene Punkte und Vorschläge zusammen, die sich auf den Spezifikationsrahmen [Tu02] im Ganzen beziehen und sich damit keiner der Spezifikationsebenen zuordnen lassen – für die einzelnen Ebenen selbst siehe Kapitel 5.

4.1 Aufbau und Spezifikationsebenen

Der Aufbau des Spezifikationsrahmens an sich und insbesondere die einzusetzenden Spezifikationsebenen bedürfen einer nochmaligen gründlichen Überprüfung. So wird z. B. in [Ov06:67] kritisiert, dass der Identifikation der Ebenen in [Tu02] nur ein empirisches Vorgehen zugrunde lag, dessen Ergebnis sich nicht in intersubjektiv nachvollziehbarer Weise begründen lässt. Im Gegensatz dazu werden die Spezifikationsebenen von UnSCom aus einem Klassifikationsschema abgeleitet, welches nach Modellperspektiven (statisch, operational, dynamisch) und Abstraktionsebenen (fachlich, logisch, physisch) unterscheidet [Ov06:125ff.]. Ein Nachteil der UnSCom-Lösung liegt jedoch darin, dass drei Spezifikationsebenen für die physische Komponentenbeschreibung entstehen, obwohl (aufgrund des Black-Box-Charakters) nur vergleichsweise wenige Informationen zu erfassen sind – die Beschreibung der nicht-funktionalen Eigenschaften wirkt damit unangemessen aufgebläht und könnte stattdessen auch auf einer zusammengefassten Ebene (= Qualitätsebene in [Tu02]) erfolgen.

4.2 Explizite Benennung der spezifikationsrelevanten Sachverhalte

Ein Defizit von [Tu02] ist, dass die spezifikationsrelevanten Sachverhalte nicht genau identifiziert werden. Eine explizite und vollständige Benennung aller spezifikationsrelevanten Sachverhalte ist jedoch sowohl für einen Spezifikationsersteller (welche Aspekte sind in der Spezifikation zu beschreiben und welche nicht?) als auch für den Hersteller eines Spezifikationstools (welche Aspekte muss man erfassen können?) unverzichtbar.

Daher wird vorgeschlagen, zu jeder Spezifikationsebene alle spezifikationsrelevanten Sachverhalte explizit aufzuführen. Als *spezifikationsrelevanter Sachverhalt* wird dabei alles bezeichnet, was die Außensicht der Komponente betrifft. Dies können Dinge (z. B. Attribute), deren Eigenschaften (z. B. Wertebereich) und zugehörige Ausprägungen (z. B. Festwerte *A* und *B*) sein. Um die Breite des Spektrums auszudrücken, wurde der allgemeine Begriff *Sachverhalt* gewählt. Beispiele solcher Sachverhalte sind Name und Dienste einer Schnittstelle oder Name und Wertebereich eines Qualitätsmerkmals.

Objekte	Spezifikationsrelevante Sachverhalte
Komponente	Name, Zugeordnete Schnittstellen (angeboten/benötigt)
Schnittstelle	Name, Dienste (Name, Parameter)
Datentypen	Name, Attribute (Name, Datentyp, Kardinalität) bzw. Aufzählungswerte
Typ im Spezifikationsdatenmodell	Name, Attribute (Name, Datentyp, Kardinalität)
Beziehung im Spezifikationsdatenmodell	Beziehungsart, Beteiligte Typen, Kardinalitäten, Rollen

Abb. 2: Spezifikationsrelevante Sachverhalte auf der Schnittstellenebene

Die spezifikationsrelevanten Sachverhalte lassen sich entweder tabellarisch erfassen [Ac07a:97f.] oder in Form von Metaschemata abbilden [Ov06:133]. Abb. 2 zeigt exemplarisch, wie eine tabellarische Darstellung der spezifikationsrelevanten Sachverhalte für die Schnittstellenebene aussieht. Um die Übersichtlichkeit der Einträge zu erhöhen, werden die Sachverhalte nach Bezugsobjekten geordnet dargestellt [Ac04:141]. Motivation dafür ist, dass spezifikationsrelevante Sachverhalte nicht isoliert existieren, sondern meist zu einem übergeordneten Bezugsobjekt gehören.

(*Qualitätsmerkmal* ist ein Beispiel für ein Bezugsobjekt, welches u. a. die spezifikationsrelevanten Sachverhalte *Name* und *Wertebereich* umfasst.)

In [Ac07a:98-123] findet sich die vollständige Auflistung der spezifikationsrelevanten Sachverhalte aller Spezifikationsebenen, auf deren Darstellung in dieser Arbeit aus Platzgründen verzichtet werden muss.

4.3 Ablage ebenenübergreifender Informationen

Das Memorandum schlägt vor, die Beschreibung von Komponenten auf verschiedenen Spezifikationsebenen vorzunehmen [Tu02:3]. Durch eine solche Aufteilung der Spezifikationsinhalte auf verschiedene thematische Beschreibungsebenen wird die Komplexität reduziert – Spezifikationen werden damit für den Produzenten einfacher zu erstellen und für den Konsumenten leichter zu verstehen.

Allerdings hat sich herausgestellt, dass eine Spezifikation auch Informationen enthalten muss, die Sachverhalte verschiedener Ebenen betreffen. Beispiele dafür sind die Zuordnung von Sachverhalten aus fachlicher und technischer Sicht [Tu02:5] oder die ebenenübergreifende Zusammenfassung von Parametrisierungsauswirkungen [Ac07a:227ff.]. Solche ebenenübergreifenden Informationen ließen sich in der bisherigen Struktur des Memorandums nur unzureichend abbilden – so wurden z. B. die Zuordnungstabellen für die fachliche und technische Sicht einfach der Schnittstellenebene zugeordnet [Tu02:5]. Die Zuweisung solcher übergreifenden Informationen zu einer einzelnen Spezifikationsebene ist jedoch dann unangemessen, wenn die Ebenenauswahl willkürlich ist und die Informationen von Charakter und Notationstechnik nicht zu den übrigen Informationen der Ebene passen.

Aus diesem Grund wird vorgeschlagen, dass der Spezifikationsrahmen [Tu02] um einen weiteren Bereich für ebenenübergreifende Informationen erweitert wird. Dieser Bereich kann bei Bedarf für unterschiedliche Aspekte nochmal unterteilt werden. Wichtig ist dabei, dass bei jedem Kandidaten für den ebenenübergreifenden Bereich untersucht wird, ob die Informationen tatsächlich ebenenübergreifend sind (wie bei einer Zuordnung von Sachverhalten verschiedener Ebenen) oder ob ein falscher Zuschnitt der Ebenen die Ursache dafür ist, dass sich die betroffenen Informationen nicht einer Spezifikationsebene zuordnen lassen.

4.4 Mehrsprachige Komponentenspezifikationen

Eine Komponentenspezifikation ist eine formalisierte Dokumentation einer Softwarekomponente. Analog zu anderen Dokumentationen und Produkthandbüchern ist davon auszugehen, dass Komponentenanwender eine Spezifikation in ihrer Landessprache bevorzugen. Damit stellt sich die Frage nach der Mehrsprachigkeit von Komponentenspezifikationen, wenn eine Komponente in verschiedenen Ländern vertrieben wird. Damit muss auch ein Spezifikationsrahmen und darauf aufbauende Spezifikationstools die Möglichkeit vorsehen, eine Spezifikation in mehreren Sprachen

zu erstellen – diese Problematik wurde von den gängigen Spezifikationsansätzen [DW98, CD01, Tu02, Ov06] bisher nicht diskutiert.

Bei einer Analyse der Sprachabhängigkeit der Spezifikationsartefakte [Ac07a:95f.] ergibt sich, dass manche Artefakte einsprachig sein müssen (Namen von Schnittstellen und Datentypen), andere mehrsprachig sein könnten (Begriffe und Aufgaben) sowie die Sprachkonstrukte von OCL und QML einsprachig auf Englisch sind.

Im Spezifikationsrahmen [Tu02] sind die Spezifikationsbeispiele in der deutschen Version komplett auf Deutsch und wurden für die englische Version komplett ins Englische übersetzt – damit wird als Lösung des Sprachproblems suggeriert, eine Spezifikation einfach komplett in andere Sprachen zu übersetzen. Diese auf den ersten Blick naheliegende Lösung ist jedoch nicht möglich, da Schnittstellen- und Datentypnamen (beim derzeitigen Stand der Technik) eindeutig und damit einsprachig sein müssen!

Stattdessen wird nachfolgend vorgeschlagen, wie durch eine geeignete Sprachauswahl das Ziel einer verständlichen und konsistenten Spezifikation erreicht werden kann:

- Auf den technischen Ebenen (Schnittstellen, Verhalten, Abstimmung, Qualität) besteht das Problem, dass Teile der Spezifikation einsprachig sein müssen (Schnittstellen), andere mehrsprachig sein können (Datenmodell) und andere fest in Englisch sind (z. B. OCL). Da eine Mischung von Sprachen die Verständlichkeit der Spezifikation beeinträchtigt, wird vorgeschlagen, die Spezifikation auf diesen Ebenen einsprachig in Englisch zu erstellen. Dieser Vorschlag ist angemessen, da bei Technikexperten (Zielgruppe dieser Ebenen) englische Sprachkenntnisse typischerweise vorhanden sind und technische Spezifikationen sowieso häufig in Englisch erstellt werden.
- Für die fachlichen Ebenen (Terminologie, Aufgaben) und die Vermarktungsebene wird vorgeschlagen, die Spezifikation in allen benötigten Sprachen zur Verfügung zu stellen. Dies begründet sich dadurch, dass zum einen ausreichende englische Sprachkenntnisse bei Fachexperten (Zielgruppe dieser Ebenen) nicht vorausgesetzt werden können und dass zum anderen die vollständige Übersetzung aller Spezifikationsinhalte dieser Ebenen technisch möglich ist. Diese Mehrsprachigkeit muss vom Spezifikationsrahmen (sprachabhängige Satzbaupläne) und von darauf aufbauenden Tools (sprachabhängige Spezifikationsinhalte) explizit vorgesehen werden.

Diesen Richtlinien folgend sind in dieser Arbeit alle Spezifikationsbeispiele auf den technischen Ebenen in Englisch und auf den fachlichen Ebenen in Deutsch.

4.5 Erweiterbarkeit des Spezifikationsrahmens

Aus dem Ziel des Spezifikationsrahmens [Tu02], einen Standard zur Spezifikation von betrieblichen Fachkomponenten zu schaffen, ergibt sich, dass dieser Spezifikationsvorschriften für alle spezifikationsrelevanten Sachverhalte enthalten muss.

Daher sind auch spezielle Aspekte zu berücksichtigen (wie z. B. die Spezifikation von parametrisierbaren Komponenten [Ac07a] oder von Komponenten mit graphischen Benutzeroberflächen [Ov06:275]), die zwar für manche, aber nicht für alle Komponenten relevant sind. Werden alle diese Vorschriften in [Tu02] direkt aufgenommen, ist zu befürchten, dass der Spezifikationsrahmen bald unübersichtlich wird und damit an Akzeptanz verliert.

Aus diesem Grund wird vorgeschlagen, bei der Weiterentwicklung von [Tu02] zwischen einem Kernstandard und ergänzenden Zusatzstandards zu unterscheiden. Sinnvoll wäre in diesem Zusammenhang, wenn schon der Kernstandard Möglichkeiten zur Erweiterung vorsieht, so dass zusätzliche Aspekte einfach ergänzt werden können, ohne jeweils den Kernstandard erweitern zu müssen. So sollte z. B. der Kernstandard erlauben, die verwendeten UML-Elemente zu stereotypisieren oder zusätzliche Satzbaupläne aufzunehmen.

5 Ebenenspezifische Weiterentwicklungsvorschläge

In diesem Kapitel werden Vorschläge zur Weiterentwicklung des Memorandums unterbreitet, die sich konkret auf die einzelnen Spezifikationsebenen beziehen. Dabei orientiert sich die Darstellung an den derzeit bestehenden Ebenen von [Tu02] – ggf. werden sich diese im Rahmen der Weiterentwicklung ändern (vgl. dazu die Diskussion in Abschnitt 4.1).

5.1 Schnittstellenebene

Die Schnittstellenebene dient dazu, grundlegende Vereinbarungen zur technischen Kommunikation zu treffen [Tu02:5]. Dazu gehören die Namen der Dienste, die eine Komponente anbietet oder benötigt, die Definition der verwendeten Datentypen, die Signaturen der Dienste sowie die Deklaration von Fehlermeldungen. Die daraus resultierenden Vereinbarungen garantieren, dass Dienstnehmer und Dienstgeber auf technischer Ebene miteinander kommunizieren können – semantische Aspekte bleiben dagegen weitgehend unberücksichtigt.

Die im Memorandum [Tu02] vorgesehenen Spezifikationsvorschriften für die Schnittstellenebene weisen allerdings einige eklatante Einschränkungen und Probleme auf: es wird von einem eingeschränkten Komponentenmodell mit nur einer angebotenen Schnittstelle ausgegangen, der Name der Komponente kann nicht explizit definiert werden (sondern ergibt sich aus dem Namen der Schnittstelle), benötigte Dienste werden durch einen Workaround mithilfe einer Schnittstelle *Extern* deklariert und durch die vorgeschlagene Verwendung der OMG IDL als Notationstechnik kommt es zu einem Methodenbruch bei der Verhaltensebene (in UML OCL formulierte Constraints erfordern ein UML-Modell – der Bezug auf OMG IDL-Ausdrücke ist syntaktisch falsch [Ac03:26f.]).

Die bei der Erstellung von [Tu02] gültige Version 1.4 der UML enthielt zwar das Modellierungskonstrukt *Komponente*, war jedoch ausschließlich für die Modellierung von physischen Komponenten vorgesehen. Bei einer Spezifikation sind jedoch logische Komponenten einzusetzen [Ac03:24], die von UML 1.4 nicht explizit unterstützt wurden. Diese Einschränkung entfällt mit der aktuellen Version UML 2.0 [OMG05b] – damit ist nun möglich, das Konstrukt *Komponente* zur Modellierung logischer Komponenten zu verwenden.

Zur Weiterentwicklung des Spezifikationsrahmens [Tu02] wird vorgeschlagen, auf der Schnittstellenebene die UML 2.0 als primäre Notationstechnik einzusetzen:

- Zur Spezifikation einer Komponente wird ein Komponentendiagramm eingesetzt, aus welchem der Name der Komponente sowie die Schnittstellen der Komponente hervorgehen – vgl. dazu Abb. 3. Die UML erlaubt die Angabe mehrere Schnittstellen pro Komponente und (anhand unterschiedlicher Notationen) eine Unterscheidung zwischen angebotenen (z. B. *IStockManagement*) und benötigten Schnittstellen (*IMaterial*).



Abb. 3: Beispielkomponente *Lagerverwaltung* mit Schnittstellen

- Die genaue Spezifikation der Schnittstellendienste erfolgt mit Hilfe des UML-Konstrukts *Schnittstelle* (engl. *interface*) und wird für die Schnittstelle *IStockManagement* beispielhaft in Abb. 4 dargestellt. Dabei werden zu jedem Dienst die Parameter (mit Parameterrichtung (in, out, inout), Name, Datentyp und Kardinalität), ein optionaler Returnparameter (mit Datentyp) und eventuell eingesetzte Ausnahmen (engl. *exceptions*) spezifiziert.

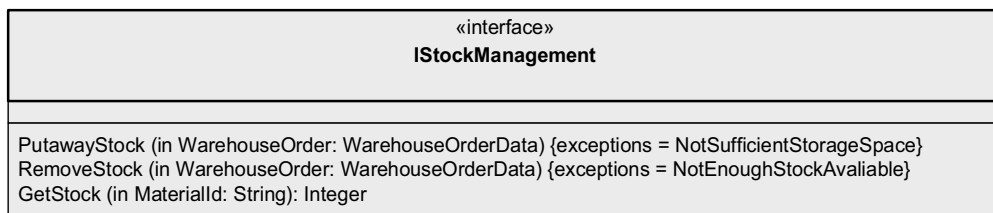


Abb. 4: Detaillierte Spezifikation der Schnittstelle *IStockManagement*

- Werden neben UML-Standarddatentypen (wie *String* oder *Integer*) auch eigene Datentypen und Ausnahmetypen verwendet, so müssen diese ebenfalls mit Hilfe der UML spezifiziert werden. Dazu können die UML-Modellierungskonstrukte *dataType* und *exception* eingesetzt werden.

Durch den Einsatz der UML in der angegebenen Form ergeben sich folgende Vorteile für die Spezifikation auf der Schnittstellenebene: 1. Schnittstellen-, Verhaltens- und Abstimmungsebene verwenden mit der UML eine einheitliche Notationstechnik und es kann auf eine Notationstechnik (OMG IDL) verzichtet werden. (Bei Bedarf lässt sich eine IDL-Spezifikation aus der UML-Beschreibung ableiten.) 2. Der bisher vorkommende Methodenbruch zwischen Schnittstellen- und Verhaltensebene sowie die anderen oben aufgeführten Einschränkungen werden vermieden. 3. Der Spezifikationsrahmen bleibt konform zur aktuellen UML-Version und die Spezifikation profitiert von den erweiterten Ausdrucksmöglichkeiten von UML 2.0. Damit wird also vorgeschlagen, die UML konsequent auf den Ebenen einzusetzen, wo sie die Spezifikation verbessern kann und wo sie international als Modellierungsstandard anerkannt ist. Betriebliche Fachkomponenten lassen sich jedoch nicht vollständig mit der UML spezifizieren, da auch spezifikationsrelevante Objekte und Spezifikationsebenen (z. B. Terminologie und betriebliche Aufgaben) zu betrachten sind, die von der UML nicht abgedeckt werden.

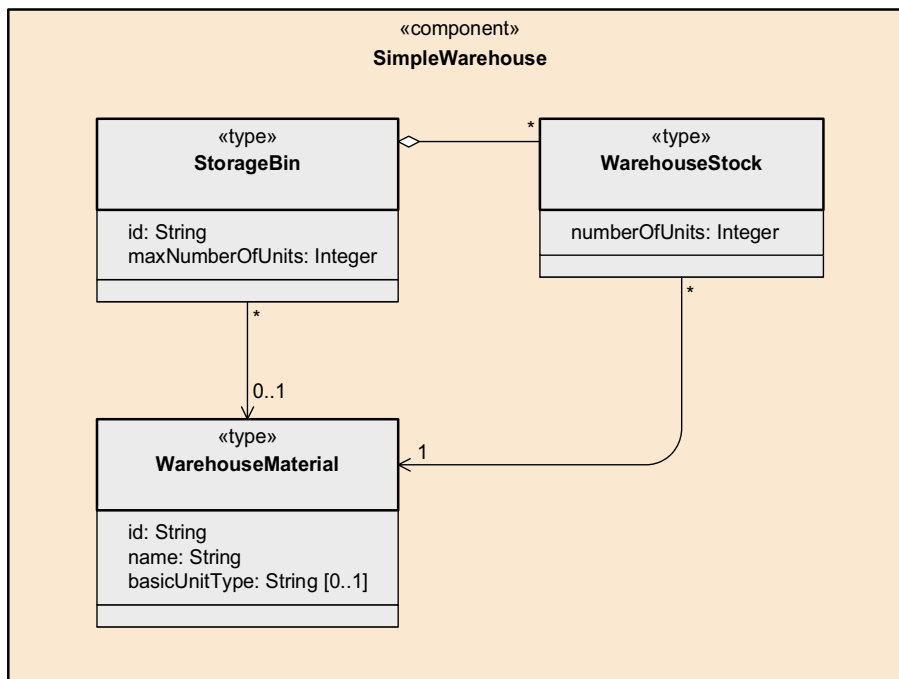


Abb. 5: Spezifikationsdatenmodell der Beispielkomponente *Lagerverwaltung*

Betriebliche Fachkomponenten verwalten häufig betriebliche Daten – so erlaubt die Komponente *Lagerverwaltung* beispielsweise Lagerplätze anzulegen, zu speichern und zu einem späteren Zeitpunkt wieder anzuzeigen oder zu verändern. Um solches Verhalten vollständig und syntaktisch korrekt spezifizieren zu können, muss die Spezifikation mit einem geeigneten Datenmodell versehen werden. Zu diesem Zweck wurde das Konzept eines *Spezifikationsdatenmodells* eingeführt. (Das Modell modelliert nur Daten und keine Operationen und ist daher kein Objektmodell.) Ein solches Modell

repräsentiert die von einer Komponente verwalteten Daten auf konzeptioneller Ebene – es enthält dazu die vertragsrelevanten Daten und abstrahiert von den internen Implementierungsdetails. Die Darstellung dieses Modells erfolgt mithilfe des UML-Konstrukts *component* und wird in Abb. 5 beispielhaft für die Komponente *Lagerverwaltung* gezeigt. In [Ac07b] findet sich eine ausführliche Darstellung des Konzepts inklusive der Begründung seiner Notwendigkeit und einer Diskussion, welche Daten in einem solchen Modell abzubilden sind und welche nicht.

Es wird vorgeschlagen, ein solches (optionales) Spezifikationsdatenmodell der Schnittstellenebene zuzuordnen. Für diese Zuordnung spricht, dass das Modell die vertragsrelevanten Daten der Komponente deklariert und damit gut zu den anderen Deklarationen auf der Schnittstellenebene passt, dass die eingesetzten Notationen (UML-Modelle) übereinstimmen und dass das Modell sowohl auf der Verhaltens- als auch auf der Abstimmungsebene benötigt wird und sich daher keiner der beiden Ebenen eindeutig zuordnen lässt.

5.2 Verhaltensebene

Aufgabe der Verhaltensebene ist, das Verhalten einer Komponente näher zu beschreiben – dazu wird spezifiziert, wie sich die Komponente im Allgemeinen und insbesondere bei Grenzfällen verhält [Tu02:7]. So kann beispielsweise angegeben werden, dass die Summe aller Bestände auf einem Lagerplatz die für den Lagerplatz festgeschriebene Maximalbelegung nicht überschreiten darf oder dass die Einlagerung von Lagermaterialien den Bestand auf dem verwendeten Lagerplatz entsprechend erhöht. Wichtig ist dabei, dass das Verhalten der Komponente vollständig zu beschreiben ist.

Als primäre Notationstechnik wird auf der Verhaltensebene die *UML Object Constraint Language* (OCL) [OMG05a] eingesetzt. Verhaltensbeschreibungen mithilfe der OCL erfolgen durch die Angabe von allgemeinen Invarianten sowie von dienstspezifischen Vor- und Nachbedingungen.

In dieser Arbeit wird weitgehend den vom Memorandum vorgegebenen Spezifikationsvorschriften gefolgt [Tu02:7-9]. Es ergeben sich jedoch folgende Detailverbesserungen:

- Das neu eingeführte Spezifikationsdatenmodell ermöglicht, Invarianten und mit persistierten Daten zusammenhängende Vor- und Nachbedingungen syntaktisch korrekt zu formulieren.
- Es vereinfacht sich die Verhaltensspezifikation der benötigten Dienste, da diese direkt der Komponente zugeordnet sind und der Workaround über eine gedachte Komponente *Extern* entfällt.
- Alle in OCL formal spezifizierten Bedingungen sind zusätzlich (im Sinne einer sekundären Notationstechnik) natürlichsprachlich anzugeben – es wird vorgeschlagen, dafür die von der OCL vorgesehenen Kommentare zu verwenden.

```

context StorageBin
  inv: self.maxNumberOfUnits > 0
      -- attribute maxNumberOfUnits must be greater than 0

  inv: self.WarehouseStock.numberofUnits->sum() <= self.maxNumberOfUnits
      -- sum of all stock can not exceed attribute maxNumberOfUnits

```

Abb. 6: Invarianten für den Typ *StorageBin*

Abb. 6 zeigt exemplarisch zwei Invarianten. Diese legen fest, dass die Maximalbelegung eines Lagerplatzes immer größer als null sein muss und dass die Summe aller auf einem Lagerplatz gelagerten Bestände nicht größer sein kann als die für den Lagerplatz vorgesehene Maximalbelegung.

5.3 Abstimmungsebene

Vereinbarungen auf der Abstimmungsebene regeln die Reihenfolge, in der Dienste aufgerufen werden können, sowie die Synchronisationserfordernisse zwischen Diensten [Tu02:10]. Beispielsweise wird hier festgelegt, dass ein Lagermaterial zunächst definiert werden muss, bevor eine Einlagerung für dieses Lagermaterial durchführbar ist.

Mögliche Notationsalternativen auf dieser Ebene sind die um temporale Operatoren erweiterte OCL [CT01] (vorgeschlagen in [Tu02:10]) sowie Zustandsautomatenmodelle (z. B. vorgeschlagen in [Ov06:189ff.]). Für die Verwendung von Automaten spricht, dass diese (z. B. im Rahmen von Kompatibilitätstests oder Qualitätsvorhersagen) statisch auswertbar sind. Der Nachteil von Automaten ist jedoch, dass diese alle erlaubten Aufruffreihenfolgen explizit spezifizieren – eine vollständige Erfassung ist jedoch nicht immer praktikabel. In einem solchen Fall haben temporale Operatoren einen Vorteil, da sie lediglich Einschränkungen bei der Reihenfolge von Dienstaufrufen festlegen. Ein Nachteil der temporalen OCL in der Form von [CT01] liegt darin, dass die formalen Grundlagen der Erweiterung nicht expliziert werden [ZG03:351].

Daher wird für die Abstimmungsebene vorgeschlagen, noch einmal die genaue Zielstellung der Spezifikation festzulegen und darauf aufbauend die Wahl der geeigneten Notationstechnik zu treffen.

5.4 Qualitätsebene

Ziel der Qualitätsebene ist die Spezifikation der nicht-funktionalen Eigenschaften einer Komponente – Beispiele dafür sind die Verfügbarkeit, das Performanzverhalten und die Wartbarkeit eines durch die Komponente angebotenen Dienstes [Tu02:12]. Dazu muss die Qualitätsebene zu beschreibende Qualitätskriterien, verwendbare Messgrößen und Methoden zu deren Quantifizierung sowie ggf. durch die Komponenten einzuhaltende Service Level (Qualitätsgrenzen zur Laufzeit) enthalten. Darüber hinaus ist festzulegen,

in welcher Form diese Informationen dem Benutzer einer Komponente zur Verfügung gestellt werden.

Der Spezifikationsrahmen [Tu02] erkennt zwar die Notwendigkeit der Spezifikation nicht-funktionaler Eigenschaften, enthält jedoch nur ein eher allgemeines Vorgehensmodell, aber kein Qualitätsmodell und keine konkreten Spezifikationsvorschriften.

Deutlich konkretere Vorgaben zur Spezifikation nicht-funktionaler Eigenschaften sind im Spezifikationsrahmen UnSCom enthalten. Besonders hervorzuheben sind dabei folgende Ergebnisse [Ov06:212-236]:

- Es wird ein abstraktes Qualitätsmodell entwickelt, welches alle spezifikationsrelevanten Sachverhalte strukturiert in Form eines Metamodells darstellt. Dabei muss eine Komponentenspezifikation sowohl die Definition der verwendeten Qualitätskategorien und -merkmale (auf Typebene) als auch die für die Komponente konkreten Kennzahlen (Ausprägungen) der Merkmale enthalten.
- Es werden sechs Qualitätskategorien (*Verwendbarkeit*, *Wartbarkeit*, *Portabilität*, *Funktionalität*, *Zuverlässigkeit* und *Effizienz*), insgesamt 49 Qualitätsmerkmale sowie geeignete Messverfahren vorgegeben, welche bei der Spezifikation einer Softwarekomponente einsetzbar sind.
- Es wird vorgeschlagen, die Spezifikation mithilfe der *Quality of Service Modeling Language (QML)* [FK98] durchzuführen. Erwähnenswert ist dabei, dass die QML sowohl für die Definition der Kategorien und Merkmale als auch für die Angabe der Kennzahlen einsetzbar ist.

Abb. 7 zeigt beispielhaft eine QML-Spezifikation für die Qualitätskategorie *Effizienz* (*efficiency*). Im oberen Teil werden z. B. für das Merkmal *Antwortzeit* (*responseTime*) die Eigenschaften rationaler Wertebereich, Maßeinheit *Sekunden* und die Präferenz für kleinere Werte festgelegt. Danach wird (für die Beispielkomponente) z. B. festgelegt, dass die Antwortzeit des Dienstes *GetStock* im Mittel kleiner als eine Sekunde ist.

```
type Efficiency = contract {
  responseTime : decreasing numeric s;
  throughput : increasing numeric calls/s; ... };

StockManagement for IStockManagement = profile {
  from GetStock require Efficiency = contract {
    responseTime {mean < 1 s;};
    throughput {mean > 25 calls/s; percentile 80 > 18 calls/s;}; }; }
```

Abb. 7: Beispielhafte Spezifikation für die Qualitätskategorie *Effizienz*

Es wird vorgeschlagen, die Ergebnisse aus [Ov06] auf den Spezifikationsrahmen [Tu02] zu übertragen. Zu beachten ist dabei, dass die in [Ov06] auf mehrere Spezifikationsebenen verteilten Sachverhalte gemeinsam in die Qualitätsebene zu integrieren sind.

5.5 Terminologieebene

Die Terminologieebene dient als zentrales Begriffsverzeichnis einer betrieblichen Fachkomponente [Tu02:16]. Dazu werden in einem Lexikon alle (wesentlichen) Fachbegriffe mit ihrer Definition, erklärenden Beispielen und ihrer Beziehung zu anderen Begriffen gespeichert. Die Terminologieebene wendet sich hauptsächlich an Fachexperten (zur Komponentenauswahl), aber auch an Technikexperten (zur Komponentenanpassung und -komposition), und hat zum Ziel, alle relevanten Fachbegriffe eindeutig und allgemein verständlich zu definieren.

Dazu wird als Notationstechnik der Einsatz von Normsprachen vorgeschlagen, bei denen es sich um Ontologiedefinitionssprachen handelt, welche sowohl für einen menschlichen Benutzer als auch maschinell lesbar sind [OS96]. Dazu besteht eine Normsprache aus einem Lexikon mit geklärten Fachbegriffen und verwendet zur Bildung von Aussagen eine rationale Grammatik (rekonstruierte Grammatik der Umgangssprache), die eine Reihe von Satzbauplänen als Schablonen vorgibt. Die Verwendung von Satzbauplänen hat zwei Vorteile gegenüber der Verwendung natürlicher Sprache: Spezifikationen werden präziser und das entstehende Lexikon bildet eine einfache Ontologie, welche das automatische Auffinden von Zusammenhängen vereinfacht. Der Vorteil von Satzbauplänen gegenüber anderen Ontologienotationen (wie z. B. dem Resource Description Framework (RDF)) liegt darin, dass die entstehenden Sätze auch für Fachexperten verständlich sind, die keine Erfahrung mit formalen Modellen oder Ontologienotationen haben.

Das Defizit des Spezifikationsrahmens [Tu02] besteht darin, dass die Spezifikationsvorschriften insgesamt vage bleiben und keine Satzbaupläne angegeben werden. Die einzusetzenden Satzbaupläne sollten allerdings vom Spezifikationsrahmen vorgegeben werden, da zum einen nur durch deren Standardisierung das Ziel erreicht werden kann, Ungenauigkeiten und Vagheiten zu vermeiden [Za05:41ff.], und zum anderen nicht jedem Komponentenhersteller zugemutet werden kann, eigene Satzbaupläne zu entwerfen.

Zur Präzisierung der Vorschriften auf der Terminologieebene wird vorgeschlagen, dass die Spezifikation eines Fachbegriffs aus drei Teilen besteht:

- A) der eigentliche Lexikoneintrag, der zur Klärung des Begriffs dient;
- B) eine Aussagensammlung, die den Begriff in Beziehung zu anderen Begriffen (des Lexikons) setzt und damit eine umfassende Spezifikation des Begriffsgeflechts innerhalb der Komponente (bzw. der unterstützten Domäne) erlaubt;
- C) Restriktionen, welche Abhängigkeiten zwischen Begriffen spezifizieren.

Zu A) Die Definition eines Begriffs erfolgt in Form einer Tabelle [Tu02:18; Ov06:164]. Diese enthält neben der genauen Bezeichnung des Begriffs (Begriffswort) eine natürlichsprachliche Kurzdefinition, (optional) eine ausführlichere, ebenso natürlichsprachliche Langdefinition sowie erklärende Beispiele und ggf. Gegenbeispiele. Ein Beispiel dazu findet sich in den ersten vier Tabellenzeilen von Abb. 8.

Zu B) Der Spezifikationsrahmen UnSCom enthält eine Liste vordefinierter Satzbaupläne zur Spezifikation von Begriffen [Ov06:171]. Es wird vorgeschlagen, diese Satzbaupläne für die Terminologieebene von [Tu02] zu übernehmen. Mit Hilfe dieser Satzbaupläne lässt sich eine Aussagensammlung erstellen, welche die Zusammenhänge zwischen Begriffen standardisiert beschreibt und (unter der gleichnamigen Rubrik) Teil der Definition eines Begriffs ist. So wird z. B. in Abb. 8 festgelegt, dass ein Lagerplatz aus 0 bis beliebig vielen Lagerbeständen besteht.

Begriffswort: LAGERPLATZ
Kurzdefinition: Ein <i>Lagerplatz</i> ist die kleinste adressierbare Raumeinheit innerhalb eines LAGERS.
Langdefinition: Ein <i>Lagerplatz</i> ist die kleinste adressierbare Raumeinheit innerhalb eines LAGERS. An einem LAGERPLATZ werden Bestände von LAGERMATERIAL gelagert.
Beispiel: Lagerplatz 015-07-02 (Reihe 015, Regal 07, Ebene 02)
Aussagensammlung: Ein LAGERPLATZ hat eine ID und eine MAXIMALBELEGUNG. Eine ID ist eine Zeichenkette. Eine MAXIMALBELEGUNG ist eine Ganzzahl. Ein LAGERPLATZ besteht aus 0 bis beliebig vielen LAGERBESTAND. Ein LAGERPLATZ steht in Beziehung zu 0 bis 1 LAGERMATERIAL.
Restriktionen: Die Summe aller Bestände (LAGERBESTAND) an einem LAGERPLATZ darf den Wert von MAXIMALBELEGUNG nicht überschreiten. Ist einem LAGERPLATZ ein LAGERMATERIAL zugeordnet, kann der LAGERPLATZ nur Bestände (LAGERBESTAND) dieses Materials enthalten.

Abb.8: Spezifikation des Begriffs *Lagerplatz*

UnSCom beinhaltet Satzbaupläne für die Beschreibung der folgenden Beziehungsarten: Inklusion, Merkmalsvereinbarung, Aggregation, Konnexion und Relation. Beispielsweise lassen sich mit Hilfe der *Merkmalsvereinbarung* (auch *Partizipation*) die Merkmale eines Begriffs beschreiben – dafür dient der Satzbauplan „(Ein | Eine) A hat

(einen | eine | ein) B (und (einen | eine | ein) C)⁺“. Ein Anwendungsbeispiel dafür findet sich in der ersten Aussage (Rubrik *Aussagensammlung*) in Abb. 8, die festlegt, dass ein Lagerplatz durch die Merkmale ID und Maximalbelegung näher beschrieben wird. Durch entsprechende Erweiterungen der Satzbaupläne lassen sich auch Rollen und Kardinalitäten berücksichtigen – so handelt es sich bei der letzten Aussage in Abb. 8 um eine Relation mit der Kardinalität [0..1]. Für die vollständige Liste aller verfügbaren Satzbaupläne vgl. [Ov06:171].

Zu C) Die letzte Tabellenzeile von Abb. 8 zeigt, dass (über die Aussagensammlung hinausgehende) Abhängigkeiten zwischen Begriffen bestehen können. Dabei kann es sich um Einschränkungen bei der Komponentenverwendung handeln, die einem (die Komponente evaluierenden) Fachexperten zur Verfügung zu stellen sind. Aus diesem Grund sind solche Restriktionen ebenfalls in die Definition eines Begriffs aufzunehmen und werden unter der Rubrik *Restriktionen* in der tabellarischen Darstellung berücksichtigt [Ac07a:114]. Der Einsatz von vorgegebenen Satzbauplänen ist an dieser Stelle schwierig, da die auftretenden Restriktionen beliebig komplex sein können – stattdessen wird vorgeschlagen, die Restriktionen natürlichsprachlich zu formulieren. Eine formale Spezifikation dieser Restriktionen wäre in Zukunft wünschenswert. Dies könnte z. B. durch eine automatische Konvertierung formaler Bedingungen (in UML OCL) in natürliche oder normierte Sprache (vgl. beispielweise [Hä02] für erste Ansätze) oder durch die Verwendung einer Fachnormsprache für prädikatenlogische Aussagen (wie in [Ov06:173] angedacht) erfolgen.

5.6 Aufgabenebene

Ziel der *Aufgabenebene* ist die Beschreibung der von einer Komponente unterstützten betrieblichen Aufgaben sowie ggf. deren Zerlegung in mehrere Teilaufgaben auf fachlichem (konzeptionellem) Niveau [Tu02:19]. Die Aufgabenebene wendet sich hauptsächlich an Fachexperten und soll eine Beurteilung erlauben, ob die Komponente aus fachlicher Sicht für einen konkreten Anwendungsfall einsetzbar ist. Analog zur Terminologieebene wird festgelegt, dass die Spezifikation mit Hilfe von Normsprachen erfolgt.

Wie auch auf der Terminologieebene bleibt [Tu02] bei der Aufgabenebene vage und verzichtet auf die vollständige Angabe aller einsetzbaren Satzbaupläne. Daher sind bei einer Weiterentwicklung von [Tu02] die Spezifikationsvorschriften für Aufgaben zu präzisieren. Dazu wird vorgeschlagen, die Ergebnisse aus [Ov06:158-160,172-174] auf die Aufgabenebene zu übertragen. Zentrales Element ist dabei der Vorschlag, Begriffe und Aufgaben auf ähnliche Weise zu spezifizieren – durch eine solche Vereinheitlichung wird die Spezifikation sowohl für menschliche Benutzer (durch Systematisierung) als auch für Tools vereinfacht. Analog zu Begriffen besteht damit auch die Aufgabendefinition aus drei Teilen:

- A) Die Definition der betrieblichen Aufgabe erfolgt durch Angabe ihres Namens, einer Kurzdefinition, einer optionalen Langdefinition sowie von Beispielen und ggf. Gegenbeispielen. Der Name einer betrieblichen Aufgabe besteht aus einem Substantiv (dem primären Objekt der Handlung), einem optionalen Adjektiv (zur

Präzisierung der Handlung) und einem Verb (dem Handlungsbezeichner) – Beispiele dafür sind die betrieblichen Aufgaben *Lagerbestand abfragen* und *Lagermaterial manuell einlagern*.

- B) Die Aussagensammlung spezifiziert, in welcher Beziehung die Aufgabe zu anderen betrieblichen Aufgaben sowie zu den auf der Terminologieebene beschriebenen Begriffen steht. Im Rahmen der Aussagensammlung kann die Zerlegung einer Aufgabe in Teilaufgaben (Komposition), die Angabe von Aufgabenvarianten (Inklusion) sowie die Beschreibung von Aufgabenmerkmalen (Merkmalsvereinbarung), mit denen die Beziehung der Aufgabe zu Begriffen angegeben wird, beschrieben werden. Für eine vollständige Übersicht über alle verfügbaren Satzbaupläne vgl. [Ov06:176].
- C) Für die Ausführung betrieblicher Aufgaben können Restriktionen bestehen – so kann beispielsweise nur ein Lagerungsauftrag der Art *Einlagerung* zum Ausführen der Aufgabe *Lagermaterial einlagern* verwendet werden. Einschränkungen solcher Art sind für Fachexperten bei der Komponentenbewertung relevant und müssen deshalb (als Restriktion unter der gleichnamigen Rubrik) in die Aufgabenspezifikation aufgenommen werden. Wie auf der Terminologieebene erfolgt die Spezifikation von Restriktionen in natürlicher Sprache.

Ein Beispiel für die Anwendung der Spezifikationsvorschriften findet sich in Abb. 9.

Aufgabe: LAGERMATERIAL EINLAGERN
Kurzdefinition: Das <i>Einlagern von Lagermaterial</i> umfasst die Aufnahme einer bestimmten Anzahl von LAGERMATERIAL in das LAGER und dessen physische Lagerung an einem (oder mehreren) LAGERPLATZ.
Beispiel: Drei Paletten von Lagermaterial ABC-XYZ ins Lager aufnehmen und auf Lagerplatz 015-07-02 lagern
Aussagensammlung: Ein LAGERVERWALTER tut LAGERMATERIAL EINLAGERN mit einem LAGERUNGSAUFTRAG.
Restriktionen: Die AUFTRAGSART von LAGERUNGSAUFTRAG ist Einlagerung. Der LAGERUNGSAUFTRAG enthält (EINHEITENANZAHL viele) LAGERPLÄTZE.

Abb. 9: Spezifikation der betrieblichen Aufgabe *Lagermaterial einlagern*

5.7 Vermarktungsebene

Zweck der Vermarktungsebene ist die Spezifikation solcher Merkmale, die benötigt werden, um die Komponente *betriebswirtschaftlich-organisatorisch* handhabbar zu machen. Die auf dieser Ebene spezifizierten Merkmale sind insbesondere für Verkäufer und Einkäufer sowie für Assemblierer und Qualitätssicherer von Interesse [Tu02:21]. Als primäre Notationstechnik wird die Tabellenform vorgeschlagen – außerdem werden die im Rahmen der Vermarktungsebene zu spezifizierenden Merkmale explizit vorgegeben. Als Erweiterung wird vorgeschlagen, bei allen vorgegebenen Merkmalen zu überprüfen, ob die Angabe mehrerer Merkmalswerte möglich sein sollte. So ist es denkbar, dass eine Komponente mehrere Komponententechnologien (typischerweise mehrere Versionen derselben Technologie) unterstützt.

6 Zusammenfassung

Diese Arbeit beschäftigte sich mit der Weiterentwicklung der Spezifikation betrieblicher Softwarekomponenten. Ausgangspunkt der Untersuchungen war der Spezifikationsrahmen für betriebliche Fachkomponenten (Memorandum) [Tu02]. Es wurde vorgeschlagen, diesen Spezifikationsrahmen weiterzuentwickeln – dazu wurden bestehende Defizite identifiziert und es wurde ein umfassender Vorschlag unterbreitet, wie das Memorandum weiterentwickelt werden sollte. Die Ergebnisse dieses Beitrags können damit als Ausgangspunkt für Diskussionen zu einem „Memorandum 2.0“ dienen. Neben einer Überarbeitung des Spezifikationsrahmens an sich sollte in Zukunft ein Spezifikationstool erstellt werden, welches auf dem Memorandum basiert, und es sollten Möglichkeiten zur Vereinfachung der Komponentenspezifikation (z. B. durch den Einsatz von Spezifikationsmustern [AT06]) identifiziert und umgesetzt werden.

Literaturverzeichnis

- [Ac03] *Ackermann, J.*: Spezifikation von Fachkomponenten mit der UML 2.0. In: *Turowski, K. (Hrsg.): 4. Workshop Modellierung und Spezifikation von Fachkomponenten*. Bamberg 2003, S. 23-30.
- [Ac04] *Ackermann, J.*: Zur Beschreibung datenbasierter Parametrisierung von Softwarekomponenten. In: *Turowski, K. (Hrsg.): Architekturen, Komponenten, Anwendungen – Proceedings zur 1. Verbundtagung Architekturen, Komponenten, Anwendungen (AKA 2004)*. LNI Band P-57. Augsburg 2004, S. 131-149.
- [Ac07a] *Ackermann, J.*: Spezifikation der parametrisierungsbezogenen Eigenschaften betrieblicher Fachkomponenten. Dissertation. Augsburg 2007.
- [Ac07b] *Ackermann, J.*: Using a Specification Data Model for Specification of Black-Box Software Components. In: *Enterprise Modelling and Information Systems Architectures 2 (2007) 1*, S. 3-13.

- [AT06] *Ackermann, J.; Turowski, K.*: A Library of OCL Specification Patterns for Behavioral Specification of Software Components. In: *Dubois, E.; Pohl, K. (Hrsg.): CAiSE 2006.* Springer-Verlag LNCS 4001. Luxemburg 2006, S. 255-269.
- [Be99] *Beugnard, A.; Jézéquel, J.-M.; Plouzeau, N.; Watkins, D.*: Making Components Contract Aware. In: *IEEE Computer 32 (1999) 7*, S. 38-44.
- [Br00] *Brown, A.W.*: Large-Scale, Component-Based Development. Prentice Hall. Upper Saddle River 2000.
- [CD01] *Cheesman, J.; Daniels, J.*: UML Components. Addison-Wesley. Boston 2001.
- [CT01] *Conrad, S.; Turowski, K.*: Temporal OCL: Meeting Specification Demands for Business Components. In: *Siau, K.; Halpin, T. (Hrsg.): Unified Modeling Language: Systems Analysis, Design and Development Issues.* Idea Group. Hershey 2001, S. 151-165.
- [DW98] *D'Souza, D.F.; Wills, A.C.*: Objects, Components, and Frameworks with UML: The Catalysis Approach. Addison-Wesley. Reading 1998.
- [FK98] *Frolund, S.; Koistinen, J.*: QML: A Language for Quality of Service Specification. Technical Report HPL-98-10. Hewlett-Packard Laboratories 1998.
- [GG06] *Geisterfer, C.J.M., Ghosh, S.*: Software Component Specification: A Study in Perspective of Component Selection and Reuse. In: *Proceedings of the 5th International Conference on COTS-Based Software Systems (ICCBSS'06).* IEEE Computer Society Press. Orlando 2006, S. 100-108.
- [Gr98] *Griffel, F.*: Componentware. Konzepte und Techniken eines Softwareparadigmas. dpunkt Verlag. Heidelberg 1998.
- [Hä02] *Hähnle, R.; Johannisson, K.; Ranta, A.*: An Authoring Tool for Informal and Formal Requirements Specifications. In: *Kutsche, R.-D.; Weber, H. (Hrsg.): Fundamental Approaches to Software Engineering: 5th International Conference (FASE 2002).* Springer-Verlag LNCS 2306. Grenoble 2002, S. 233-248.
- [HL01] *Hemer, D.; Lindsay, P.*: Specification-based retrieval strategies for module reuse. In: *Grant, D.; Sterling, L. (Hrsg.): Proceedings 2001 Australian Software Engineering Conference.* IEEE Computer Society. Canberra 2001, S. 235-243.
- [HT02] *Hahn, H.; Turowski, K.*: General Existence of Component Markets. In: *Trappl, R. (Hrsg.): Sixteenth European Meeting on Cybernetics and Systems Research (EMCSR).* Vol. 1. Vienna 2002, S. 105-110.
- [OMG05a] *OMG (Hrsg.):* Object Constraint Language. Version 2.0, formal/06-05-01. URL: <http://www.omg.org/technology/documents>, Abruf am 2008-07-07.
- [OMG05b] *OMG (Hrsg.):* Unified Modeling Language: Superstructure. Version 2.0, formal/05-07-04. URL: <http://www.omg.org/technology/documents>, Abruf am 2008-07-07.
- [OS96] *Ortner, E., Schienmann, B.*: Normative Language Approach: A Framework for Understanding. In: *Thalheim, B. (Hrsg.): Conceptual Modeling. ER '96, 15th International Conference on Conceptual Modeling.* Springer-Verlag LNCS 1157. Cottbus 1996, S. 261-276.
- [Ov04] *Overhage, S.*: Zur Spezifikation von Komponenten der Informationssystementwicklung mit Softwareverträgen. In: *Rebstock, M. (Hrsg.): Tagungsband Modellierung betrieblicher Informationssysteme – MobIS 2004.* LNI Band P-40. Essen 2004, S. 3-20.

- [Ov06] *Overhage, S.*: Vereinheitlichte Spezifikation von Komponenten – Grundlagen, UnSCom Spezifikationsrahmen und Anwendung. Dissertation. Augsburg 2006.
- [Sz02] *Szyperski, C.; Gruntz, D.; Murer, S.*: Component Software: Beyond Object-Oriented Programming. 2. Aufl. Addison-Wesley. Harlow 2002.
- [Tu02] *Turowski, K. (Hrsg.)*: Vereinheitlichte Spezifikation von Fachkomponenten – Memorandum des Arbeitskreises 5.10.3 Komponentenorientierte betriebliche Anwendungssysteme. Universität Augsburg. Augsburg 2002. URL: <http://www.fachkomponenten.de>. Abruf am 2008-07-07.
- [YS97] *Yellin, D.; Strom, R.*: Protocol Specifications and Component Adaptors. In: ACM Transactions on Programming Languages and Systems 19 (1997), S. 292–333.
- [Za05] *Zaha, J.M.*: Automatisierte Kompatibilitätstests für fachliche Software-Komponenten. Dissertation. Augsburg 2005.
- [ZG03] *Ziemann, P.; Gogolla, M.*: OCL Extended with Temporal Logic. In: *Broy, M.; Zamulin A.V. (Hrsg.)*: Perspectives of Systems Informatics – 5th International Andrei Ershov Memorial Conference. Springer-Verlag LNCS 2890. Novosibirsk 2003, S. 351-357.

Formale Kontrolle kollaborativer B2B-Geschäftsprozesse

Janina Fengel¹, Michael Rebstock¹, Carlo Simon²

¹ Fachbereich Wirtschaft
Hochschule Darmstadt
Haardtring 100
64285 Darmstadt
janina.fengel@h-da.de
rebstock@h-da.de

² Fachbereich Wirtschaftsinformatik
Provadis School of International
Management and Technology AG
Industriepark Höchst
65926 Frankfurt am Main
carlo.simon@provadis-
hochschule.de

Abstract: Die Basis der Gestaltung von B2B-Kollaboration ist die Koordination der Beschaffungs- und Vertriebsaktivitäten. Zentraler Interaktionspunkt dabei sind Verhandlungen, besonders im Falle von Geschäftstransaktionen zu industriellen Gütern und Dienstleistungen mit multiplen Positionen und Eigenschaften. In solchen komplexen Szenarien hängen Entscheidungen verschiedener Prozesse voneinander ab. Um diese überwachen und managen zu können, wird hier ein SPL-basierter formaler Kontrollmechanismus für multilaterale multiattributive B2B-Verhandlungsprozesse vorgestellt.

1 Kollaboration im B2B

Die Fundamente einer marktwirtschaftlich orientierten Volkswirtschaft sind der Ausgleich von Angebot und Nachfrage durch Tausch und das grundsätzliche unternehmerische Motiv der Wertsteigerung. Markttransaktionen ermöglichen Handel durch Vereinbarung eines Leistungsaustausches und dienen damit der Leistungscoordination [Sc93; PRW98]. Ein- und Verkauf von Konsumgütern widmet sich dem Handel mit überwiegend standardisierten Gütern, während bei Transaktionen von Industriegütern und Dienstleistungen häufig erst in deren Verlauf Preis, Leistung, Ausgestaltung des Gutes und Austauschbedingungen bestimmt werden [Ho06; He07]. Anders als auf Konsumgütermärkten ist die Kaufentscheidung dabei nicht allein von Marketing und Vertrieb des Anbieters abhängig, sondern unterliegt der gegenseitigen Einflussnahme [He07]. Diese Entscheidungsfindung dient der gemeinsamen Entwicklung von Lösungen und schafft Vertrauen zwischen den Partnern [Ho06]. Zentraler Bestandteil solcher B2B-Interaktionen sind Verhandlungen, die als kollektive Entscheidungsfindungsprozesse zur Aushandlung von Leistung und Gegenleistung verstanden werden können [He07], welche die dezentralisierten, interdependenten Entscheidungsprozesse von zwei oder mehr Parteien zum gegenseitigen Nutzen kombinieren [BS97; Bi05]. Ein solcher Prozess beruht auf gegenseitigem Informationsaustausch, in der Regel mit dem Ziel einer Einigung [Re01], meist in Form eines Kompromisses [Ke00].

¹ Die Autoren werden vom BMBF gefördert unter FKZ 1728X07 Projekt MODI.

Häufig sind der Preis und die Angebotsmerkmale vor der Vermarktung eines industriellen Gutes noch nicht bekannt, da kunden- oder auftragsindividuell produziert wird. Zum Zeitpunkt der Verhandlung ist das Gut noch nicht erstellt, Kundenwünsche können abhängig von dessen Präferenzen individuell beachtet oder gemeinsam mit dem Kunden ausgestaltet werden [Ge07]. Entwicklung und Produktion erfolgen zeitlich nach der Vermarktung. Daher sind Absatz und Beschaffung industrieller Güter und Dienstleistungen mehr als einfache Einkaufs- bzw. Verkaufsvorgänge [He07]. Handelt es sich nicht um katalogbasierte Beschaffung eindeutig definierter unveränderbarer Produkte zu einem bestimmten Preis, sondern um den Handel beliebiger Sachgüter oder Dienstleistungen mit verschiedenen verhandelbaren Eigenschaften, werden diese Vorgänge sehr komplex und bedingen einander.

Im Folgenden beschreiben wir eine Methode zur Herstellung formaler Kontrolle über mehrere in Beziehungen stehende elektronische Geschäftsprozesse am Beispiel kundenindividueller Aufträge. Dazu werden in Kapitel 2 die Abläufe bei der Angebotsentwicklung für solche Aufträge dargestellt. In Kapitel 3 wird unter Verwendung der Semantic Process Language (SPL) das Grundmodell eines Verhandlungsszenarios entwickelt. In Kapitel 4 zeigen wir die Anwendung dieser Methode beispielhaft für ein prototypisch existierendes, semiautomatisiertes, multiattributives elektronisches Verhandlungsunterstützungssystem. Der Beitrag endet mit der Darstellung themenverwandter Arbeiten in Kapitel 5 und einer kurzen Diskussion und Ausblick in Kapitel 6.

2 Interdependenzen interorganisationaler Geschäftsprozesse im B2B

Eine Grundvoraussetzung termin- und qualitätsgerechter Lieferung ist effektives Management, insbesondere bei unternehmensübergreifenden Prozessen in Wertschöpfungsketten und Netzwerken [Krup07]. Unternehmen interagieren darin mit den ihnen vor- und nachgelagerten Stufen über die Beschaffung bzw. den Verkauf von nicht standardisierten Sachgütern und Dienstleistungen. Dabei sind die Rollen des Anbieters bzw. Zulieferers und des Nachfragers bzw. Herstellers nicht an ein Unternehmen gebunden, sondern ergeben sich aus der jeweiligen Transaktionssituation und hängen vom Verhandlungsgegenstand in seiner Funktion als konkretes Entscheidungsproblem ab. In der Interaktion mit der vorgelagerten Stufe kann ein Unternehmen als Nachfrager auftreten, während es parallel für die nachgelagerte Stufe als Anbieter fungiert [He07]. Unternehmen sind dabei Teil unterschiedlicher Kooperationskonstellationen bei der Vermarktung, Entwicklung, Herstellung und dem Verkauf. Zur Entscheidungsfindung in Markttransaktionen ist daher eine marktgerichtete Integration von Absatz- und Beschaffungsmanagement unabdingbar [Krus07].

2.1 Abhängigkeiten von Entscheidungen

Bei kundenspezifischer oder auftragsindividueller Produktion ist eine enge interne Abstimmung von Beschaffung und Vertrieb notwendig. Zur Erstellung wettbewerbsfähiger Angebote sind für den Vertrieb genaue Kenntnisse zu den von den Lieferanten erhältlichen Leistungen nötig, während die Beschaffung durch Kenntnis der Kundenanforderungen gezielter agieren kann [Krus07].

Langfristig kann eine marktorientierte Abstimmung zu prozessorientiertem Netzwerkmanagement führen [Krus07]. Dabei sind Verhandlungen zentraler Bestandteil des Beschaffungs- bzw. Vertriebsprozesses [Ko07]. Im Geschäftsalltag sind die meisten Beschaffungsentscheidungen nicht ausschließlich preisabhängig, wie beispielsweise bei Auktionen, bei denen das Transaktionsobjekt und der Interaktionsverlauf vorab fix definiert sind, sondern sind beeinflusst von der Produktbeschaffenheit und dessen Verfügbarkeit, den Vertragskonditionen, der Historie der Geschäftsbeziehung sowie strategischen Überlegungen und firmenpolitischen Vorgaben [Re01]. Verhandelt wird zumeist mit mehreren Leistungsträgern parallel über mehrere Positionen mit unterschiedlichen Eigenschaften [BPS05], unabhängig davon, ob es sich um Vorgänge des operativen Einkaufs oder Fragen des Sourcing handelt [Sa07]. Dabei können Verhandlungen verschiedene Formen annehmen, von einfachen Verfügbarkeitsklärungen über Bestimmungen der vom Hersteller akzeptierten Preis-Mengen-Kombinationen bis zur kollaborativen Ausarbeitung von Lösungen für einmalig zu erstellende Leistungen [He07].

Unternehmen, die sich auf die Herstellung kunden- oder auftragsspezifischer Güter oder Dienstleistungen spezialisiert haben, benötigen meist individuell für jeden Auftrag andere Zusammenstellungen von Rohmaterialien, Produktbestandteilen oder Dienstleistungskomponenten von unterschiedlichen, manchmal konkurrierenden Lieferanten. Daher ist das Ergebnis einer der den Kundenauftrag betreffenden Verhandlungen häufig ausschlaggebend für die Entscheidungsfindung bei den anderen Verhandlungen. Kann beispielsweise ein spezieller, nicht austauschbarer Bestandteil nicht beschafft werden, führt die resultierende Lieferunfähigkeit der zu erstellenden Leistung dazu, dass die Beschaffung aller anderen Bestandteile nicht mehr nötig ist. Werden zu einem Auftragsbestandteil Verhandlungen mit verschiedenen Lieferanten über ein austauschbares Bestandteil geführt, hängt die Entscheidung für einen von ihnen vom Status und Ergebnis der Verhandlungen mit den anderen Lieferanten ab. Der Einkäufer sollte dann sicherstellen, dass das zu beschaffende Auftragsbestandteil nicht mehrfach eingekauft wird. Solchartige multilaterale Verhandlungen sind voneinander abhängig, da sie sich beeinflussen und sich der Stand und Inhalt einer der Prozesse auf die anderen auswirken, obwohl sie bilateral geführt werden können. Ein vergleichbares Ergebnis wäre durch das Führen mehrerer unabhängiger bilateraler Verhandlungen nicht erreichbar [Hü04]. Ohne die Beachtung der Ergebnisse der in Bezug stehenden Verhandlungen ließe sich nicht aus einer Menge von Angeboten das Beste ermitteln oder eine Auswahl zugunsten eines Gutes treffen, wenn diese von einer bereits getätigten Vorauswahl abhängt.

2.2 Grundlegende Kommunikationsstrukturen

Basierend auf dem multilateralen Verhandlungsmodell nach [GM98] lassen sich im B2B zwei grundsätzliche Kommunikationsstrukturen beschreiben. Bei wettstreitenden Verhandlungen wird versucht, das beste Angebot aus einer Menge konkurrierender Angebote herauszufinden, während Verhandlungen, die in wechselseitiger Abhängigkeit stehen, nur dann mit einem Vertragsabschluß beendet werden können, wenn alle darin angefragten Güter oder Dienstleistungen tatsächlich beschafft werden können [SFR07]. Abbildung 1 zeigt die entstehenden Strukturen.

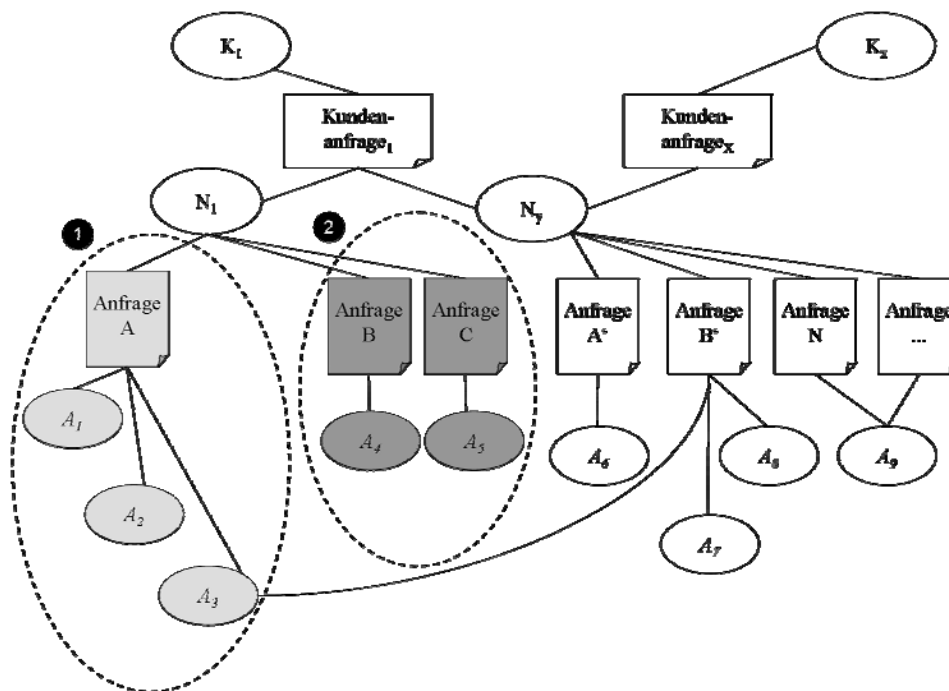


Abb. 1: Kommunikationsstrukturen multilateraler B2B-Prozesse

Kunden ($K_1 \dots K_N$) fragen ihre Zulieferer an, die dadurch zum Nachfrager ($N_1 \dots N_j$) werden, indem sie wiederum ihre Zulieferer anfragen.

In Szenario 1 wird eine Konkurrenzstruktur dargestellt. Beispielsweise kann hierbei eine Ware, sei es ein Sachgut oder eine Dienstleistung, vom Angefragten, nun Nachfrager (N_1), in vergleichbarer Qualität von verschiedenen Anbietern (A_1, A_2, A_3) bezogen werden und es können mehrere parallele Verhandlungen mit ihnen begonnen werden, um das Gewünschte in der benötigten Qualität zum bestmöglichen Preis beziehen zu können. Dies kann ein Wettbewerbsvorteil sein, besonders da hier der Auftraggeber des Nachfragers seinerseits mehrfach für denselben Auftrag angefragt hat. Die Anbieter wiederum können ihrerseits auch in mehrere Verhandlungen bezüglich derselben Ware involviert sein, unabhängig davon, ob bei konkurrierenden Nachfragern dieselbe Kundenanfrage vorliegt oder nicht.

So entsteht für alle Beteiligten die Notwendigkeit mehrere parallel laufende, teilweise voneinander abhängige Verhandlungen zu überwachen. Es gilt sicherzustellen, dass nur eines, vorzugsweise das günstigste, der konkurrierenden Angebote akzeptiert wird. Dies erfordert zum einen Kenntnis bezüglich des Status einer jeder Verhandlung und zum anderen genauen Angebotsvergleich. Oft ist es im Verlauf nötig einige Angebote in schwebendem Zustand zu belassen, während auf noch ausstehende Antworten der Konkurrenz gewartet wird. Die letztendliche Entscheidung zugunsten eines der Angebote gebietet die Ablehnung aller anderen. Ziel ist es, den Beschaffungsbedarf nur einmal bestmöglichst zu erfüllen.

Szenario 2 beschreibt die Fälle, in denen verschiedene Waren als Bestandteile für einen kundenindividuellen Auftrag bezogen werden. Hierbei wird es für einen Nachfrager (N_i) nötig, mit mehreren Anbietern (A_4, A_5) zu verhandeln, da jeder eines oder mehrere der benötigten Teile im Angebot hat, jedoch keiner alle. Somit liegen mehrere Verhandlungen vor, die überwacht werden müssen. Ziel ist es hierbei, alle benötigten Auftragsbestandteile zur Auftragserfüllung einzukaufen. Die Unmöglichkeit eines der Bestandteile beziehen zu können, führt gegebenenfalls dazu, dass die Beschaffung aller anderen Bestandteile unnötig wird. Der Nachfrager muss alle zusammenhängenden Verhandlungen daraufhin überprüfen, ob sein Bedarf vollständig erfüllt ist, aber nur jeweils einmal und in der gewünschten Qualität und Menge. Dabei sind die Angebote der Verkäufer besonders dann genau zu überwachen, wenn Bestandteile nicht nur von einem, sondern von mehreren von ihnen angeboten werden. Stellt sich heraus, dass der Auftrag des Kunden nicht erfüllt werden kann, ist mit diesem entsprechend neu zu verhandeln oder alle Verhandlungen mit den betroffenen Anbietern müssen beendet werden.

3 Informationsstrukturen ausführbarer Prozesse

Laufen zwei oder mehr Prozesse gleichzeitig ab, und eines der Prozessergebnisse hat Auswirkungen auf die Fortführung der anderen, muss ihre Ausführung aufeinander abgestimmt werden. Zwecks elektronischer Unterstützung sind dafür durchgehend eindeutig interpretierbare Informationen erforderlich. Das erfolgreiche Managen von solcherart in Wechselbeziehung stehenden Vorgängen erfordert Informationsaustausch zwischen ihnen in Echtzeit [MC94]. Von Interesse hierbei ist Wissen bezüglich der einzelnen Prozessstatus und der Prozessergebnisse, besonders im Falle multilateraler Verhandlungen, da mehrere Parteien an der Ergebnisfindung beteiligt sind. Die Komplexität multilateraler Verhandlungen ist dabei deutlich höher im Vergleich zu isolierten bilateralen Verhandlungen, da die zu verarbeitende Informationsmenge für die Ausführung der nächsten Schritte bedeutend größer ist. Die Ausführung eines Verhandlungsprozesses hängt jeweils vom Status der anderen, dazu in Beziehung stehenden ab. Der Prozessstatus ergibt sich aus dem formalen Prozessmodell, dem Zustand der Ausführung des aktuellen Prozesses bezüglich der nächstmöglichen Schritte sowie den aktuellen Informationsobjekten, hier insbesondere die aktuell verbindlichen Angebote, die in Beantwortung der Anfragen erstellt wurden.

Diese Art der Information ist genau auch die, die für eine automatisierte Ausführung von Geschäftsprozessen in Workflow-Management-Systemen relevant ist. Ein prozessorientiertes elektronisches Verhandlungsunterstützungssystem kann somit im weitesten Sinne als Workflow-Management-System verstanden werden, denn eine solche Unterstützungsumgebung arbeitet mit dem Wissen zu diesen Aspekten eines Verhandlungsstatus. Dabei sind in semiautomatisierten Unterstützungssystemen die endgültigen Entscheidungen dem Nutzer überlassen, sodass im Gegensatz zu überbetrieblichen Workflows, wie beispielsweise bei EDI, gegeben durch die Interaktionsmöglichkeiten keine festen Protokolle nutzbar sind. Daher kann auch eine Verlaufsabbildung mit Hilfe von Business Rules nicht das Spektrum der Möglichkeiten erfassen.

Ist das Modell selbst direkt ausführbar und es muss keine zusätzliche Prozessinformation manuell hinzugefügt werden, sollte die Modellierungssprache eine lokale Zustandssemantik aufweisen. Dies ist beispielsweise bei Workflow Nets gegeben [Aa98; AH02]. Allerdings bieten diese keine Unterstützung bei der Modellentwicklung. Daher wird hier die Semantic Process Language (SPL) verwendet [Si06]. Es handelt sich dabei um eine formale, petrinetzbasierte Prozesssprache. Sie ist unabhängig von der verwendeten Technologie nutzbar. Für die Definition der Semantik jedes ihrer Worte werden Module verwendet, die mit Hilfe so genannter Module Nets, einer Art Workflow Net, implementiert werden. Dabei interpretieren Module Prozessmengen, wobei in einem Prozess Abfolgen bestimmter Aktionen enthalten oder verboten sein können. Ebenso erlauben Module die Festlegung alternativen, nebenläufigen und iterierenden Verhaltens. Ein Tutorial sowie eine umfassende formale Einführung zur Implementierung finden sich in [Si08]. Module Nets sind formal beschreibbar als:

Definition: Module Net

Ein *Module Net* $M = (N, i)$ besteht aus einem Petrinetz $(N = (P, T, F, W))$ mit einer eindeutigen Starttransition $s(\bullet s = \emptyset)$, einer eindeutigen Zieltransition $g(\bullet g = \emptyset)$, und einer Interpretationsfunktion i über T mit $i(s) \mapsto start$, $i(g) \mapsto goal$ und für alle anderen Transitionen $i(t) \mapsto E_A$, die (möglicherweise leere) Menge miteinander vereinbarer Elementarprozesse einer Menge von Aktionen A .

SPL baut auf der Semantik von Petrinetzen auf. Da die Dynamik von Petrinetzen über eine zustandsbasierte Semantik definiert ist, eignen sie sich für die Spezifikation ausführbarer Workflow-Beschreibungen. Weiterhin ist damit direkt eine graphische Visualisierung gegeben. Somit bietet der Einsatz der SPL formale Spezifikation und Visualisierung, vergleichbar mit dem Vorgehen bei der Spezifizierung von Geschäftsprozessen mit Hilfe der Business Process Execution Language BPEL [An03]. Bei der Definition von Prozessen in BPEL wird indes die genutzte Semantik nicht eindeutig beschrieben und kann aufgrund des entstehenden Interpretationsspielraums zu Missverständnissen führen. Bei Nutzung der SPL lassen sich Prozessmodelle formal mathematisch durch Verwendung der Module beschreiben und mit den ihnen entsprechenden Module-Net-Interpretationen darstellen.

Prozesse der Module-Net-Implementierung einer SPL-Spezifikation entsprechen Schaltfolgen, in denen die beiden Start- und Zieltransitionen genau einmal schalten und ansonsten die leere Anfangsmarkierung des Netzes reproduziert wird.

Abbildung 2 zeigt die Spezifikation zweier Verhandlungsprozesse in SPL als Module-Net [SR04]. Es handelt sich um die Darstellung einer Verhandlung und zeigt den Prozess eines Nachfragers (N) und den eines Anbieters (A). Dabei wird davon ausgegangen, dass die Verhandlungspartner ihre Prozesse jeweils unabhängig voneinander modellieren. Die Rollen der Teilnehmer sind hier nicht als Käufer und Verkäufer definiert, sondern als Nachfrager und Anbieter, da innerhalb des zugrunde liegenden generischen Szenarios ein Initialgebot sowohl ein Verkaufsangebot als auch ein Kaufangebot sein kann.

Ein Nachfrager (N) gibt ein *Initialangebot* ab, indem er eine bestimmte Leistung anfragt, und erwartet eine Reaktion des Anbieters (A). Dies kann ein *Gegenangebot*, die *Annahme* oder der *Abbruch* sein. In den beiden letzten Fällen endet die Verhandlung. Ansonsten nimmt der Nachfrager eine Angebotsprüfung und Modifikation vor, sodass sich der Prozess wiederholen kann. Alternativ kann der Nachfrager durch *Abbruch* oder *Annahme* die Verhandlung beenden.

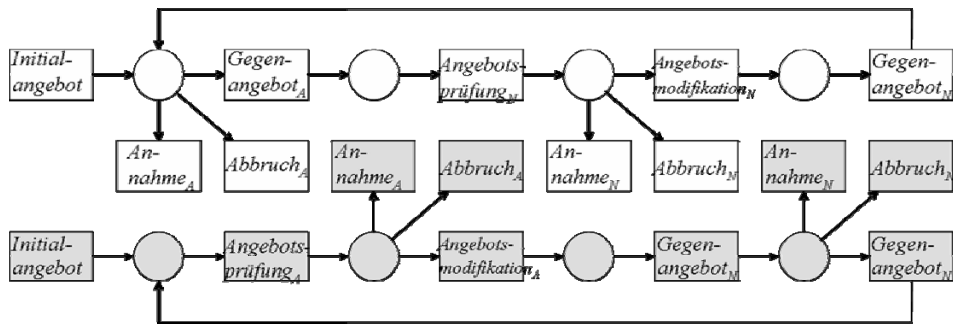


Abb. 2: Module-Net-Implementierung einer Verhandlung

Nach dem Feuern der Starttransition wird das aktuell bindende Vertragsangebot ausgetauscht, überprüft und vermutlich mehrmals abgewandelt, bis einer der Verhandlungspartner das Angebot akzeptiert oder die Verhandlung abbricht. In jedem dieser Fälle feuert eine der Zieltransitionen, sodass die Markierung des Module-Nets entfernt und das korrekt erreichte Prozessende angezeigt wird.

Dieses Modell einer bilateralen Verhandlung ist das Fundament für die Modellierung multilateraler Verhandlungen. Insgesamt erfordert die mehrfache, parallele Ausführung eines Prozessmodells in einem Workflow-Management-System die Repräsentation der Prozessstruktur und des aktuellen Status gleichermaßen. Der aktuelle Status eines Petrinetzes wird durch seine Markierung angezeigt. In der Module-Net-Spezifikation eines Geschäftsprozesses sind die Markierungstoken die Prozessidentifikationsnummern der einzelnen, parallel zu anderen Prozessspezifikationen ausführbaren Prozesse. Ebenso wird die Zugehörigkeit eines Prozesses zu einer spezifischen Prozessversion repräsentiert, da mehr als eine Version eines bestimmten Prozesstyps vorliegen kann, wie beispielsweise im Falle unterschiedlicher Verhandlungsstrategien. Durch die Verwendung des Konstrukts *Action* lassen sich die Transitionen weiter detaillieren [Si08]. Alle für die Ausführung einer Module-Net-Spezifikation eines Geschäftsprozesses relevanten Informationen werden in Abbildung 3 im Überblick gezeigt.

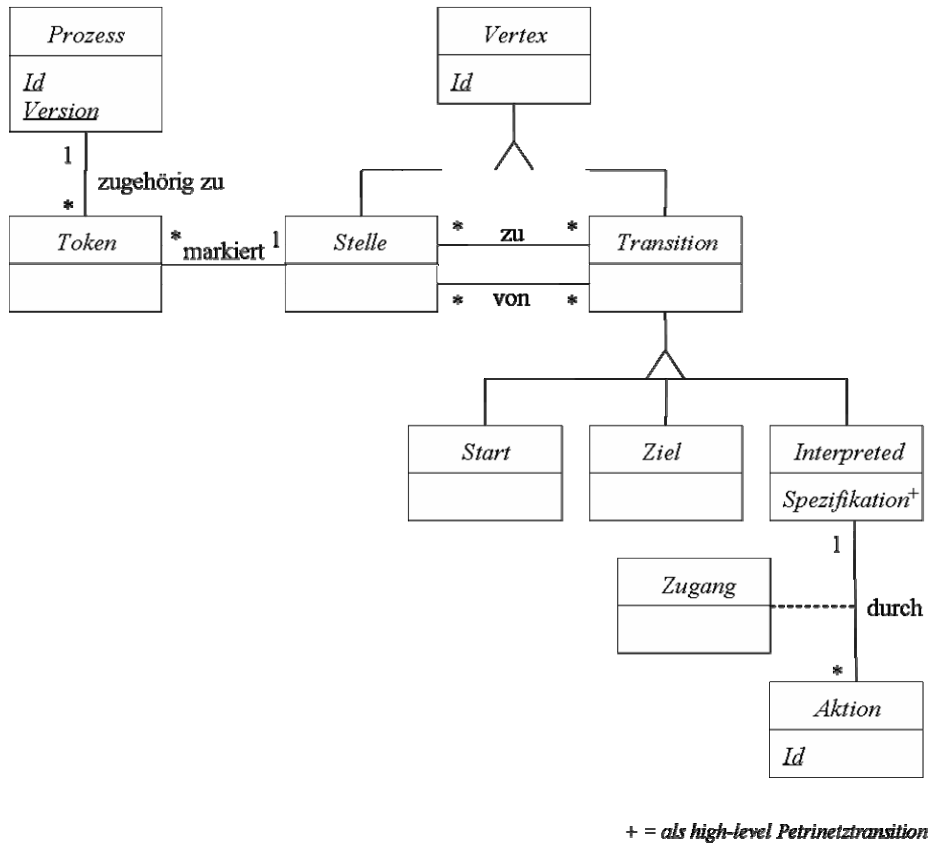


Abb. 3: Datenmodell für Prozessausführungen

Das Datenmodell zeigt die Informationsobjekte, die für die Ausführung mehrerer Instanzen des Prozessmodells in verschiedenen Versionen erforderlich sind. Dabei kann die Beziehung zwischen dem Prozessfortschritt, repräsentiert als *Interpreted Transition*, und der aktuellen Aktion mit einer Zugangskontrolle (*Access*) gesichert werden.

4 Implementierung eines formalen Kontrollmechanismus

Die Nutzung der beschriebenen Informationsobjekte erlaubt die Kontrolle mehrerer parallel ablaufender, voneinander abhängiger Prozesse. Betrachtet man Verhandlungen als Knotenpunkte im Verlauf von B2B-Interaktionen, bietet es sich an, eine entsprechende Kontrollfunktion in ein Unterstützungssystem zu integrieren.

4.1 Erweiterung eines Verhandlungsunterstützungssystems

In Verwirklichung des vorgestellten Kontrollmechanismus wird ein prototypisch vorliegendes, semiautomatisiertes, multiattributives elektronisches Verhandlungsunterstützungssystem namens M2N erweitert. Es handelt sich um ein webbasiertes Anwendungsframework, das sowohl stand-alone als auch über Web-Services in bestehende Architekturen eingebunden betrieben werden kann, unabhängig vom gewünschten Endgerät. Das System ist in Java implementiert und erfordert keine Softwareinstallation auf Nutzerseite. Nutzer können in beliebigem Umfang verhandeln, Angebote und Gegenangebote unterbreiten, diesen zuzustimmen, sie verändern oder ablehnen. Der Verhandlungsinhalt ist im Sinne der Verhandlungsattribute und -positionen unbeschränkt, es können n beliebige Positionen mit m beliebigen Attributen und beliebigen Attributwerten eingeführt werden. Verhandlungen können ad-hoc auch mit bisher, technisch wie unternehmerisch, Unbekannten begonnen werden. Es ist vorab nicht notwendig Schnittstellen zu spezifizieren oder eine Ontologie oder Vorabdefinition von Produkten und ihrer Konfigurationsmöglichkeiten in Katalogform vorzunehmen. Zur Auflösung dadurch eventuell entstehender semantischer Mehrdeutigkeiten kann ein semantischer Synchronisationsdienst genutzt werden, wie in [RFP08] beschrieben. Somit bietet M2N ungeachtet der Unternehmensgröße, Anwendungsdomäne oder Plattform Unterstützung für frei formulierbare Verhandlungen. Der Basis-Workflow umfasst die Instanziierung, das wechselseitige Zuweisen im Rahmen bilateraler Verhandlungen und den Abschluss. Es ist dabei keine Vollautomatisierung angestrebt, sondern der menschliche Akteur kann jederzeit eingreifen und die letztendlichen Entscheidungen treffen [Re01; Hü04]. Zur Koordination aller Aktivitäten bietet das System eine Übersicht und Statuskontrolle aller laufenden Verhandlungen.

4.2 Modellierung der Kontrollregeln

Dank der durchgängigen Modellierung aller Aktionen eines Prozesses können ausgewählte bilaterale Verhandlungen in Bezug gesetzt werden, sodass sie als ein multilaterales Verhandlungsbündel behandelt werden können. Jede einzelne Verhandlung wird zwischen Anbieter und Nachfrager ausgeführt, wobei die Vertraulichkeit des Geschäftsvorfalles vollumfänglich garantiert ist, so als ob die Verhandlung unabhängig von allen anderen Vorgängen stattfände. Die Bündelung ermöglicht Nutzern automatisch Statusprüfungen aller verbundenen Verhandlungen vorzunehmen, wobei das System Empfehlungen und Warnungen generieren kann. Auf dieser Basis können informierte Entscheidungen getroffen werden und kritische Informationen werden aktiver Bestandteil. Durch die Formalisierung und Bezugsherstellung zwischen Verhandlungen wird Wissen gespeichert, welches sonst separat oder nur im Gedächtnis eines Nutzers vorhanden ist. So können andere Nutzer laufende Verhandlungen übernehmen, ohne aufgrund fehlenden Wissens über Zusammenhänge Fehlentscheidungen zu riskieren.

Dazu wurden alle Anforderungen in einem Module-Net spezifiziert, wie in Abbildung 4 dargestellt. Hierbei sind Aktionen und Entscheidungen als aktive Grundelemente von Prozessen einheitlich in das Modell integriert.

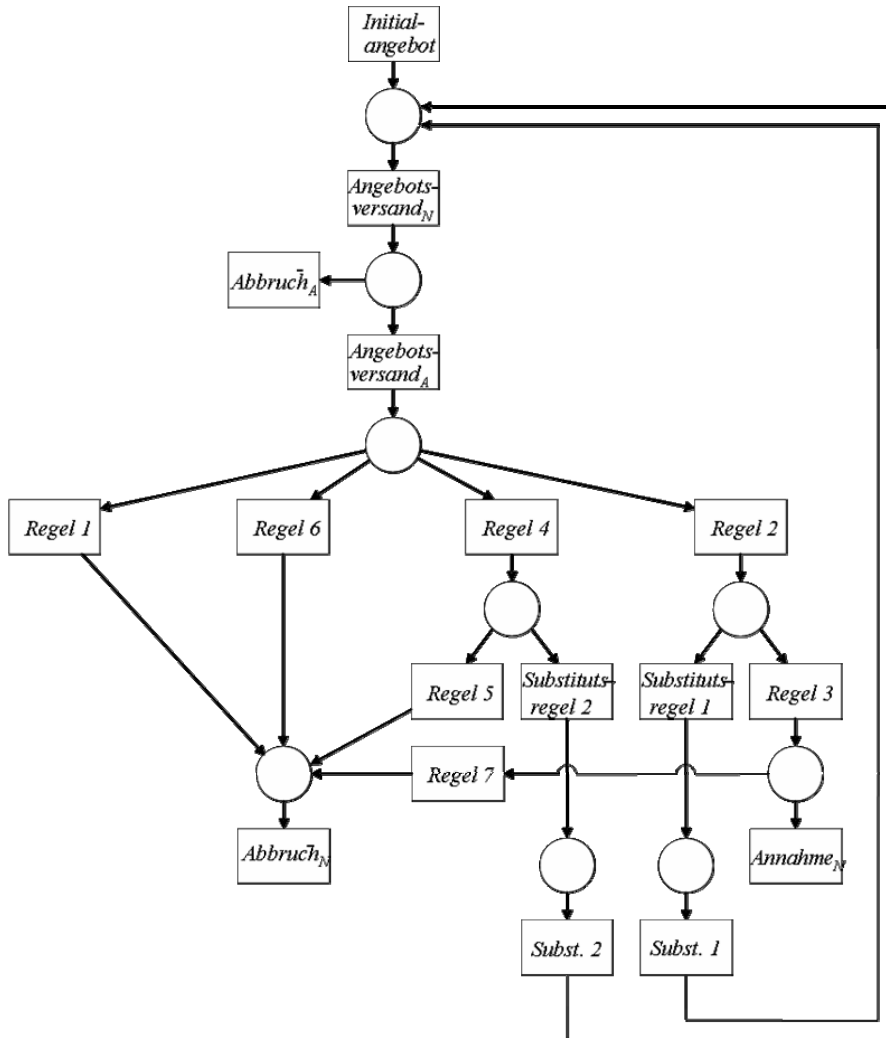


Abb. 4: Prozessmodell des Anfragers mit integrierter Entscheidungsfindung

Zu sehen ist hier der Verhandlungsprozess eines Nachfragers und seine Entscheidungsfindung, ausgedrückt als Regeln, die allerdings jederzeit vom Nutzer missachtet werden können. Der Prozess beginnt mit der Aktion *Initialangebot*, welches die Anfrage umfasst, und vom Nachfrager (*N*) in Form von *Angebotsversand_N* an den Anbieter (*A*) übermittelt wird. Hier ist modelliert, dass vom Anbieter keine sofortige Angebotsannahme erwartet wird, wie im Falle kleiner Abnahmemengen von Konsumgütern, sondern die Verhandlung wird entweder mit *Abbruch_A* beendet oder einem, vermutlich modifiziertem, Antwortangebot als *Angebotsversand_A* fortgesetzt. Dieses wird geprüft und entweder akzeptiert mit *Annahme_N*, die Verhandlung beendet mit *Abbruch_N* oder das Angebot wird wiederum modifiziert (*Subst. 1* und *Subst. 2*) und versendet mit *Angebotsversand_N*. Mit den verbleibenden Transitionen wird der Entscheidungsprozess des Nachfragers beschrieben. Dies sind *Angebotsprüfung_N* und *Angebotsmodifikation_N*, wie in Abbildung 3 gezeigt.

Im Falle multilateraler multiattributiver Verhandlungen hängen Entscheidungen von mehreren Einflussfaktoren ab. Eine Nutzwertfunktion kann angewendet werden, um festzustellen, ob ein Angebot akzeptabel oder inakzeptabel ist oder ob eine weiterführende Verhandlung sinnvoll erscheint [RFS05]. Die Entscheidungen der verschiedenen Regeln sind:

- *Regel 1* beendet die Verhandlung, wenn keines der angefragten oder angebotenen Güter oder Dienstleistungen ein Gegenstück seitens des Nachfragers aufweist. Diese Entscheidung ist unabhängig vom Status der Parallelverhandlungen.
- *Regel 2* klassifiziert, ob gemäß einer vorab implementierten Nutzwertfunktion ein Angebot akzeptabel ist oder gemäß *Regel 3* verbesserbar. Ist das Angebot akzeptabel, kann allerdings der Nutzwert geringer sein als der der Parallelverhandlungen, sodass gemäß *Regel 7* die Verhandlungen abzubrechen ist obwohl ein lokales Optimum erreicht wurde. Ist das Angebot ausbaufähig gemäß *Substitutsregel 1*, muss eine mögliche Verbesserung einen höheren Nutzwert aufweisen als den in den parallelen Verhandlungen ermittelten.
- *Regel 4* besagt, wann ein Angebot nicht akzeptabel ist, obwohl eine akzeptable Modifikation erreicht werden könnte. Eine definitive Ablehnung ergibt sich nach *Regel 5*, während *Substitutsregel 2* eine Verbesserung konkretisiert. Auch in diesem Fall sollte eine Verbesserung einen höheren Nutzwert aufweisen als alle Parallelverhandlungen.
- *Substitut 1* und *Substitut 2* spezifizieren die konkrete mögliche Modifizierung des Angebots. Dies basiert auf den früheren Ergebnissen von *Substitutsregel 1* und *Substitutsregel 2*.

Jede dieser hier natürlichsprachlich beschriebenen Regeln kann mit Hilfe eines High-Level-Petrinetzes und Prädikatenlogik spezifiziert werden [Si06]. Die Modelle sind also hinsichtlich ihrer datenbezogenen Präzisierung innerhalb der vorgestellten Modellierungsmethode skalierbar.

4.3 Realisierung im System für ein beispielhaftes Dienstleistungsszenario

Das Prozessmodell und weitere Spezifikationen bezüglich der Entscheidungsregeln werden in das Verhandlungsunterstützungssystem M2N implementiert. Die Nutzung der Kontrollmechanismen wird hier anhand eines Beispiels aus der Reisebranche gezeigt. Das hier anonymisiert vorgestellte Unternehmen arbeitet als Spezialreiseveranstalter für Eventdesign und –management sowie kundenindividuelle Reisen. M2N dient dabei nicht als Plattform zur Bündelung von Angeboten im Sinne eines Produktconfigurators, sondern der Verhandlung und damit Ausgestaltung von Events und Trips mit beliebigen Bestandteilen. Abbildung 5 zeigt den Prototyp der Übersichtsoberfläche.



Abb. 5: Übersichtsoberfläche zu multilateralen Verhandlungen

Angenommen, die Anfrage eines Reisenden für eine Reise in eine bestimmte Stadt umfasst die Verfügbarkeit und den Preis für ein Hotelzimmer in einem zentral gelegenen 4-Sterne-Hotel sowie die Bahnfahrt dorthin. Das Reisebüro fragt dann entsprechend mehrere Alternativen an. Nach Angebot an und Auswahl durch den Kunden kann das Reisebüro das gewählte Hotel fix reservieren und die anderen Möglichkeiten absagen. Dies entspricht Szenario 1 wie in Abbildung 1 gezeigt. Die Zusammenstellung eines individuellen Plans und Buchung der Events für Kunden wiederum kann von der Verfügbarkeit, der Gestaltungsmöglichkeiten und Preise einzelner Reisebestandteile abhängen. Hierbei entsteht ein Szenario wie unter 2 in Abbildung 1 dargestellt. Angenommen, für ein Incentiveevent einer Firma wird eine Reise in eine bestimmte Stadt zu einem bestimmten Datum geplant. Sollte das gewünschte Hotel ausgebucht sein, die gewünschte Lokalität nicht verfügbar, der zu buchende Gastredner anderweitig engagiert und die angefragte Speisefolge nicht realisierbar sein, würde der Kunde vielleicht andere Aktivitäten arrangieren lassen wollen, die Reise verschieben oder die ursprünglich gewählte Stadt ablehnen wollen. Dann sind entsprechend Änderungen oder Absagen bei den Leistungsträgern zu platzieren.

5 Verwandte Arbeiten

Im Bereich der Forschung zur Frage der Kontrolle interorganisationaler Prozesse gibt es verschiedene Vorschläge, wie Kollaborationen koordiniert und überwacht werden können. Technologiegebunden gibt es standardisierte Ansätze dazu, beispielsweise vom W3C für Web-Service-Choreographien. Für kontextunabhängigen Einsatz wird in [LS03] durch Einführung einer Prozesssicht von den Einzelprozessen abstrahiert unter Nutzung virtueller Zustände zur Überwachung kollaborativer Prozesse. In [EG08] wird dies um die Anpassbarkeit von Prozesssichten erweitert. [FSB07] beschreiben die Formalisierung von Workflows mit Hilfe von YAWL-Patterns zwecks Verifikation kollaborativer Prozesse. In [LV07] wird beschrieben, wie objekt-petritnetzbasierend in Wertschöpfungsnetzen Ablaufkoordination realisiert werden kann.

Diese Vorschläge sind generell für Geschäftsprozesse erarbeitet und bisher nicht auf Verhandlungsprozesse anwendbar, da im Gegensatz zu anderen überbetrieblichen Workflows hier jederzeitige menschliche Interaktion typisch ist. Business Rules geben zwar definierte Entscheidungsregeln vor, wobei allerdings umstände- oder fallweise menschliche Reaktionen nicht ausreichen fein strukturiert als strikte Vorgaben formulierbar sind. Obwohl es Untersuchungen zur Möglichkeit der Effizienzsteigerung durch elektronische Unterstützung in komplexen Wertschöpfungsketten gibt, wie beispielsweise [HFS06], gibt es in der Praxis wenig Realisierungen von Verhandlungsunterstützungssystemen [KL07]. Verhandlungsunterstützungssysteme sind meist auf die Entscheidungsunterstützung ihrer Nutzer oder den Prozessverlauf ausgerichtet. Ihr individuelles Design hängt dabei von ihrer Ausrichtung, den unterstützten Protokollen und eingesetzten Technologien ab. Ein umfassender Überblick findet sich in [KL07]. Einige der nicht-vollautomatisierten Systeme ermöglichen Verhandlungen mit mehreren Positionen und variablen Eigenschaften. In [Hü04] können konfigurierbare Produkte dargestellt werden, in [St00; Ba04; Ke04; Sc05] Attribute aus einer vorab erstellten Ontologie genutzt werden. Alle diese Systeme erlauben die Verfolgung von Änderungen über die Zeit und Statusbeobachtungen. Dabei sind allerdings Abhängigkeiten nur außerhalb der Systeme vermerk- und damit kontrollierbar. Bei [KS04] wird eine BPEL-basierte Architektur vorgeschlagen, um über einen elektronischen Marktplatz das Management laufender Prozesse zu ermöglichen, allerdings ohne diese bündeln zu können. In [BPS05] werden kombinatorische Auktionen zur Verhandlung von Produktbündeln als Alternative zur Führung paralleler bilateraler Verhandlungen vorgeschlagen. Keines der Systeme unterstützt allerdings eine formale Kontrolle zur Überwachung von Abhängigkeiten.

6 Diskussion und Ausblick

Im vorliegenden Beitrag wurde ein formaler Kontrollmechanismus basierend auf SPL vorgestellt und auf ein Verhandlungsunterstützungssystem angewendet. Es lassen sich formal und semantisch eindeutig Entscheidungsregeln implementieren, die jederzeit vom Nutzer überschrieben werden können. Dadurch werden Nutzer in die Lage versetzt, innerhalb des Systems mit dessen Unterstützung komplexe voneinander abhängige Prozesse zu überwachen und Abhängigkeiten zu kontrollieren.

Der vorgeschlagene Mechanismus ließe sich ebenso in vollautomatisierten Workflows anwenden, was im vorliegenden Fall aber nicht beabsichtigt ist. Eine Grundüberlegung beim Design von M2N war das Angebot, Nutzer bei der Entscheidungsfindung durch Informationslieferung zu unterstützen, nicht die automatisierte Übernahme der Entscheidung zu realisieren. Die Nutzer sollen unabhängig von vorgegebenen Workflows interagieren und letztendlich Entscheidungen treffen können, auch im Hinblick auf das Management der Kunden- und Lieferantenbeziehungen. Die Ausübung der Kontrolle darf keine negativen Auswirkungen auf sich abzeichnende langfristige Kollaborationsmöglichkeiten und entstehende Geschäftsbeziehungen haben. Die Beziehungen zwischen allen Geschäftspartnern dürfen nicht durch direkten automatisierten Versand vorformulierter Standardnachrichten beeinflusst werden, sondern bedürfen des Beziehungsmanagements durch menschliche Interaktion. Obwohl Verhandlungsunterstützungssysteme mit dem Ziel eingesetzt werden, die Beschaffungs- bzw. Vertriebskosten zu senken, darf dies nicht zulasten langfristiger Netzwerkbildung geschehen.

In Fortführung des Beschriebenen sollen die Implementierungsarbeiten abgeschlossen und die begonnene Evaluierung fertig gestellt werden. Weiterhin soll die Kontrollfunktion nicht nur für Prozesse anwendbar sein, sondern auf Positionsebene in den Verhandlungen ausgeweitet werden. Für die Beschaffung beliebig gestaltbarer Produkte ist Wissen über die Abhängigkeiten zwischen den Verfügbarkeiten spezifischer Bestandteile und Eigenschaften erforderlich. Die Bündelung multilateraler Verhandlungen nicht nur auf Basis ihres Prozessstatus, sondern basierend auf dem Status und den Eigenschaften der verhandelten Einzelpositionen, erlaubt die Kombination verschiedenster zusammenhängender nicht-standardisierter Sachgüter und Dienstleistungen unabhängig davon, welcher Anbieter in welchem Umfang liefern kann.

Eine solche umfassende Nutzerunterstützung auch in sehr komplexen Szenarien kann eine Möglichkeit für umfassendere elektronische Kollaboration bieten und langfristig als Koordinationsstelle die dynamische Bildung von Geschäftsbeziehungen und flexiblen Netzwerken unterstützen.

Literaturverzeichnis

- [Aa98] Aalst, W. M. P., van der: The Application of Petri Nets to Workflow Management. In: The Journal of Circuits, Systems and Computers 8(1), 1998, S. 21–66
- [AH02] Aalst, W. M. P., van der; Hee, K. van: Workflow Management – Models, Methods, and Systems. MIT Press, Cambridge, MA, 2002

- [An03] Andrews, T.; Curbera, F.; Dholakia, H.; Golland, Y.; Klein, J.; Leymann, F.; Liu, K.; Roller, D.; Smith, D.; Thatte, S.; Trickovic, I.; Weerawarana, S.: Business Process Execution Language for Web Services - Version 1.1., 2003. <http://www.ibm.com/developerworks/library/specification/ws-bpel/>, 01.07.2008
- [Ba04] Bartelt, A.: Elektronische Verhandlungen in verteilten E-Business-Systemen. Diss., Universität Hamburg, 2004
- [Bi05] Bichler, M.: A roadmap to auction-based negotiation protocols for electronic commerce. In: Proceedings of the 33rd Hawaii International Conference on System Sciences HICSS '00, vol. 6. IEEE Computer Society, Washington, DC, USA, 2000
- [BPS05] Bichler, M.; Pikovskiy, A.; Setzer, T.: Kombinatorische Auktionen in der betrieblichen Beschaffung. Eine Analyse grundlegender Entwurfsprobleme. In: Wirtschaftsinformatik 47 (2), 2005, S 126-134
- [BS97] Beam, C.; Segev, A.: Automated negotiations: a survey of the state of the art. In: Wirtschaftsinformatik 39 (3), 1997, S. 263–268
- [BW07] Brenner, W.; Wenger, R.: Anforderungen an Electronic Sourcing Systeme. In: Brenner, W.; Wenger, R. (Hrsg.): Elektronische Beschaffung. Springer, Berlin Heidelberg New York, 2007, S. 1-21
- [EG08] Eshuis, R.; Grefen, P.: Constructing customized process views. In: Data & Knowledge Engineering 64, 2008, S. 419–438
- [FSB07] Farrell, A.; Sergot, M.J.; Bartolini, C.: Formalising Workflow: A CCS-inspired Characterisation of the YAWL Workflow Patterns. In: Group Decision and Negotiation 16, 2007, S. 213-254
- [Ge07] Geiger, I.: Industrielle Verhandlungen. Universitäts-Verlag, Wiesbaden, 2007
- [GM98] Guttman, R. H.; Maes, P.: Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce. In: Klusch, M.; Weiß, G. (Hrsg.) Cooperative Information Agents II; Learning, Mobility and Electronic Commerce for Information Discovery of the Internet. Proceedings of the Second International Workshop CIA '98. LNCS 1435, Springer, Paris, 1998
- [He07] Herbst, U.: Präferenzmessung in industriellen Verhandlungen. Universitäts-Verlag, Wiesbaden, 2007
- [HFS06] Hausen, T.; Fritz, M.; Schiefer, G.: Potential of electronic trading in complex supply chains: An experimental study. In: International Journal of Production Economics 104, 2006, S. 580–597
- [HH05] Hofbauer, G.; Hellwig, C.: Professionelles Vertriebsmanagement. Der prozessorientierte Ansatz aus Anbieter- und Beschaffersicht. Publicis, Erlangen, 2005
- [Ho06] Hoffmann, A.: Interaktionen zwischen Anbietern und Nachfragern bei der Vermarktung und Beschaffung innovativer Dienstleistungen. Universitäts-Verlag, Wiesbaden, 2006
- [Hü04] Hümmer, W.: Vertragsverhandlungen um konfigurierbare Produkte im elektronischen Handel. Diss., Universität Erlangen-Nürnberg, 2004
- [Ke00] Kersten, G. E.; Noronha, S. J.; Teich, J.: Are All E-Commerce Negotiations Auctions?. In: Proceedings of the Fourth International Conference on Design of Cooperative Systems COOP'2000. Interneg Research Paper INR08/99, 2000
- [Ke04] Kersten, Gregory E. (2004): E-negotiation Systems: Interaction of People and Technologies to Resolve Conflicts. Interneg Research Paper INR08/04, 2004
- [KL07] Kersten, G.E.; Lai, H.: Negotiation Support and E-negotiation Systems: An Overview. In: Group Decision and Negotiation 16, 2007, S. 553–586
- [Ko07] Koppelman, U.: Internet und Beschaffung. In: Brenner, W.; Wenger, R. (Hrsg.): Elektronische Beschaffung. Springer, Berlin Heidelberg New York, 2007, S. 23-38
- [Krup07] Kruppe, E.: Prozessgestaltung – Grundlage und Mittel des Supply Chain Event Management. In: Ijioui, R.; Emmerich, H.; Ceyp, M. (Hrsg.): Supply Chain Event Management Konzepte, Prozesse, Erfolgsfaktoren und Praxisbeispiele. Physica,

- Berlin Heidelberg, 2007, S. 141-155
- [Krus07] Kruse, T.: Marktgerichtete Abstimmung in Unternehmen Bedeutung und Gestaltung der Schnittstelle von Absatz- und Beschaffungsmanagement. Universitäts-Verlag, Wiesbaden, 2007
- [KS04] Kim, J.; Segev, A.: A Web Services-enabled marketplace architecture for negotiation process management. In: *Decision Support Systems* 40, 2004, S. 71– 87
- [LS03] Liu, D.-R.; Shen, M.: Business-to-business workflow interoperability based on process-views. In: *Decision Support Systems* 38, 2003, S. 399– 419
- [LV07] Loos, P.; Vanderhaeghen, D. (Hrsg.): *Kollaboratives Prozessmanagement Unterstützung kooperations- und koordinationsintensiver Geschäftsprozesse am Beispiel des Bauwesens*. Logos, Berlin, 2007
- [MC94] Malone, T.W.; Crowston, K.: The Interdisciplinary Study of Coordination. In: *ACM Computing Surveys* 26 (1), 1994, S. 87-119
- [PRW98] Picot, A.; Reichwald, R.; Wigand, R.: *Die grenzenlose Unternehmung Information, Organisation und Management*, 3. Aufl. Gabler, Wiesbaden, 1998
- [Re01] Rebstock, M.: An Application Architecture for Supporting Interactive Bilateral Electronic Negotiations. In: Bauknecht, K.; Madria, S. K.; Pernul, G. (Hrsg.): *Electronic Commerce and Web Technologies. Proceedings of the EC-Web 2001*. Springer, Berlin Heidelberg New York, 2001, S. 196-205
- [RFP08] Rebstock, M.; Fengel, J.; Paulheim, H.: *Ontologies-Based Business Integration*. Springer, Berlin Heidelberg, 2008
- [RFS05] Rebstock, M.; Fengel, J.; Simon, C.: An Empirical Method for Utility Assessment in Real-World Electronic Negotiations. In: *Proceedings of Group Decision and Negotiation (GDN) 2005*, Wien, CD-ROM, 2005
- [Sa07] Saggau, B.: *Organisation elektronischer Beschaffung*. Universitäts-Verlag, Wiesbaden, 2007
- [Sc05] Schoop, M.: A Language-Action Approach to Electronic Negotiations. In: *Journal of Systems, Signs & Action* 1(1), 2005, S. 62-79
- [Sc06] Schmitz, C.: *Internationales Vertriebsmanagement für Industriegüter*. Universitäts-Verlag, Wiesbaden, 2006
- [Sc93] Schmidt, B. F.: Elektronische Märkte. In: *Wirtschaftsinformatik* 35 (5), 1993, S. 465-480
- [SFR07] Simon, C.; Fengel, J.; Rebstock, M.: Formal Control of Multilateral Negotiations. In: *Formal Control of Multilateral Negotiations. In: Proceedings of Group Decision and Negotiation (GDN) 2007*, vol. 2. Concordia University, Interneg, 2007, S. 371–381
- [Si06] Simon, C.: Integration of Planning and Production Processes. In: Troch, I.; Breitenacker F. (Hrsg.): *Proceedings of Mathmod 2006, Special Session "Petri nets: Current Research Topics and their Application in Traffic Safety and Automation Engineering. Series ARGESIM Reports No. 30*, Wien, 2006
- [Si08] Simon C.: *Negotiation Processes The Semantic Process Language and Applications*. Shaker, Aachen, 2008
- [SR04] Simon, C.; Rebstock, M.: Integration of Multi-attributed Negotiations within Business Processes. In: Desel, J.; Pernici, B.; Weske, M. (Hrsg.): *Proceedings of Business Process Management (BPM 2004)*, LNCS 3080. Springer, Berlin Heidelberg New York, 2004, S. 148-162
- [St00] Ströbel, M.: A Framework for Electronic Negotiations Based on Adjusted Winner Mediation. In: *Proceedings of the 11th International Workshop on Database and Expert Systems Applications (DEXA)*, IEEE Computer Society, Los Alamitos CA, 2000, S. 1020-1028

Process Modeling for Network Organizations – The Impact of the Process Environment

Stefan Große Böckmann*, Jan vom Brocke*, Kai Riemer[#], Daniel Richter*

* Martin Hilti Chair of Information Systems and Business Process Management,
Institute of Information Systems, University of Liechtenstein,
Hochschule Liechtenstein, Fürst-Franz-Josef-Straße
FL-9490 Vaduz, Fürstentum Liechtenstein
jan.vom.brocke@hochschule.li
{stefan.grosse-boeckmann,daniel.richter}@ercis.de

[#] European Research Center for Information Systems (ERCIS)
Universität Münster, Leonardo Campus 3
D-48149 Münster, Deutschland
wikari@wi.uni-muenster.de

Abstract: Attention to the specific characteristics of rather unstructured processes and the consequences for the modeling of these processes can only rarely be observed. This paper presents the results of a process analysis in a research network covering highly cooperative processes. The analysis shows that the network exhibits processes where conventional modeling methods fail to generate meaningful results. An in-depth analysis reveals that an optimized execution and an optimized outcome depend more on the environment of the processes than on an optimized sequence. Therefore it can be concluded that in order to achieve optimization it is fundamental to analyze the environmental factors.

1 Introduction

Although Keen and Scott Morton already differentiated between highly structured, semi-structured and unstructured processes in 1978 [KS78], until today there is only little attention to the specific characteristics of these types of business processes [Da07]. This is especially notable in the face of constant change of value adding structures towards more flexible and dynamic organizational structures [Or02]. The rapidly changing organizational environment forces companies, as well as scientific organizations, to cooperate in networks in order to access and develop the knowledge necessary to achieve flexible adjustments. Since the 1990s, business processes in organizations have been effectively analyzed and optimized using well-established Business Process Modeling (BPM) methods [Da93, HC93]. However, due to the increased process dynamic and flexibility as well as the emergent role of knowledge within the process design and for the processes itself, the analysis and design of business processes faces new challenges [DJB96, MMG02]. Therefore the suitability of traditional modeling methods in this changing context needs to be evaluated.

In this paper the authors present a process analysis that has been carried out in the research network GARNET (www.garnet-eu.org). The network covers structured as well as more unstructured processes. GARNET is a Network of Excellence (NoE) that is being funded by the European Commission (EC) within the 6th Framework Programme. It was the objective of the EC to strengthen research and technological development within the European Union and to increase the international competitiveness. The purpose of the network is to strengthen and develop community, scientific and technological excellence by means of integrating existing and emerging research activities and by exchanging knowledge. It aims to integrate the research capacities of the network partners and, at the same time, to advance knowledge within their topic domain [Eu03a]. GARNET is a science network addressing the phenomenon of globalization and regionalization. It comprises 42 leading research centers and universities from 17 European countries.

Within the GARNET network the authors of this paper were in charge of the development of a collaboration platform and its introduction into the network. The platform aims at supporting communication, coordination, and collaboration within knowledge creation processes. Carrying out a web usage analysis, we observed that the GARNET users did not use the platform in the intended way – which is mainly for collaborative purposes – but instead for coordinative and administrative tasks [BG08]. Driven by the results we subsequently carried out an analysis and optimization of the processes in the GARNET network. The results are presented in this paper. The analysis was directed to achieve a better integration of the platform with the network processes. The integration aimed to optimize the support of coordinative and administrative tasks as well as to foster virtual collaboration. One objective was to create a better environment for information exchange to initiate the emergence of new relationships. The idea was to create the essential social capital that guarantees knowledge exchange and a functional network [Ri05]. For further details about the GARNET network, the platform, and its analysis please refer to [RBRG08, BG08, BRR08].

The objective of this paper is to present the results of the process analysis and optimization, as well as a discussion thereof with respect to the problems that appeared during the modeling phase. We will show that the modeling worked well with administrative and coordinative processes, but due to a lack of structure it was not reasonable to model the core processes of the network. First, in section two we provide an introduction to the remainder of the paper with a presentation of the research context; we then present related work in section three. In section four we discuss the problems experienced during the modeling phase and present the results of a series of interviews that have been carried out afterwards in response to the problems. Based on the interview findings we then argue that it is the environment of the unstructured processes that fundamentally affects its outcome. We will conclude the paper with a short summary.

2 Research Design

2.1 Research Question

Subject of our research work is the management of research networks. Research networks aim to provide increased connectivity between the network partners in order to improve the dissemination of knowledge within the network [CCMV08]. In order to analyze the applicability of specific management methods in a network organization, in this paper we present the results of the application of BPM in a research network. BPM provides methods to analyze and design processes within the process management life-cycle [We07]. Through analysis and optimization of the processes, we aim at a better integration of technology with the working practices. Within this paper we want to focus on the limitations of traditional modeling methods. Therefore our research question is:

Which potentials and limitations characterize the application of traditional process modeling methods in research networks?

Our objective is to highlight two major findings. Firstly, we want to show that when conventional process modeling are applied in a dynamic work context, a significant part of the network's activities remain uncovered. Secondly, we want to point out that for the optimization of those processes that are left uncovered, in this context it is the process environment that plays a significant role. To substantiate these findings we will present the procedure and the results of a process analysis in the GARNET research network.

2.2 Research Design

As a preparation for the process analysis we developed a regulatory framework for the classification of the process models. Sources for the creation were publications of the European Commission within the 6th Framework Programme [Eu03b, Eu03a]. These documents structure activities in a research context and are therefore also representative for other types of research networks. To model the processes we picked Value Chain Diagrams on an abstract level and the Event-driven Process Chain (EPC) due to its wide acceptance in practice ([Da04]) on a detailed level. The EPC comprehends different views on a process: data, function, service, organization and process view. For details please refer to [Sc99]. The data used in the process modeling was collected in a series of interviews. The analysis was followed by an optimization that aimed at improving process efficiency through a better integration of collaboration systems. During the modeling phase we encountered problems modeling the core research processes due to their lack of structure. In order to better understand these problems, we applied a further set of modeling techniques which we identified from the literature (see section 3). Motivated by the problems we encountered and in order to better explore the factors that drive and determine the flow within this type of processes, we initiated a second series of interviews. The interview results show that the process environment is more important than the optimization of the temporal and logical chain of process activities for achieving intended network process outcomes.

The two interview series were open but guided [Kv96]. The guidelines have been slightly modified for every interview partner to take their role and position within GARNET into account. Face-to-face interviews were chosen where possible. Alternatively we arranged telephone conferences. As interview partners, we selected actors that play key roles within the network. Altogether 10 different GARNET members including the network manager, the PhD School manager, network and research group coordinators, as well as senior and junior researchers, have been consulted. The interview partners came from all organizational units that have been identified in the reference framework. They have been prepared in the run-up to the interviews with information on the intended topics. Some of them have been consulted in both interview series. Every interview lasted between 1 and 2.5 hours. A summary of the interview guidelines is shown in Table 1. It was in principle the same for the first and the second interview series. While the interviews in the first series concentrated on the sequence and characteristic of activities, those in the second focused on why specific activities were performed in a specified way and not in a way that might be more efficient or effective from a technical point of view.

Interview section	Questions Synopsis
Personal and Research Background	What is your research background? What kind and characteristics do other research projects you are also involved in have? If there are any, what kind and characteristics do other collaboration platforms you already used have?
GARNET	What's your position and role within the GARNET network? What is your attitude towards GARNET? What is your contribution to GARNET? What is the contribution of GARNET to your work? How cooperation in general does take place in GARNET? How do you characterize your working activities related with GARNET in detail? - Which tasks do you perform in which sequence, which documents, organizational units and technical tools are related with the tasks?
GARNET Collaboration Platform	How important is the platform for your personal work – what did you like and dislike? Do you have suggestions for improvements? Can you think of other situations where the platform might be of use in GARNET? Compared to other research projects, are there any particular advantages and disadvantages concerning the GARNET platform?

Table 1: Interview Guideline

3 Related Work

Our research focuses on methods for managing network organizations. Since the 1990s and based on the publications of Hammer, Champy and Davenport [Da93, HC93], Business Process Management and BPM are well established as management methods in research and practice (e.g. compare [BK03, Sc99]). However, our analysis within the research network revealed a specific type of process that cannot be modeled sensibly due to their inherent lack of structure. Therefore the following literature review focuses on publications that deal with semi- and unstructured processes. The review aims to characterize this type of processes and to also identify methods that describe approaches for handling these processes.

One central characteristic and at the same time the main reason for the weak structure of processes within GARNET is the impact of creativity. In the literature some theoretical contributions have looked into the characteristics of creativity (e.g. compare [Br89, DL02, FCG08, Sh00]). Creativity is an inherent part of scientific collaboration; research work quite often is innovation, it manifests as interaction between a person's thoughts and a socio-cultural context. The specific role of creativity within business processes has been analyzed by Seidel et al. [SMRB08, SRB08]. They introduce the concept of *pockets of creativity* to identify and describe creative parts of business processes. It is referred to these process parts as creative tasks. The pocket framework is based on four aspects of creativity identified by Rhodes: the creative product, the creative process, the creative person and the creative environment [Br89]. The creative product corresponds to the business process object; the creative persons are the actors within the process. The creative environment including creative tasks, creative persons and creative products is referred to as a pocket of creativity. A creative-intensive process is a single pocket of creativity or a business process that at least contains one pocket of creativity [SMRB08]. Creative tasks happen in a creative environment [Rh61]. They are characterized by a lack of predictability concerning their sequence within a process and the process outcome, by knowledge intensity, communication intensity and a high risk. Seidel et al. emphasize the complex interplay between various conditions that shape creativity-intensive processes.

Closely related to the impact of creativity on business processes is the impact of knowledge intensity. Domain-specific knowledge appears to be a necessary condition for creativity [FCG08]. Human beings need knowledge “to guide their actions to a successful outcome” [Cr90]. Hence, the application of knowledge is an important part of most business processes to some extent. Knowledge-intensity in processes often leads to higher complexity [DTHS05]. Knowledge and the processes affected from it are subject to knowledge management (KM), which is well established as a distinct research area. A KM strategy is derived from the overall business strategy. Within the KM context, in newer publications the concept of knowledge workers is often applied. Knowledge workers are those workers within organizations that are concerned with highly complex jobs. They need to analyze as well as to solve complex problems, to develop plans and to design products, services or processes [Ha07]. Therefore knowledge workers strive for as much flexibility and autonomy as possible [RRMA05]. Their work is characterized by spontaneity, communication-intensity and low predictability. The processes and their outcome are determined by high context variability and high action complexity [RRMA05]. The context of a process is shaped by the factors that influence the process execution (e.g. people, knowledge, culture, topic etc.). The high variability within the context makes it impossible to specify details of specific process steps. Complexity of actions concerns the process steps itself. A high variability of the steps makes it difficult to describe the process as a whole including all execution alternatives. High variability in general influences the ability to plan a process.

Within the KM research area different approaches have been developed that aim at an integration of specific knowledge-related elements into existing or new modeling languages. Remus for example presents *blueprint* whose development was guided by the idea of reference modeling [RS03]. Blueprint consists of a procedure model and a conceptual model. In the conceptual model all KM activities are described that support the knowledge life cycle. The procedure model is based on the conceptual model and contains an activity set necessary to prepare a business process for KM and to integrate the business process in the enterprise-wide knowledge management concept. The conceptual model provides reference processes for knowledge management. Papavassiliou et al. present a concept that is based on the idea of specific knowledge objects [PM03]. A knowledge object represents the explicit knowledge required in a specific business process. In order to model knowledge-intensive processes, they differentiate between standard tasks and knowledge management tasks (KM tasks). KM tasks describe work associated with the generation, storage, application and distribution of knowledge in the business processes. Though knowledge-intensive processes are defined as weakly structured within the concept, no special conclusions for the modeling of processes are drawn at all. The concept of Remus as well as that of Papavassiliou is partly based on the EPC. In contrast, Gronau et al. introduce a proprietary modeling language for knowledge-intensive business processes [GH06]. The KMDL enables the modeler to add detailed information describing the transformation of a business process object within a specific process task. It provides elements to model the task input, the task output, and information- and knowledge-transformation flows. Figure 1 shows a simple KMDL example (the creation of a publication) from the activity view. The activity view extends the process view and visualizes the transformation of knowledge objects.

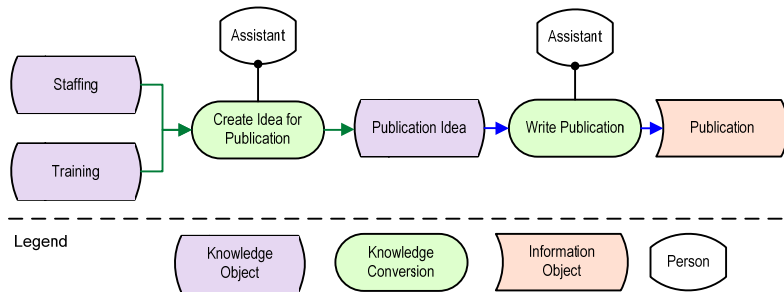


Figure 1: KMDL example (compare [GH06])

Some more work that especially focuses on a theoretical foundation has been carried out by Dalmaris and Markus. Dalmaris presents a framework for the improvement of knowledge-intensive processes [Da07]. It consists of three components: an epistemological foundation, a business process ontology, and an improvement methodology. The business process ontology is used for the capturing of data on those key components of the business process that are critical for the improvement effort. The improvement methodology provides concrete steps for a business process improvement taking the ontology into account. The process analysis is applied on the process level as well as on the function level. Markus et al developed a set of principles for the design of organizations based on a design theory [MMG02]. The principles focus on the design of IT systems to support enterprise decision support systems. The principles are shaped apart from the usage of specific modeling methods for the design of business processes.

Summing up the literature review, we conclude that especially knowledge-intensity and creativity lead to processes with a lack of structure. Flexibility and autonomy with regard to process flow are a typical (and necessary) characterization of processes whose output is characterized by a high level of knowledge-impact and creativity. Furthermore we conclude that whereas the analysis of the characteristics of more unstructured processes and the development of new modeling approaches has gained some interest in the literature, an analysis of conventional process modeling techniques for the improvement of collaborative processes can only rarely be found.

4 Discussion of the results of process modeling in research networks

4.1 Discussion of the experiences of applying conventional modeling to collaborative processes

Our process optimization project led to a detailed understanding of the functionality needed to support the research network from a technical point of view. The results show that the collaboration platform is well positioned to meet the core needs of the network but that it also needs some enhancements for an efficient integration into the existing processes. A central document repository including version management and a dedicated rights management is indispensable to overcome the regional fragmentation of the GARNET network. This is especially true when organizational units that consist of representatives of different partner institutions (e.g. the Project Management Committee) work together. The document repository also functions as an archive and as a base for the preparation of information dissemination on the website. The second component is a member and expert database comprising detailed profiles and social network browsing functions. Social browsing facilitates the creation of social networks and increases the awareness of other members and research groups [Ri05]. This is especially important for the functioning of the network, which depends on an appropriate density of social ties. Therefore not only the members should be linked with oneself, but also the members with other objects on the platform they are related to. In addition, the expert database should also allow the inclusion of external contacts. Also important is an Email-based automatic notification system that informs about ongoing changes and thus increases workspace awareness on the platform.

The regulatory process framework differentiates the processes into core and support processes. Core processes create value for the network. According to the European Commission, this applies to knowledge creation as well as to the integration of the partners' resources and competences [Eu03a]. Furthermore, the core processes can be separated into integrating activities, spreading excellence activities, jointly executed research activities and management activities. The support processes do not directly provide knowledge creation or spreading of excellence but facilitate the core processes. Our process modeling revealed one elemental perception very early on: while it was relatively easy to model processes in the context of integrating, spreading excellence, and with regard to management activities, modeling of the jointly executed research processes was only possible to a certain extent. Within this area we can differentiate between coordinative processes, the modeling of which is again relatively easy (e.g. preparation, execution and wrap-up of workshops or reporting), and the core research processes that could not be sufficiently modeled (e.g. publication creation or collaborative research).

In order to develop a more detailed understanding concerning the nature of our modeling problem, we tried to apply other knowledge-related modeling methods on the case. Papavassiliou et al. provide a workflow-oriented model that focuses on modeling objects and typical knowledge management tasks to transform these objects [PM03, PNAM03]. A typical knowledge object in our case may be a publication. Typical tasks (generation, storage, distribution and knowledge application [PM03]) could also be identified. A problem was the decomposition of the tasks into subtasks and the concatenation of the tasks to processes. It was impossible for the interview partners to specify a sequence of the tasks. In principle they could be executed in random even if they are loosely coupled. Remus et al. alternatively provide a three step approach for the modeling of knowledge-specific processes and integration into the value chain processes [RS03]. In the first step reference knowledge processes are provided that need to be adapted and integrated into the business processes. In the second step the processes are further decomposed into EPC knowledge chains. The reference processes have been deduced from KM methods and instruments and therefore are more comprehensive than necessary in the network context. But again the major problem is the concatenation of the processes as well as the decomposition into tasks and their concatenation. The KMDL as a modeling method especially for knowledge-intensive processes [GH06] provides a more detailed knowledge modeling approach in combination with a more abstract task description. It facilitates the modeling of knowledge flows and of specific knowledge transformation types. Due to the more abstract nature of the tasks it was possible to assign knowledge objects to the tasks and to describe how they are transformed within the tasks. But due to the more abstract nature it was also impossible to assign specific information system features or specific network roles to specific tasks. A great variance depending on the work context of the interviewed person made it impossible to define clear solutions for a specific situation using the KMDL.

Even if it was not possible to model straight sequences, it was possible to identify typical phases that affect the processing of the concerned knowledge object. The succession of the phases follows typical patterns. However, the patterns are not necessarily applied. The application depends on the research context, i.e. the country, the research group or the cultural background. These phases are comparable to knowledge-related tasks (e.g. those identified from Remus or Papavassiliou), but in comparison to EPC activities they are more vague, abstract and less formalized. Especially the modeling of an optimized chain is not possible. An approach for a definition of specific phases for example provide Shneiderman, who specifies four phases and eight activities describing creative processes [Sh00], and Papavassiliou [PM03]. The formulation of phases may help to analyze rather unstructured processes. Within every phase an analysis of related persons, knowledge object and application systems can lead to findings that facilitate the optimization of the working patterns.

4.2 Discussion of environmental factors that influence the collaborative processes

In order to analyze these phases and the cultural environment that affects them, we conducted a second interview series in the GARNET network. The interviews revealed some more qualitative and cultural aspects that affect the efficiency and functioning of the processes. One issue for example is that often senior researchers avoid using the platform by instructing junior researchers to execute the platform-dependent work. Junior researchers however are generally only poorly connected within the network [RBRG08]. In a network context that depends heavily on the creation of social capital, this is especially problematic, because the ability of the senior researchers to create a denser network structure cannot become effective [RBRG08]. Another issue is that most researchers seem not to hold much interest in collaborative work because single author publications are much more appreciated within the GARNET network. Furthermore, applying for mobility funding in many cases is motivated more by financial than by collaborative aspects. These cultural framework conditions highlight the importance of the environment in which the collaboration is taking place. It shows that the creation of a cooperative culture that facilitates a creative collaboration is important. A third issue stems from the observation that we detected some phases of intensified platform usage that exist outside the typical usage processes. An example is an increased usage before or after social events, because of the need to inform about the event or the attending participants. It is especially important to support these phases through appropriate process and platform design because they are essential for the buildup of social relationships that last longer than the one conference event. In this context it is notable that several interviewees explicitly pointed to the importance of trust as basis for effective collaboration.

The platform facilitates several features that are typically used within a Web 2.0 context and therefore act as tools for collaborative online content creation. However, our optimization project has shown that these features are not always used on the platform in an efficient way and that improvements are possible for some processes. A corresponding discussion in the interviews showed that network members knew about these features but did not know how to use them in their personal and well-established workflows. Hence the flexibility of the platform is on the one hand essential for an appropriate usage because of the rather unstructured usage context, on the other hand however this also acts as a barrier to its adoption. Additionally it was pointed out that using the platform was often perceived as extra workload, because parallel work inside and outside the platform was necessary to deal with specific problems (e.g. providing event information). To solve this problem, a better technical and organizational integration of the dissemination channels is necessary. On the one hand, an integrated system might provide data-centered workflow patterns (e. g. for the preparation, execution and wrap-up of conferences and seminars). On the other hand, these workflows should be adaptable to meet the needs and customs of specific user groups and in specific situations. Solutions for this can be found in the Workflow Management Systems research area (e.g. compare [ATEA06, ATPS08, AWG05]).

Summing up the observations, we can identify two different problem areas that prohibit the modeling of an optimized process flow. On the one hand, we observe a situation-specific variance of the process execution depending on a set of weak factors that cannot be identified when modeling tasks and their sequences. It can be differentiated between cultural, organizational and technical factors. Cultural factors like social relationships, the culture of collaboration and established work practices determine the flow of activities and the tools being used on a micro level. These are for example dependent on the country or the research team. Organizational factors like different organizational structures or competing business and external rules caused by a country- and institution-crossing networks lead to a divergent assignment of persons and roles to tasks within specific situations and in specific work teams. The design and availability of information systems influence the way and by whom they are used. These factors shape the environment in which the process execution takes place. On the other hand the working practice itself is affected by a need for flexibility because of a high impact of creativity and knowledge-intensity. The environmental factors need to be formed in a way that facilitates creativity and flexibility in order to achieve the intended process outcome.

4.3 Conclusions for further research

This paper focuses on processes in a highly dynamic and unstructured research context. Even though the literature review indicates that a common consensus is missing as to how such a type of business processes can be characterized, there is evidence that a special class of processes exists that must be modeled and analyzed in a specific way. The descriptions for defining this type of processes vary from “unstructured” [PM03], “knowledge-intensive” [Da07, ESR99, RS03], “with a high task complexity” [Da07], “emerging” and “unpredictable” [MMG02], “weakly-structured” because of a lack of formality [PM03] to “creative” [DL02, FCG08, SMRB08, SRB08, Sh00]. The variance and the lack of a commonly accepted definition point to a need for developing a common understanding in future research. The process as the object of research must be further characterized. The characterization should aim at a separation from those processes that can be modeled in a conventional way. There is also a need to evaluate, whether or not different classes of unstructured processes exist in different organizational contexts with different profiles.

Furthermore, the factors that determine the efficiency and effectiveness of this type of processes need to be determined. Since traditional process modeling methods lack the means to cope with them, the integration of other or the development of new methods appears necessary. The identification of typical phases shows that modeling on a high abstraction level is possible, whereas the optimized outcome with optimized resource consumption is more dependent on the environment of the processes than on an optimization of the activity chain with respect to time and sequence. The publications from Seidel [SMRB08, SRB08], the work concerning creativity in general [Br89, DL02, FCG08] and the methodological work from Dalmaris [Da07] and Markus [MMG02] provide an appropriate grounding for this. Additionally, the complexity of these factors raises the question if traditional diagram-oriented modeling is an appropriate approach in this context at all. Furthermore, it needs to be analyzed, which of the factors can be pre-determined in the optimization phase and which might change during execution and therefore need flexibly designed solutions. In order to further shape the understanding of process environment, a multidisciplinary approach integrating findings from the Knowledge Management and Computer Supported Cooperative Work research domains might be considered.

Finally, it must be taken into account that these observations are based on processes in a highly collaborative research network context. Therefore, the portability to other contexts needs to be examined. However, the literature review has shown that a high impact of creativity and knowledge generates processes of high complexity and with a lack of predictability. Hence, we can conclude that these observations might also hold true in other contexts. Furthermore, we only regarded the EPC as modeling method and it needs to be evaluated if these findings are also valid for other methods (e.g. BPMN). However, the similar structure of conventional modeling methods as all being chain-oriented indicates transferability of the findings.

5 Summary

The goal of business process reengineering is the identification and elimination of inefficiencies in the flow of activities of specific processes [BK03]. Hence an optimal sequence of activities with respect to time and logic needs to be designed. But for unstructured processes, which are often affected by a creative outcome or by a situation-specific flexibility, what matters more in the optimization is the quality of the results instead of an optimized sequence of activities. Therefore not the optimization of the task chain is to be focused but the optimization of the process environment in order to create a context that enables the creation of an optimized process outcome. This leads to a fundamental change of the perception of business process reengineering for this type of processes.

Based on the analysis of the process optimization and interview results in this paper we are able to draw two main conclusions for the analysis and optimization of unstructured processes. Firstly, the processes are not structured by a chain of micro activities but by phases that tend to be more abstract and less formalized. Secondly, because of the lack of structure, we need to integrate new or other approaches that facilitate the analysis and optimization of the environment of the process instead of its activity sequence. Due to the parallel and overlapping existence of highly structured and more or less unstructured processes in organizations, these approaches need to be coupled with conventional methods or may extend them. Further research is also necessary in order to analyze the characteristics of such a process environment in more detail.

References

- [ATEA06] Adams, M. et.al.: Worklets: A Service-Oriented Implementation of Dynamic Flexibility in Workflows. In: Proc. 14th International Conference on Cooperative Information Systems, Montpellier, France, 2006; pp. 291-308.
- [ATPS08] van der Aalst, W. et.al.: Flexibility as a Service. In: BPM Center Report BPM-08-09, 2008.
- [AWG05] van der Aalst, W. M. P.; Weske, M.; Grünbauer, D.: Case handling: a new paradigm for business process support, In: Data & Knowledge Engineering. Vol. 2005, No. 53 2005; pp. 129-162.
- [BG08] vom Brocke, J.; Große Böckmann, S.: Adaption von Kollaborationssystemen in Forschungsnetzwerken - Ergebnisse einer Web Usage Mining-Studie im Network of Excellence GARNET. In: Proc. Multikonferenz Wirtschaftsinformatik (MKWI), München, 2008.
- [BK03] Becker, J.; Kahn, D.: The Process in Focus. In (Becker, J.; Kugeler, M.; Rosemann, M., eds.): Process Management. A Guide for the Design of Business Processes, Berlin, 2003.
- [Br89] Brown, R. T.: Creativity - What Are We to Measure? In (Glover, J. A.; Ronning, R. R.; Reynolds, C. R., eds.): Handbook of Creativity. Perspectives on Individual Differences, New York, 1989; pp. 3-32.
- [BRR08] vom Brocke, J.; Riemer, K.; Richter, D.: Zur Rolle von Kooperations-systemen in verteilten Forschungsnetzen - Ergebnisse einer Social Network Analysis im Network of Excellence GARNET. In: Proc. Multikonferenz Wirtschaftsinformatik (MKWI), München, 2008.
- [CCMV08] Cassi, L. et.al.: Research Networks as Infrastructure for Knowledge Diffusion in European Regions, In: Economics of Innovation & New Technology. Vol. 17, No. 7 2008; pp. 665-678.
- [Cr90] Craig, E., Knowledge and the State of Nature, Oxford, 1990.
- [Da04] Davies, I. et.al.: How do practitioners use conceptual modeling in practice?, In: Data & Knowledge Engineering. Vol. 58, No. 3 2004; pp. 358-80.
- [Da07] Dalmaris, P. et.al.: A Framework for the improvement of knowledge-intensive business processes, In: Business Process Management Journal. Vol. 13, No. 2 2007; pp. 279-305.
- [Da93] Davenport, T. H., Process innovation: reengineering work through information technology, Boston, MA, 1993.

- [DJB96] Davenport, T. H.; Jarvenpaa, S.; Beers, M.: Improving Knowledge Work Processes, In: Sloan Management Review. Vol. 37, No. 4 1996; pp. 53-65.
- [DL02] DeGraff, J.; Lawrence, K. A., Creativity at Work. Developing the Right Practices to Make Innovation Happen, New York, 2002.
- [DTHS05] Dalmaris, P. et.al.: A Framework for the improvement of knowledge-intensive business processes, In: Business Process Management Journal. Vol. 13, No. 2 2005; pp. 279-305.
- [ESR99] Eppler, M.; Seifried, P. M.; Röpneck, A.: Improving Knowledge Intensive Processes through an Enterprise Knowledge Medium. In: Proc. Special Interest Group on Computer Personnel Research Annual Conference (SIGCPR), New Orleans, 1999; pp. 222-230.
- [Eu03a] European Commission, FP 6 Instruments Task Force 2003: Provisions for implementing Networks of Excellence. 2003.
- [Eu03b] European Commission, Classification of the FP 6 Instruments. Detailed description. 2003.
- [FCG08] Farooq, U.; Carroll, J. M.; Ganoë, C. H.: Designing for Creativity in Computer-Supported Cooperative Work, In: International Journal of E-Collaboration. Vol. 4, No. 4 2008.
- [GH06] Gronau, N.; Hasselbring, W., M-WISE: Modellierung wissensintensiver Prozesse im Software Engineering, Berlin, 2006.
- [Ha07] Harmon, P., Business Process Change: A Guide for Business Managers and BPM and Six Sigma Professionals (Second Edition), Burlington, MA, USA, 2007.
- [HC93] Hammer, M.; Champy, J., Reengineering the corporation: a manifesto for business revolution, New York, NY, 1993.
- [KS78] Keen, P. G.; Scott Morton, M., Decision support systems: An organizational perspective, 1978.
- [Kv96] Kvale, S., Interviews. An Introduction to Qualitative Research Interviewing, London, New Delhi, 1996.
- [MMG02] Markus, M. L.; Majchrzak, A.; Gasser, L.: A Design Theory for Systems that support Emergent Knowledge Processes, In: MIS Quarterly. Vol. 26, No. 3 2002; pp. 179-212.
- [Or02] Orlikowski, W. J.: Knowing in Practice: Enacting a collective capability in distributed organizing, In: Organization Science. Vol. 13, No. 3 2002; pp. 249-273.
- [PM03] Papavassiliou, G.; Mentzas, G.: Knowledge modelling in weakly-structured business processes, In: Journal of Knowledge Management. Vol. 7, No. 2 2003; pp. 18-33.
- [PNAM03] Papavassiliou, G. et.al.: Supporting Knowledge-Intensive Work in Public Administration Processes, In: Knowledge and Process Management. Vol. 10, No. 3 2003; pp. 164-174.
- [RBRG08] Riemer, K. et.al.: Cooperation Systems in Research Networks - Case Evidence of Network (Mis)fit and Adoption Challenges. In: Proc. 16th European Conference on Information Systems, Galway, 2008.
- [Rh61] Rhodes, M.: An analysis of creativity. In: Phi Delta Kappan, 1961; pp. 305-310.
- [Ri05] Riemer, K., Sozialkapital und Kooperation, Tübingen, 2005.

- [RRMA05] Riss, U. V. et.al.: Challenges for Business Process and Task Management, In: Journal of Universal Knowledge Management. Vol. 0, No. 2 2005; pp. 77-100.
- [RS03] Remus, U.; Schub, S.: A Blueprint for the Implementation of Process-oriented Knowledge Management, In: Knowledge and Process Management. Vol. 10, No. 4 2003; pp. 237-253.
- [Sc99] Scheer, A.-W., ARIS - Business Process Modeling, Berlin, Heidelberg, New York, 1999.
- [Sh00] Shneiderman, B.: Creating Creativity: User Interfaces for Supporting Innovation, In: ACM Transactions on Computer-Human Interaction. Vol. 7, No. 1 2000; pp. 114-138.
- [SMRB08] Seidel, S. et.al.: A Conceptual Framework for Information Retrieval to support Creativity in Business Processes. In: Proc. 16th European Conference on Information Systems (ECIS 2008), Galway, 2008.
- [SRB08] Seidel, S.; Rosemann, M.; Becker, J.: How does Creativity impact Business Processes? In: Proc. 16th European Conference on Information Systems (ECIS 2008), Galway, 2008.
- [We07] Weske, M., Business Process Management: Concepts, Languages, Architectures, Berlin, Heidelberg, 2007.

Generische Einschränkung der Modellierungsfreiheit in fachkonzeptuellen Modellen

Martin Jührisch¹, Gunnar Dietz², Werner Esswein³

¹Westfälische Wilhelms-Universität Münster
Projekt MIRO
Röntgenstr. 9-13
48149 Münster
juhrisch@uni-muenster.de

²Universität Hamburg
Projekt eCampus II
Schlüterstr. 70
20146 Hamburg
gunnar.dietz@uni-hamburg.de

³Technische Universität Dresden
Lehrstuhl für Wirtschaftsinformatik,
insbes. Systementwicklung
01062 Dresden
werner.esswein@tu-dresden.de

Abstract: Die zunehmende Verwendung von Informationsmodellen über technische Softwareaspekte hinaus, die ansteigende Anzahl an Modellerstellern sowie die Forderung nach Modellvergleichbarkeit bzw. Modellbewertbarkeit machen eine an Konventionen orientierte konstruktive Einschränkung der Freiheitsgrade bei der fachkonzeptuellen Modellierung notwendig. In diesem Artikel wird eine generische Methode vorgestellt, welche durch die Einführung von „Beschreibungsrahmen“ (Description Kits) sowohl eine Einschränkung der Modellierungsfreiheit in Bezug auf natürlichsprachliche Aspekte in fachkonzeptuellen Modellen erlaubt, als auch eine restriktive Benutzung bestehender Modellierungssprachen ermöglicht. Als Anwendungsfall wird die Konfiguration service-orientierter Architekturen diskutiert und die Nützlichkeit des Ansatzes in einer Pilotstudie illustriert, durchgeführt im Projekt MIRO an der Universität Münster.

1 Einleitung

In der Fachliteratur wird die Einschränkung der Modellierungsfreiheit insbesondere unter dem Aspekt der Modellintegration bei verteilten Modellierungsprojekten diskutiert [Be07]. Im Rahmen der konzeptuellen Modellierung werden Unternehmen systematisch analysiert und mit Hilfe semiformaler Modellierungssprachen rekonstruiert [Be95]. Der hohe Freiheitsgrad bei der Modellierung und die fehlende Standardisierung von Modellelementen in semiformalen Modellierungssprachen führen zu einer Reihe von Integrationskonflikten bei der Zusammenführung fachkonzeptueller Modelle [Pf07].

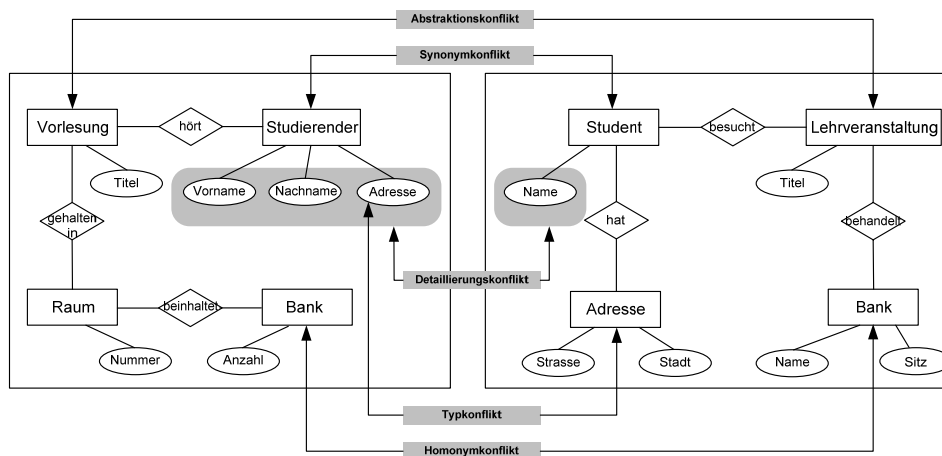


Abb. 1: Verschiedene Arten von Integrationskonflikten (in Anlehnung an [Pf07])

Modellübergreifende Referenzen können z. B. durch unterschiedliche Begriffssysteme nicht aufgelöst und Überlappungsbereiche zwischen den Teilmodellen durch unterschiedliche Abstraktionsgrade nicht identifiziert werden. Zur Vermeidung der Konflikte und zur Herstellung der Vergleichbarkeit von Modellen existieren nun zwei Ansätze: Zum einen wenden Ansätze zur Modelltransformation syntaktische Transformationen sowie semantische Tests an, um Überlappungsbereiche zu identifizieren [Ge07]. Diese Ansätze stellen nur minimale Annahmen an das vorliegende Modell. Zum anderen wird bei der Konstruktion der Modelle angesetzt, um die Konstruktion einfach vergleichbarer Modellierungsartefakte zu begünstigen. Ansätze dieses Bereichs basieren auf der Annahme, dass der Vergleich von Modellen mit willkürlicher Struktur die Identifikation von semantischen Überlappungsbereichen behindert. Becker et al. halten fest, dass, um in einem verteilten Modellierungsprojekt vergleichbare Resultate zu erhalten, die Freiheitsgrade des Modellierers stark eingeschränkt werden müssen [Be07]. Dies bedeutet, dass bei Vorliegen eines identischen Sachverhalts verschiedene Personen das gleiche Sprachkonstrukt zur Beschreibung nutzen sollten [Be95].

Basishypothese ist damit die Annahme, dass Einschränkungen bei der Modellierung den späteren Vergleich der Teilmodelle merklich vereinfachen [Pf07]. Als Konsequenz schränken Konventionen die Freiheitsgrade bei der Modellierung ein, um ein bestimmtes Maß an Compliance zu garantieren [Ro03]. Empirische Untersuchungen zeigen, dass in einem verteilten Modellierungsprozess erstellte Teilmodelle sich in Bezug auf das genutzte Vokabular, den Grad der Abstraktion und den Detaillierungsgrad unterscheiden [HS06]. Die Einschränkung der Modellierungsfreiheit für den Zweck der Schaffung vergleichbarer Teilmodelle betrifft damit im Kern die Eliminierung von Synonymen und die Schaffung semantisch disjunkter Sprachkonstrukte [Pf07]. Die Forderung nach semantisch disjunkten Sprachkonstrukten bewahrt vor Abstraktionskonflikten – alle verteilten Modelle teilen dann die gleiche Ebene der Abstraktion.

Die Modellbewertbarkeit stellt den zweiten Grund zur Einschränkung der Modellierungsfreiheit dar. Zum einen werden angesichts der zu erwartenden Fülle von Modellen innerhalb einer Prozesslandschaft manuelle Auswertungen stark erschwert. Zum anderen dienen konzeptuelle Modelle präskriptiv als Entwurf für eine zu implementierende Soll-Struktur der Organisation und ihrem Informationssystem. Die Herausforderung besteht darin, ein umfassendes Modellsystem zu erstellen, welches zum Zeitpunkt der Fertigstellung Gültigkeit besitzt – also auf aktuelle Marktanforderungen reagiert. Daher ist es sinnvoll, auf Grundlage der erhobenen Prozesslandschaft teilautomatisierte Analysen und Auswertungen als Indikatoren für zukünftige Reorganisationsmaßnahmen durchführen zu können bzw. konzeptuelle Modelle so vorzubereiten, dass sie als Input für einen Transformationsprozess in Entwurfsmodelle dienen können. Für die Auswertbarkeit ist die o. g. Forderung nach Vergleichbarkeit eine notwendige Voraussetzung [Pf07]. Zusätzlich müssen die Sprachkonstrukte eine geeignete Abstraktionsstufe und Domänennähe besitzen, um inhaltlich auswertbar zu sein.

In der vorliegenden Arbeit soll ein eigener Ansatz präsentiert werden, der die Entwicklung domänenspezifischer Sprachkonstrukte und deren eingeschränkte Nutzung in konzeptuellen Modellen generisch erlaubt. Die Methode hat ihren Ursprung in der Anpassung und Optimierung der service-orientierten Architektur an der Universität Münster [BHT07]. Organisatorische Soll-Modelle resp. Anforderungsmodelle sollen dabei die Rolle erfüllen, Input in eine Übersetzung in Servicekompositionen zu sein.

Der Artikel ist wie folgt strukturiert: Der nächste Abschnitt führt kurz die Motivation für eine Einschränkung der Freiheitsgrade bei der konzeptuellen Modellierung ein. Im dritten Abschnitt wird eine eigene Methode vorgestellt, welche die Grundlage für die generische Einschränkung der Modellierungsfreiheit in Modellen darstellt. Der Artikel schließt mit einem Anwendungsbeispiel zur Konfiguration von service-orientierten Architekturen und fasst im Ausblick wesentliche Ideen und offene Punkte zusammen.

2 Einschränkung der Modellierungsfreiheit

Die Essenz der Einschränkung der Modellierungsfreiheit in Bezug auf die Modellierung bestimmter Aspekte im Informationsmodell ist die Begrenzung des Sprachvokabulars auf eine Menge domänenspezifischer, semantisch disjunkter Sprachkonstrukte. Damit hat nicht nur das konstruierte Informationsmodell, sondern auch bereits die Modellierungssprache eine semantische Verbindung mit der Anwendungsdomäne [Pf07]. Aus Anwendungssicht semantisch bedeutende Operationen auf konzeptuellen Modellen können damit schon auf Sprachebene definiert werden.

Als Beispiel lässt sich die spezifische Problemsituation im Fall der verteilten Prozessaufnahme in öffentlichen Verwaltungen heranziehen. [Be07] stellen mit PICTURE eine Methode vor, mit der die Modellierungsfreiheit auf eine Menge an domänenspezifischen Sprachkonstrukten eingeschränkt wird. Durch die Eliminierung von Typ, Synonym, Homonym und Abstraktions-Konflikten kann der semantische Modellvergleich auf die Ebene der Syntax zurückgeführt werden [Pf07]. Flankiert wird die Einschränkung der zur Verfügung stehenden Fachsprache durch eine spezielle Prozessmodellierungssprache mit einfacher Syntax.

Da die Sprachkonstrukte aus der Fachsprache abgeleitet werden, sind sich Fachexperten in ihrer Rolle als Modellierer über deren Semantik bewusst. Die Wahl unpassender Modellierungskonstrukte wird vermieden, die Operationalisierbarkeit der Modelle ermöglicht.

Eine Fachsprache ist allerdings immer natürlich gewachsen, ihr Vokabular und ihre Grammatik sind nicht statisch, sondern Resultat des Sprachgebrauchs in einer Sprachgemeinschaft. Die Festlegung auf domänenspezifische Sprachkonstrukte auf Metamodellebene erscheint vor dem Hintergrund des dynamischen Wandels problematisch. Darüber hinaus besitzen domänenspezifische Modellierungssprachen keine ausreichende Anzahl an Sprachkonstrukten, um alle Phänomene in der jeweiligen Domäne abzubilden. Folgerichtig wird eine Vielzahl von domänenspezifischen Modellierungssprachen benötigt. Der hier vorgestellte Ansatz widmet sich der Entwicklung situationsabhängig adaptierbarer fachlicher Sprachkonstrukte, die für ein Projekt in einer bestimmten betrieblichen Domäne möglicherweise auch nur für einen kurzen Zeitabschnitt konstruiert und im Zeitverlauf angepasst werden. Dafür werden entweder existierende Sprachkonstrukte angepasst oder völlig neu aus existierenden Sprachfragmenten konstruiert. Ermöglicht wird dies durch die Unterscheidung zwischen Objekt- und Metamodelierungssprache.

Folgt man der Meta-Object-Facility (MOF) Architektur der Object Management Group (OMG) wird Sprache in der Systementwicklung in zwei Phasen genutzt [OMG02]:

- Zum einen kommt sie in der Modellierung auf Objektmodell-Ebene bei der Analyse und Dokumentation des Diskursbereichs zum Einsatz.

- Die Phase der Sprachbildung umfasst dagegen die Beschreibung der Syntax und Semantik einer Modellierungssprache unter Rückgriff auf eine Metasprache, die wieder als Modellierungssprache ausgelegt sein kann. In diesem Fall spricht man von einem Metamodell (M2M) [Fr99].

Da eine domänenspezifische Modellierungssprache – als künstliches Artefakt von einem Methodenentwickler geschaffen – über eine modifizierbare Grammatik verfügt, kann diese darüber hinaus frei an bestimmte Bedingungen angepasst werden. Software, die die Entwicklung einer Modellierungssprache unterstützt, wird unter dem Konzept der Meta-CASE Tools zusammengefasst. Bekannte Werkzeuge sind MetaEdit+ [KRT05] oder das cubetto® Toolset [Cu08].

3 Entwicklung von Beschreibungsrahmen („Description Kits“)

Im vorliegenden Paper werden Beschreibungsrahmen (Description Kits) vorgestellt, die den Konsens der Sprachgemeinschaft bezüglich der Menge und Struktur bestimmter für die Analyse relevanter sprachlicher Konzepte repräsentieren. Der Description Kit-Ansatz ist dabei generisch genug, um jede Art von Modellinformationen in deren Beschreibung zweckbezogen einzuschränken. Konkrete Beschreibungen einer fachlichen Information im Analysemodell konkretisieren die Vorstellungen des Modellierers auf einer rein sprachlichen Ebene im Rahmen des vorgegebenen Description Kits.

3.1 Description Kit Language

Der Description Kit-Ansatz stellt die Phase der Sprachbildung in den Mittelpunkt (vgl. Anhang). Im Metamodell auf Ebene 1 erfolgt die Erstellung der so genannten Beschreibungsrahmen-Sprache (Description Kit Language). Hier findet die syntaktische Festlegung der Description Kits statt. Dies beinhaltet die Hierarchisierung verschiedener Description Kit-Konzepte [JW08] sowie die Festlegung ihrer Benutzung. Letzteres erfordert eine Verknüpfung des Metamodells einer konzeptuellen Modellierungssprache, welches zu diesem Zeitpunkt bereits vorliegen muss, und dem Metamodell für die Description Kit Language. Es wird festgelegt, welche Description Kits an welchen Modell-Elementen möglich oder verpflichtend sind. Die Description Kit Language wird also auf das Metamodell der Zielsprache abgebildet.

Die Description Kit Language kann so generisch gehalten werden, dass ein oder mehrere solcher DK nur einmal im Vorfeld erstellt werden müssen und dann in verschiedenen Kontexten benutzt werden können. Im Idealfall liegt eine Description Kit Language vor, die so generisch ist, jede Modellinformation abhängig vom vorliegenden Modellierungszweck als eingeschränktes domänenspezifisches Sprachkonstrukt modellieren zu können.

Die hier vorgestellte Description Kit Language unterscheidet zwischen zusammengesetzten und atomaren Description Kits. Zusammengesetzte Description Kits (Composite Description Kits) sind wiederum eine Aggregation aus anderen Description Kits einer niedrigeren Hierarchieebene. Atomare Description Kits (Atomic Description Kits) dagegen bestehen nicht mehr aus weiteren Rahmen. Weiter kann ein Description Kit (zusammengesetzt oder atomar) im Sinne eines Metadatensatzes durch Parameter angereichert werden. Parameter sind optionale Bestandteile eines Description Kits und beim Modellieren auf Ebene 0 auszufüllen. Allen Parametern eines atomaren Description Kits können optional entweder feste Werte oder Platzhalter zugeordnet werden. Für Parameter sind Platzhalter für frei wählbare Eigenschaften und eine nichtleere Menge von möglichen Parameterwertbelegungen spezifizierbar.

Es wäre allerdings ebenso eine Description Kit Language denkbar, die im Sinne der Modellierung von Rolleninformationen das Konzept „Rolle“ festlegt. Damit werden auf Metamodellebene keine domänenspezifischen Sprachkonstrukte modelliert, sondern die Sprache für Description Kits dieser domänenspezifischen Sprachkonstrukte. Eine Erweiterung einer Metamodellierungssprache, wie sie in der Literatur bisher vorgeschlagen wurde [JW08], ist nicht mehr notwendig.

Eine erzwungene Benutzung von bestimmten Description Kits für bestimmte Modellelemente kann dabei zusätzlich nicht nur die Modellierungsfreiheit in Bezug auf die Description Kits einschränken, sondern auch eine Einschränkung in Bezug auf die Modellierung selbst bedeuten.

3.2 Modellierung und Nutzung von Description Kits

Unter Benutzung dieser Description Kit Language erfolgt dann auf Ebene 0.5 die Definition von Description Kits. Hier findet der eigentliche Adaptionsprozess statt und die Anpassung an den Modellierungszweck. Die Instanziierung von Ebene 1 auf Ebene 0.5 ist im Sinne von „dieser Description Kit entspricht diesem und jenem Konzept“ zu sehen – und legt damit auch die Benutzung der Description Kits beim Modellieren fest.

Insbesondere werden auf Ebene 0.5 adaptive Komponenten eingeführt, welche zum Teil natürlichsprachliche Konzepte darstellen. Wie schon ursprünglich beim Pattern-Ansatz [JW08] bedeutet die Einschränkung auf die auf dieser Ebene eingeführten Konzepte, die dargestellten Sprachkonflikte (vgl. Abbildung 1) teilweise aufzulösen.

Um instanziierte Description Kits (sogenannte „Descriptions“) auf Ebene 0 zu nutzen, sind Description Kits auf Ebene 0.5 zu entwerfen – im Sinne adaptierbarer domänenspezifischer Sprachkonstrukte. Dies könnten je nach Description Kit Language konkrete Rollentypen, konkrete Dokumenttypen (Email, Brief, etc.) oder konkrete Pattern [JW08] sein (vgl. Abbildung 2)

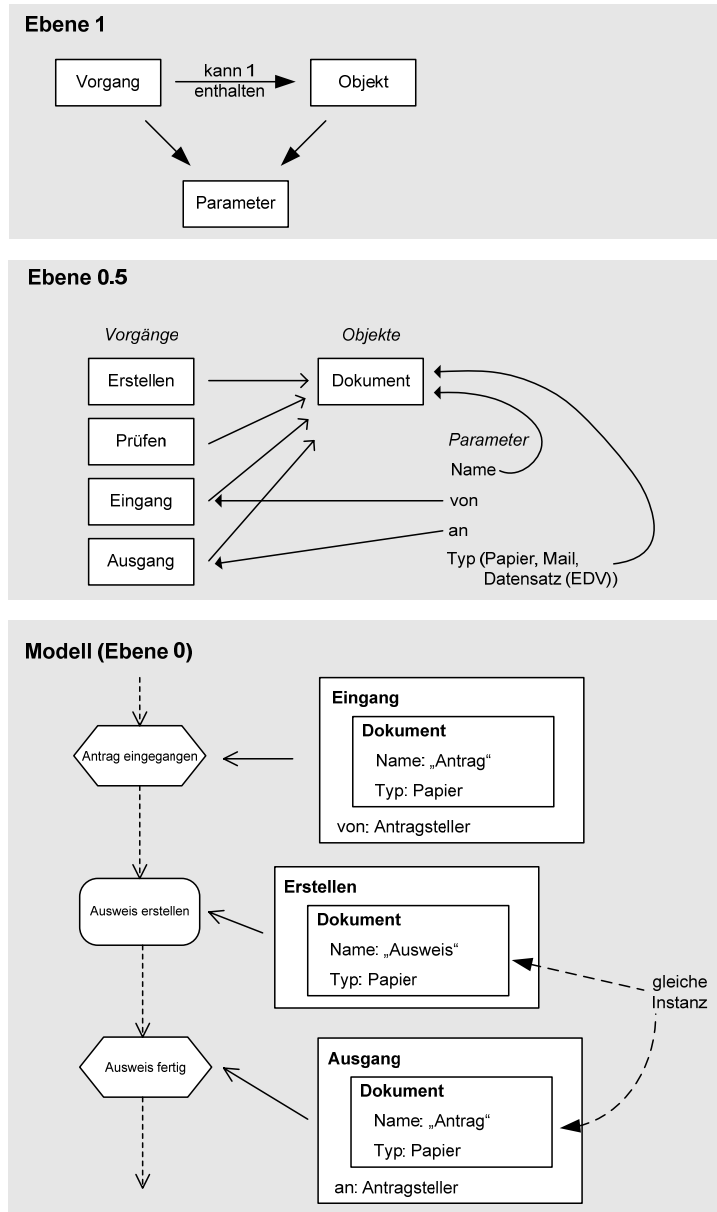


Abb. 2: Anwendungsbeispiel Description Kit Ansatz (stark vereinfacht)

Nach der Definition der Description Kits auf Ebene 0.5 kann die eingeschränkt beschreibbare Modellinformation im Modell auf Ebene 0 modelliert bzw. bereits vormodellierte Descriptions genutzt werden.

Im Beispiel der Rolleninformationen wären hier konkrete Rollen anzulegen. Jede Rolle wiederum kann unterschiedlich ausgestaltet werden und durch unterschiedliche Parameter ergänzt werden (z.B. Studienfach bei Student oder Institut bei einem Mitarbeiter). Dies wäre natürlich auch als „Freitext“ Parameter modellierbar. Genauso könnte man aber die Organisationsstruktur auf Ebene 0.5 zusätzlich zu den konkreten Rollen anlegen, so dass beim Modellieren nur noch die Organisationseinheit an die Rolle „Mitarbeiter“ angeheftet werden muss.

In einem Description Kit ist damit die Menge aller Beschreibungsmöglichkeiten eines eingeschränkten domänenspezifischen Sprachkonstrukt festgehalten, die später bei der konkreten Beschreibung modellierbar sind und durch das Setzen von Parametern konkretisiert werden können. Ein Description Kit legt somit das Vokabular und alle möglichen Beschreibungen eines domänenspezifischen Sprachkonstrukts fest.

Die Instanz eines Description Kits – also eine konkrete Beschreibung im Rahmen der Vorgaben – wird als Description bezeichnet. Die Description erhält einen eindeutigen Namen und ist Teil des fachkonzeptuellen Modells. Alle möglichen Beschreibungen, die der Description Kit zulässt, werden durch das Setzen der Parameter oder Zuordnung von Description Kits niedrigerer Hierarchieebene konkretisiert. Eine vorhandene Wertzuweisung kann während der Modellierung verändert werden. Zu jedem Zeitpunkt der Modellierung auf Ebene 0 ist immer genau eine Konfiguration eines Description Kits auf Ebene 0.5 aktiv. Wie bereits erwähnt, werden die Verbindungsmöglichkeiten des Description Kits mit den Konzepten der konzeptuellen Modellierungssprache auf Metamodellebene festgelegt und bleiben konstant. Die Description Kit Language sollte damit im Fall von Änderungen der Fachsprache nicht betroffen sein. In dem Fall wären nur die Description Kits von einer Adaption betroffen. Veränderungen an einem Description Kit, z. B. durch die Definition von weiteren Parametern, führen einen Description Kit von einer Konfiguration in eine andere über. Die Adaptionoperationen sollten dabei eine Anpassung ermöglichen, ohne die Konsistenz des existierenden fachkonzeptuellen Modells zu gefährden. Dies ist insofern problematisch, als bei der Modellierung der Description Kits keine Annahmen gemacht werden, wann bestimmte Adaptionoperationen realisiert werden.

Vorteil des hier vorgestellten Ansatzes ist die generische Modellierung domänenspezifischer Sprachkonstrukte. Gleichzeitig wird die Weiternutzung bestehender Modellierungssprachen ermöglicht. Durch die Einführung der zusätzlichen Ebene 0.5 ist ein Ansatz gefunden worden, der eine Erweiterung einer Metamodellierungssprache wie dem E³-Modell unnötig macht. Eine Unterstützung der Nähe der Ebenen 0.5 und 0 durch das Modellierungstool macht den Adaptionprozess wesentlich einfacher als im bisherig verwendeten Pattern-Ansatz [JW08]. Die Möglichkeit, unterschiedliche Description Kit Languages zu definieren, bietet eine höhere Flexibilität als der ursprüngliche (prinzipiell ja schon generische) Pattern-Ansatz und erlaubt eine direktere Umsetzung von Ansätzen wie die Anreicherung von Modellen durch Rolleninformationen im Sinne von [JWD08].

4 Anwendungsfall: Konfiguration service-orientierter Architekturen

Mit der stetigen Konsolidierung der Web Service Standards vollzieht sich in der Diskussion zu service-orientierten Architekturen ein Wechsel weg von der Implementierung und dem Deployment von Services hin zum Service Management. Die Indikatoren: Anzahl der Standardisierungsanfragen und die Menge großer Forschungsprogramme in diesem Bereich sind evident für einen wachsenden Bedarf an Managementmethoden für die Abbildung von Geschäftsanforderungen auf Servicekompositionen. Es wird eine Anwendung des Description Kit-Ansatzes vorgestellt, die eine zielgerichtete Abstimmung zwischen den relevanten Geschäftsprozessen und der SOA in fachkonzeptuellen Modellen erlaubt.

Bezug nehmend auf die Abstraktionsebenen der Model Driven Architecture MDA [MD03] ordnet sich diese Abbildung als Transformationsaufgabe zwischen Analysemodellen (Computation Independent Models; CIM) und Entwurfsmodellen (Plattform Independent Models; PIM) ein. Aus dem vorliegenden Modellierungszweck lässt sich nun ableiten, dass eine automatische Überführung die Übersetzung des realweltlichen Problems in eine Form voraussetzt, die eine entsprechende maschinelle Verarbeitung ermöglicht. In der Reduzierung der semantischen Heterogenität innerhalb der Modellelemente aus Analyse- und Entwurfsmodellen liegt ein entscheidender Schritt zu einer automatisierbaren Überführung. Basishypothese ist, dass eine Verbindung zwischen Organisation und IT Domäne unter der Voraussetzung hergestellt werden kann, dass eine Menge an sprachlichen Konzepten simultan in Analyse- und Entwurfsmodellen eingesetzt wird.

Im weiteren Textverlauf wird daher die exemplarische Anwendung des Description Kit-Ansatzes für die Überwindung der semantischen Lücke zwischen semi-formaler Problembeschreibung in Analysemodellen und formaler Lösung in Entwurfsmodellen angewendet. Ziel ist die Vergleichbarkeit von Analyse- und Entwurfsmodellen zu erreichen.

Für den vorliegenden Modellierungszweck wird die Freiheit in Bezug auf die Modellierung von Objekten bzw. Teilen von Objekten eingeschränkt. Diese wurden als Konzepte gewählt, da sie zum einen in der Analyse- als auch in der Entwurfsphase Bedeutung besitzen, damit die Modellierung nicht nachhaltig negativ beeinflussen, und sich zum anderen gut für die Prüfung von Analysemodellen auf Servicekandidaten und den Abgleich zwischen Analyse- und Entwurfsmodellen eignen. Im Sinne der Description Kit-Methode wird folgendermaßen vorgegangen:

- Ebene 1 (Metamodellebene): Auf Metamodellebene wird eine hierarchische Description Kit Language entwickelt. Die Description Kit Language „Objekt“ umfasst die Konzepte Objekt, Attribut und Attributwert. Zusammen dienen sie der Modellierung von Description Kits des Typs „Objekt“ auf Ebene 0.5. Die Konzepte der Description Kit Language werden auf das existierende Metamodell der Modellierungssprache emuliert – im vorliegenden Fall wird auf das Metamodell der Ereignisgesteuerten Prozessketten (EPK) der ARIS Methode Bezug genommen [Sc00]. Damit sind die syntaktischen Beziehungen zwischen Modellierungssprache und der Description Kit Language festzulegen.
- Ebene 0.5: Mit Hilfe der Description Kit Language werden Description Kits für Objekte auf Ebene 0.5 entwickelt. Diese sind adaptiv, können also zur Laufzeit der Modellierung auf Objektmodellebene an verschiedene Anforderungen an die Modellierung von Objekten angepasst werden. In diesem Sinne bilden sie den Konsens zwischen Softwarearchitekten und Fachexperten bezüglich der Menge und Struktur bestimmter für die Analyse und den Entwurf relevanter sprachlicher Konzepte entsprechen.
- Ebene 0 (Modellebene): Auf Ebene 0 werden die Description Kits von Ebene 0.5 instanziiert und bilden individuelle Descriptions. Die Modellierung wird dabei auf den vorgegebenen Description Kit eingeschränkt.

Eine Description wird auf Ebene 0 als konkretes Objekt (z. B. Geschäftsobjekt) oder eines Teils davon interpretiert. Die Konkretisierung auf ein ganz bestimmtes Objekt erfolgt durch die Parametrisierung der Attributwerte, die u. U. durch logische Operatoren miteinander kombiniert werden können.

Description Kits vom Typ „Objekt“ haben damit einen entscheidenden Effekt auf die Aktivitäten während der Modellierung auf Modellebene, da der Freiheitsgrad bei der Beschreibung von Geschäftsobjekten bzw. Teilen von Geschäftsobjekten auf das vormodellierte Vokabular eingeschränkt wird.

4.1 Konzeptuelle Modellierung mit Description Kits

Werden Description Kits in Analysemodellen benutzt, so konkretisieren sie die Vorstellungen des Modellierers auf einer rein sprachlichen Ebene. Dafür werden Zustandsübergangsmodelle eingeführt, die basierend auf einer Erweiterung der EPK funktionale Anforderungen beschreiben (vgl. Abbildung 3).

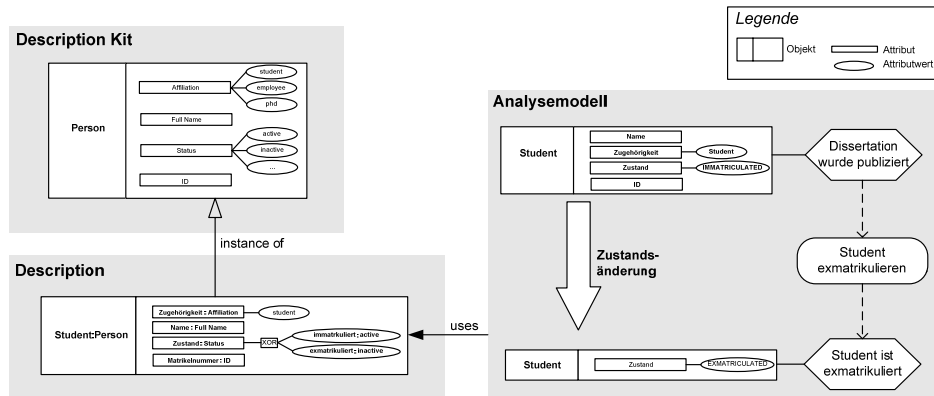


Abb. 3: Erweiterung der EPK als Zustandsübergangsmodell

Dabei wird eine semantische Beziehung zwischen Ereignis und Objekt hergestellt und als betrieblicher Zustand interpretiert. Dieser expliziert sich durch die Sammlung der Zustände der zugeordneten Objekte. Zu automatisierende Geschäftslogik wird damit als Menge von Zustandsübergängen von Objekten dokumentiert.

4.2 Entwurfsmodellierung mit Descriptions

Mit der E³-Methode [Gr04] wurde ein Modell für Services entworfen, das die Abbildung einer Serviceschnittstelle mit Hilfe grafischer Zeichen vorsieht und die Methodensignatur an Descriptions koppelt.

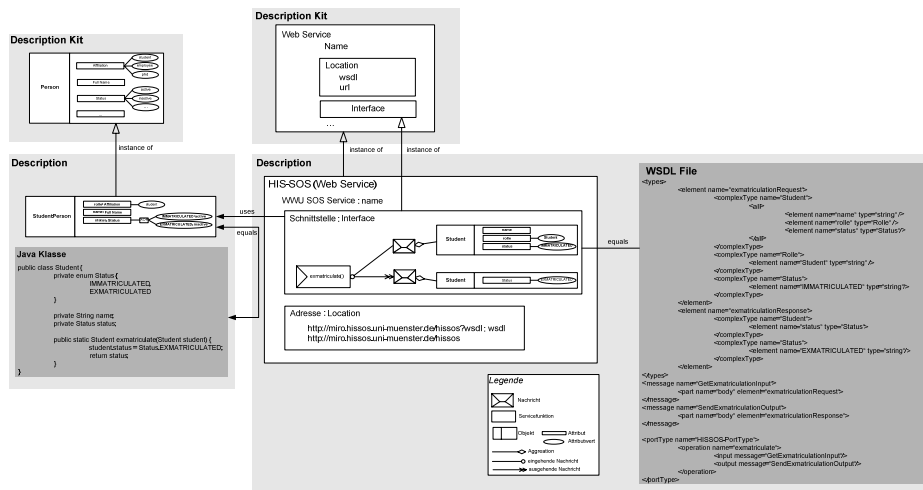


Abb. 4: Modellierung von Web Services mit dem Web Service Description Kit

Bei der Definition der Nachrichtenparameter kommen die Description Kits von Objekt zum Einsatz. Dadurch, dass bei der Modellierung von Analyse- und Entwurfsmodellen in beiden Fällen entsprechende Descriptions benutzt werden, wird die Brücke zwischen beiden Domänen geschlossen. Es wird angenommen, dass die Servicefunktion eines Web Services sich dann für einen bestimmten betrieblichen Kontext eignet, wenn das auslösende respektive resultierende Ereignis einer Prozessfunktion im Analysemodell Descriptions mit einem Zustand umfasst, der dem im Modell der Servicefunktion entspricht. Die Konfiguration wählt automatisch passende Servicefunktionen aus und schlägt dem Modellierer gegebenenfalls eine Anpassung des Soll-Modells vor. Als Ergebnis ist das EPK Modell idealerweise mit der vorliegenden SOA abgeglichen. Ein angepasstes Soll-Modell kann als Referenzmodell – fachliche Lösung mit unterliegender Implementierung – auf andere Organisationen übertragen werden.

Ausblick

Der vorgestellte Ansatz leistet einen ersten Schritt hin zu einer modellgetriebenen Konfiguration einer SOA durch Geschäftsprozessmodelle. Mit der Produktivschaltung von cubetto® Toolset [Cu08] in naher Zukunft und dem dann folgenden Einbezug der dezentralen IT-Betreuungsorganisationen der Universität wird sich auch die Akzeptanz der Methode abzeichnen. Die konsequente Verwendung des Modellierungswerkzeugs im Projekt MIRO wird helfen, es als Mittel der Wahl für das Dokumentieren und Verwalten der SOA zu verwenden und als Teil des integrierten Informationsmanagements zu betrachten.

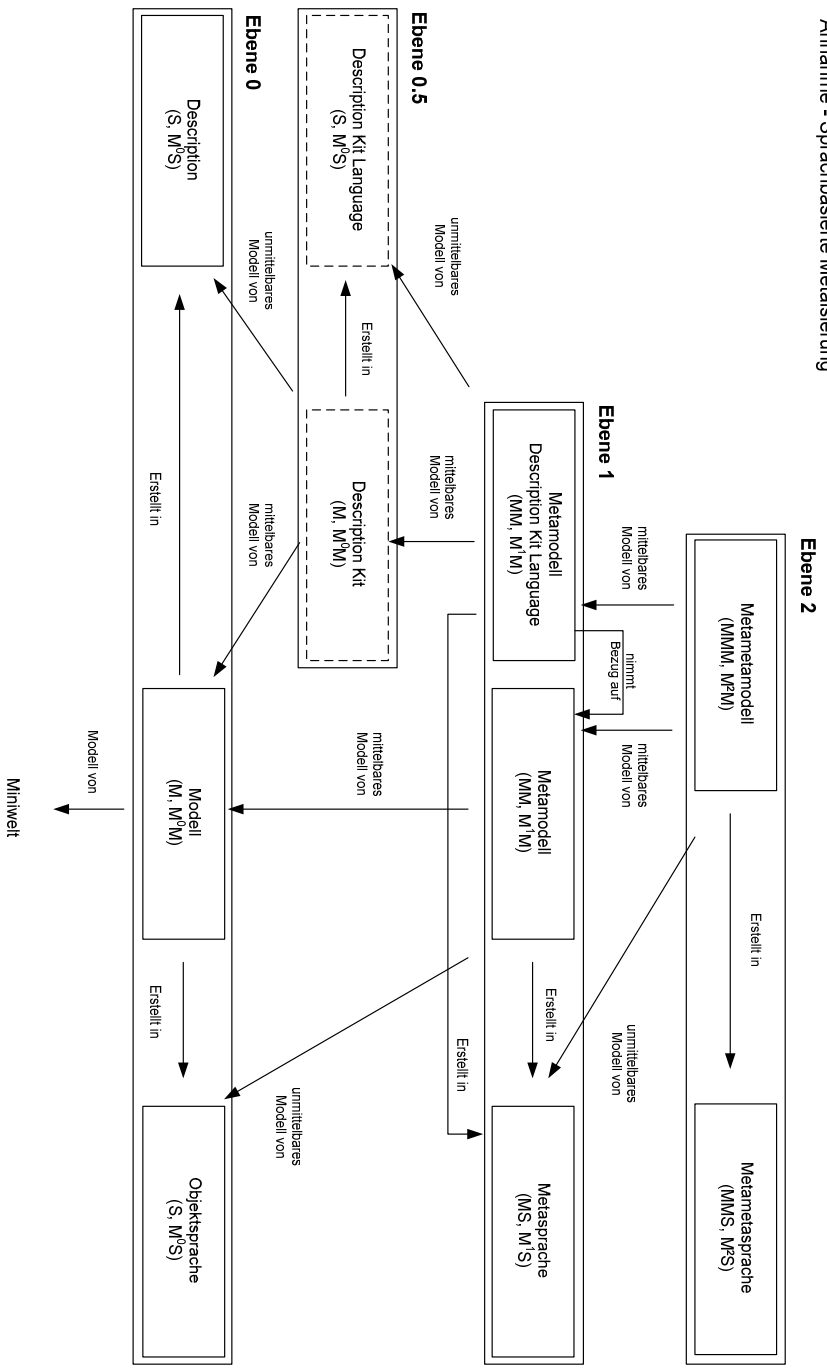
Literaturverzeichnis

- [Be07] Becker, J. et al.: Bausteinbasierte Modellierung von Prozesslandschaften mit der PICTURE-Methode am Beispiel der Universitätsverwaltung Münster. In *Wirtschaftsinformatik* 49 (4), 2007; S. 267
- [Be95] Becker, J.: Grundsätze ordnungsgemäßer Modellierung. In *Wirtschaftsinformatik* 37 (5), 1995; S. 435
- [BHT07] Böhm, B.; Held, W.; Tröger, B.: Integrated Information Management at the University of Munster. In *Changing Infrastructures for Academic Services*. Bad Honnef, 2007.
- [Cu08] Cubetto Toolset: Semture GmbH, www.semture.de, Dresden, 2008.
- [Fr99] Frank, U.: Conceptual Modelling as the Core of the Information System Discipline – Perspectives and Epistemological Challenges. In *Proceedings of the 5th Americas Conference on Information Systems, AMCIS'99*, 1999; S. 695 – 697.
- [Ge07] Gehlert, A.: *Migration fachkonzeptueller Modelle*. Logos Berlin, 2007

- [Gr04] Greiffenberg, S.: Methodenentwicklung in Wirtschaft und Verwaltung. Dr. Kovac, Hamburg, 2004
- [HS06] Hadar, I.; Soffer, P.: Variations in Conceptual Modeling: Classification and Ontological Analysis. In Journal of the AIS 7 (8), 2006; S. 568-592
- [JW08] Jührisch, M.; Weller, J.: Connecting Business and IT – A Model-driven Web Service based Approach. In Proceedings of the 12th Pacific Asia Conference on Information Systems (PACIS 2008), Suzhou, China, 2008
- [JWD08] Jührisch, M.; Weller, J.; Dietz, G.: Application Access Control using Enterprise Models. In Proceedings of the 12th Pacific Asia Conference on Information Systems (PACIS 2008), Suzhou, China, 2008
- [KRT05] Kelly, S.; Rossi, M.; Tolvanen, J. P.: What is Needed in a MetaCASE Environment? In Enterprise Modelling and Information Systems Architectures, 2005; S. 22 – 35.
- [MD03] MDA: Guide Version 1.0.1, Document Number omg/2003-06-01, 2003.
- [OMG02] Object Management Group: Meta Object Facility (MOF) Specification, version 1.4, 2002.
- [Pf07] Pfeiffer, D.: Constructing comparable conceptual models with domain specific languages. In Proceedings of the 15th European Conference on Information Systems (ECIS 2007), St. Gallen, 2007
- [Ro03] Rosemann, M.: Preparation of Process Modeling. In (Becker, J.; Kugeler, M.; Rosemann, M. Hrsg.) Process Management, Berlin et al., 2003; S. 41-78
- [Sc00] Scheer, A.-W.: ARIS – Business Process Modeling. Springer, Berlin, 2000
- [St98] Strahinger, S.: Ein sprachbasierter Metamodellbegriff und seine Verallgemeinerung durch das Konzept des Metaisierungsprinzips. In Proceedings of the Modellierung 98 – Astronomical Society of Australia, 1998

Metastierungsbeziehungen (in Anlehnung an [St98]):

Annahme - Sprachbasierte Metastierung



ANHANG

Ein konfiguratives Metamodellierungswerkzeug

Patrick Delfmann, Sebastian Herwig, Milan Karow, Łukasz Lis

European Research Center for Information Systems
Westfälische Wilhelms-Universität Münster
Leonardo-Campus 3
48149 Münster

[vorname.nachname]@ercis.uni-muenster.de

Abstract: Im vorliegenden Beitrag wird ein Metamodellierungswerkzeug vorgestellt, das in der Literatur vielfach geforderte Funktionalitäten aus den Bereichen des Method Engineering sowie des Modell-Variantenmanagements umsetzt. Aufbauend auf konzeptionellen Überlegungen wird ein Werkzeug konstruiert, das die flexible Spezifikation von beliebigen, untereinander integrierten Modellierungssprachen ermöglicht. Die Anwendung der Sprachen im Rahmen der Modellerstellung wird durch ein redundanzfreies Variantenmanagement auf Sprach- und Modellebene flexibilisiert. Der vorliegende Beitrag beschreibt die Herleitung des Fachkonzepts, sowie die Architektur der entwickelten Software und zeigt deren Funktionsweise exemplarisch.

1 Motivation

Die Informationsmodellierung ist ein essentielles Werkzeug in Informationssystementwicklungs- und Organisationsprojekten [KK84], [Ka88], [DB95], [Da04]. In Organisationsprojekten dienen Modelle als strukturiertes Kommunikationsmittel zwischen den Stakeholdern. Mit Hilfe konkreter Sprachen werden dabei organisationale Informationen wie Geschäftsprozesse, die Aufbauorganisation oder Ressourcen beschrieben. Diese Modelle dienen einerseits als Teil der Anforderungsdokumentation für Softwareentwicklungs- oder Softwareauswahlprozesse, können andererseits jedoch auch (zunächst) unabhängig von IT-Artefakten zur Prozessbeschreibung und -reorganisation von Organisationen in Wirtschaft und Verwaltung eingesetzt werden. In der Gestalt von Referenzmodellen können solche Artefakte über Projektgrenzen hinaus wiederverwendet werden [BDK06]. Beispiele für fachkonzeptuell eingesetzte Modellierungssprachen bzw. Sprachsysteme sind u. a. ARIS [Sc00], BPMN [Ob06], SOM [FS93], und MEMO [Fr99]

In der Softwareentwicklung sind Informationsmodelle ein integraler Bestandteil der Anforderungsbeschreibung und der Entwurfsdokumentation, bzw. nehmen im Rahmen eines modellgetriebenen Systementwicklungsprozesses (z.B. als Umsetzung der Model Driven Architecture, vgl. [Ob03]) die Rolle des zentralen Entwicklungsartefaktes ein. Dabei werden Softwarebestandteile auf unterschiedlichen Abstraktionsebenen durch

Modelle beschrieben und über automatische Transformationsverfahren in spezifischere Modelle oder in Programmcode übersetzt. Im Softwarekontext werden insbesondere die Diagrammtypen der UML eingesetzt [Ob04].

Die o.g. Einsatzszenarien stellen je nach Fokus und Modellierungszweck sehr unterschiedliche Anforderungen an die einzusetzenden Beschreibungsmittel. So ist bspw. im Rahmen eines frühen Fachkonzeptes eine einfache, intuitive Verständlichkeit der Modelle für die Kommunikation mit Fachanwendern und Kunden notwendig. Bei Modellen zur automatischen Quellcodeerzeugung tritt diese Anforderung jedoch vor der Notwendigkeit formaler Präzision in den Hintergrund.

Neben der Einordnung der Modelle in den Entwicklungsprozess spielen Einflussfaktoren wie die fachliche Domäne, Projektziele, eingesetzte Technologien und Produkte, Entwicklungsparadigmen, Wiederverwendung, sowie auch Vorlieben und Expertise der Projektmitarbeiter eine wichtige Rolle bei der Wahl der Beschreibungsmittel. In Gestalt des Method Engineering [Br96], [Ro97] hat sich in diesem Zusammenhang ein Forschungsfeld eröffnet, in dessen Rahmen die Konstruktion spezifischer Modellierungssprachen und Sprachsysteme für individuelle Anwendungen propagiert wird. Arbeiten aus dem Bereich des Modellvariantenmanagements [SGD03], [BDK04], [De06], [DK07], [RA07] betonen darüber hinaus die Notwendigkeit der redundanzfreien Verwaltung von Sprach- und Modellvarianten für unterschiedliche Anwendungsszenarien. Als Beispiele können hier zielgruppenadäquate Perspektiven auf Organisationsmodelle oder plattformabhängige Modellvarianten für Entwurfsmodelle im Rahmen der modellgetriebenen Softwareentwicklung genannt werden. Empirische Studien haben gezeigt, dass in Wissenschaft und Praxis Bedarf nach entsprechenden methodischen Unterstützungen besteht, eine adäquate Werkzeugunterstützung aber bisher fehlt [DK07].

Ziel unserer Forschungsarbeit ist es, diese methodisch-technische Lücke zu schließen und ein integriertes Werkzeug zur Unterstützung des Method Engineering und des Modellvariantenmanagements zu entwickeln. Als Ergebnis dieser Arbeit stellen wir im vorliegenden Beitrag das Modellierungswerkzeug [em] vor, welches die individuelle Spezifikation von Modellierungssprachen, deren Nutzung zur Erstellung von Modellen, sowie das Variantenmanagement auf Ebene von Sprachen und Modellen ermöglicht. Dazu werden zunächst relevante Vorarbeiten vorgestellt und die angewendete Forschungsmethode erläutert (Abschnitt 2). Nach der konzeptionellen Spezifikation des Modellierungswerkzeugs (Abschnitt 3) wird dessen Architektur und Funktionsweise (Abschnitt 4) erläutert. Abschließend werden aus den Ergebnissen weitere Forschungsperspektiven abgeleitet (Abschnitt 5).

2 Eigene und verwandte Vorarbeiten; Forschungsmethode

2.1 Eigene und verwandte Vorarbeiten

Die Konstruktion des Werkzeugs [em] baut auf Ergebnissen umfassender Vorarbeiten auf und repräsentiert die technische Umsetzung der *konfigurativen Referenzmodellierung*

– einer methodischen Konzeption zur redundanzfreien Verwaltung von Sprach- und Modellvarianten. Diese erlaubt die automatisierte Anpassung von konzeptionellen Modellen an spezifische Anwendungskontexte und nutzt das Prinzip der Modellprojektion. Für spezifische Anwendungen nicht relevante Modellelemente werden ausgeblendet. Ein solches Verfahren ermöglicht es, die konfigurative Referenzmodellierung auf beliebige konzeptionelle Modellierungssprachen anzuwenden [BDK04], [De06].

Verwandte Arbeiten, z. B. die Arbeit von SOFFER, GOLANY und DORI, konstruieren anpassbare Modelle einer speziellen Sprache – sogenannte Object-Process Diagrams – mit dem Ziel, ein modellbasiertes Customizing von ERP-Systemen zu ermöglichen. Dabei wird der jeweilige Anwendungskontext anhand von Attributen spezifiziert. Im Anschluss werden die Modelle abhängig von den Attributausprägungen angepasst [SGD03]. ROSEMANN und VAN DER AALST unterstützen in ihrer Arbeit zur Erweiterung Ereignisgesteuerter Prozessketten (EPK) [KNS92] die Spezifikation von Regeln, mit welchen sich Abhängigkeiten zwischen Modellelementen explizieren lassen. Trifft ein Modellersteller z. B. die Entscheidung, bestimmte Funktionen aus einem Modell zu entfernen, leitet ihn die Regelbasis dabei an, auch von diesen Funktionen abhängige Modellbereiche korrekt anzupassen. Ebenso wird die kontextspezifische Anpassung der Modelle ähnlich wie bei [SGD03] methodisch unterstützt [RA07]. Weiterentwicklungen des Ansatzes, die eine Toolunterstützung adressieren sowie die Spezifikation der Anwendungskontexte verfeinern – liefert bspw. [Ro07]. Im Rahmen der hier vorzustellenden Werkzeugentwicklung wird der Ansatz der konfigurativen Referenzmodellierung bevorzugt, da er nicht an eine konkrete Modellierungssprache gebunden ist.

Softwarewerkzeuge, die grundsätzlich für die Spezifikation von individuellen Modellierungssprachen geeignet erscheinen und gleichzeitig Skriptsprachen zur eventuellen Erweiterung durch ein Sprach- und Modellvariantenmanagement bereitstellen, liegen als Metamodellierungswerkzeuge vor (Eine Übersicht gängiger Werkzeuge findet sich bei [DK07]). Empirische Studien haben jedoch gezeigt, dass entweder die Skriptsprachenfunktionalität zur Umsetzung eines Variantenmanagements nicht ausreicht oder die Ergonomie der Benutzerschnittstelle einen praktischen Einsatz nicht vorsieht [DK07]. Aus diesem Grund wird das zu erstellende Werkzeug als Eigenentwicklung realisiert.

2.2 Forschungsmethode

Sowohl die eigenen Vorarbeiten als auch der vorliegende Beitrag folgen dem Forschungsansatz der Design Science [He04], der im Kern die Konstruktion von *wissenschaftlichen Artefakten*, wie z. B. Methoden, Sprachen und Softwarewerkzeugen zum Inhalt hat. Im Rahmen eines Design Science-Ansatzes ist sicherzustellen, dass die durchgeführten Forschungsarbeiten ein relevantes Thema behandeln. Diese Relevanz ist zu belegen. Darüber hinaus ist sicherzustellen, dass die konstruierten Forschungsartefakte einen *innovativen Beitrag* zur bereits in der Disziplin vorhandenen Wissensbasis darstellen, d. h., dass vergleichbare Lösungen nicht bereits existieren. Die konstruierten Artefakte sind im Rahmen eines Design Science-Ansatzes zu *evaluieren*, um sicherzustellen, dass die gesetzten Forschungsziele tatsächlich erreicht werden. Demgemäß sind die Vorarbeiten insbesondere in den Bereichen der Identifikation der *Relevanz* des The-

mas [DK07], [De06], dem Beleg der *Innovationskraft* [De06] sowie der *Konstruktion von wissenschaftlichen Artefakten* [BDK04], [De06] zuzuordnen. Dem gegenüber ist die Implementierung von [εm] neben ihrer Eigenschaft als *Artefaktkonstruktion* auch der praktischen *Evaluation* des perspektivenbasierten Variantenmanagements zuzuschreiben.

3. [εm]-Fachkonzept

Die konzeptionelle Grundlage von Modellierungswerkzeugen ist in der Regel ein umfassendes Datenmodell, welches die Elemente und Beziehungen der instanzierbaren Modellelemente enthält. Bei Modellierungswerkzeugen, welche dediziert eine vorgegebene Modellierungssprache oder eine definierte Reihe von Modellierungssprachen unterstützen, kann die Typ- bzw. Sprachebene als Datenschema fest implementiert werden. Das Daten- bzw. Klassenmodell entspricht in diesem Fall also konzeptionell genau dem Metamodell der Sprache(n). Beispiele einer solchen Umsetzung sind Instanzen des Java Metadata Interface (JMI), welches Java-Klassenstrukturen für MOF-Modelle bereitstellt und insbesondere im UML-Kontext eingesetzt wird [Su02].

Im Falle von [εm] werden Sprachdefinitionen erst zur Laufzeit im Werkzeug selbst festgelegt, so dass auch die Elemente der Sprachebene wiederum Instanzen einer übergeordneten Typebene sind, welche im Datenmodell abzubilden ist. Die Mechanismen der konfigurativen Referenzmodellierung spezifizieren durch ihr Drei-Ebenen-Metamodellsystem [BDK04], [De06] eine weitere Abstraktionsebene. Die direkte Implementierung dieses Systems ist wenig praktikabel, weshalb es für die Konstruktion eines Softwarewerkzeugs in einem einzigen Datenmodell zu konsolidieren ist. Die entsprechenden Mechanismen des Variantenmanagements können somit direkt implementiert werden. Abbildung 1 zeigt schematisch die Gegenüberstellung des [εm]-Datenmodells und der Struktur der Sprach- und Modelldaten zur Laufzeit des Werkzeugs.

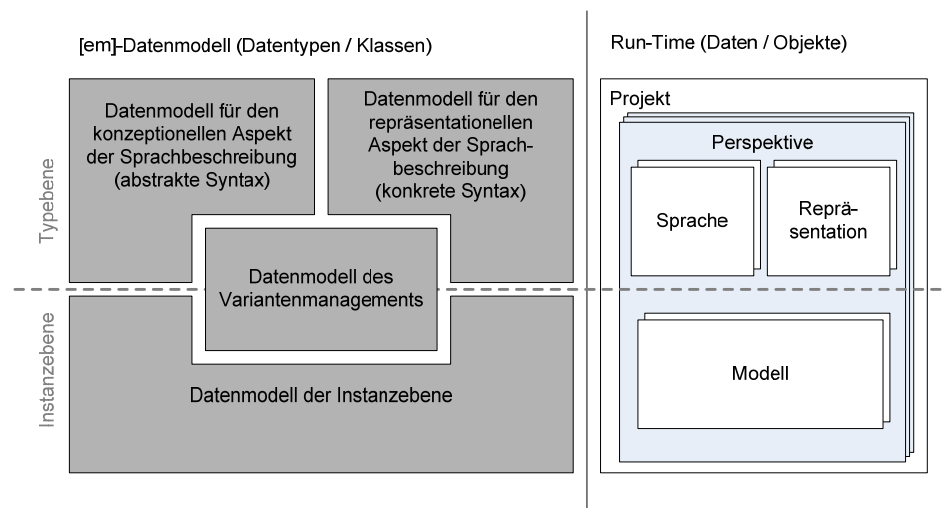


Abbildung 1: Typ-Instanz-Strukturen in [εm]

Das [em]-Datenmodell repräsentiert die Verwaltung sämtlicher Modellierungssprachen- bzw. modellbezogenen Daten sowie deren Zuordnung zu etwaigen Varianten. Ein Ziel der Spezifikation ist dabei die möglichst breite Einsetzbarkeit bei der Definition nutzerindividueller Modellierungssprachen. Hierfür ist eine genaue Untersuchung der syntaktischen Dimension sowohl auf konzeptioneller (auch: abstrakte Syntax) als auch auf repräsentationeller Ebene (konkrete Syntax) notwendig. Zur Darstellung der Teilaspekte des Datenmodells wird im Folgenden die (min,max)-Notation des Entity-Relationship-Modells (ERM) nach SCHLAGETER und STUCKY verwendet [SS83].

3.1 Konzeptioneller Aspekt

Sprachen sind nach ihrer semiotischen Definition *Systeme von Zeichen*, wobei das Zeichen in eine Inhaltsebene (das Bezeichnete) und eine Ausdrucksebene (Zeichenträger) zerlegt werden kann [Ec91]. Die Beziehung zwischen diesen beiden Ebenen wird als *semantische Dimension* der Semiotik untersucht [Mo88]. Die Einbeziehung der *Intention* einer sprachlichen Handlung durch einen Zeichennutzer kennzeichnet die *pragmatische Dimension* der Semiotik. Die Beziehungen der Zeichenträger untereinander werden durch die *syntaktische Dimension* der Semiotik beschrieben.

Für die Konzeption eines Modellierungswerkzeuges zur Spezifikation von Modellierungssprachen und ihrer Nutzung sind die Aspekte der *Semantik* und *Pragmatik* im semiotischen Sinne irrelevant, da die *Interpretation* der Modellelemente (Zeichen) nicht innerhalb des formalen Systems der Software stattfindet. Modellierungstools sind entsprechend auf die *syntaktische Dimension* von Sprachhandlungen beschränkt. Die *Syntax* einer Sprache lässt sich entsprechend minimal durch ihre Zeichenträger und die Beziehung zwischen diesen beschreiben. Im Rahmen von Modellierungssprachen sprechen wir hier von *Elementtypen* und *Beziehungen*. Abbildung 2 zeigt diesen generischen Ansatz.

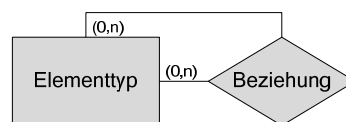


Abbildung 2: Generisches Datenmodell zur Sprachspezifikation

Im abgebildeten Datenmodell werden sämtliche zu instanzierenden Sprachelemente durch Elementtypen dargestellt. Das schließt neben Knoten bspw. Kanten, Attribute etc. ein. Diese generische Sichtweise stellt eine erhebliche Spezifikationskomplexität bereits für vergleichsweise einfache Sprachen dar, da durch die fehlende Spezialisierung sämtliche der Elementtypen über die Beziehungsstruktur zu interpretieren sind. Bei konzeptuellen Modellen im Sinne der Systementwicklung handelt es sich um *Graphen*, so dass *Knoten* und *Kanten* zentrale Sprachelemente darstellen. Abbildung 3 zeigt ein erweitertes Modell, welches die Definition von Kanten sowie deren Richtung und Kardinalität vereinfacht.

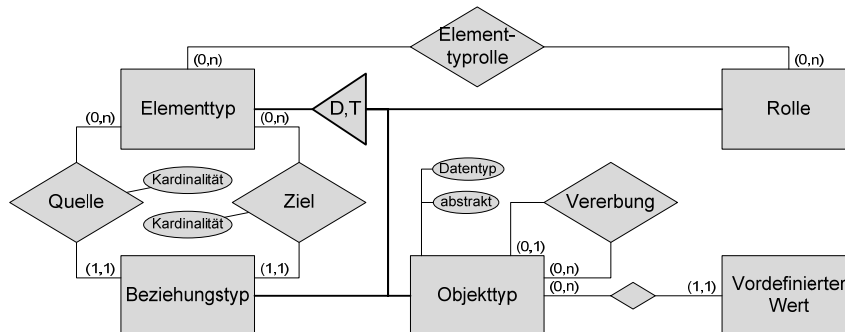


Abbildung 3: Erweitertes Datenmodell zur konzeptuellen Sprachspezifikation

Zentrales Element des Datenmodells ist weiterhin der *Elementtyp*. Hinzugefügt wird der *Beziehungstyp*, der die Spezifikation von zulässigen Beziehungen zwischen Elementen erlaubt, wobei durch *Quelle* und *Ziel* eine Richtung angegeben wird. Die typische Repräsentation einer Instanz des Beziehungstyps ist die *Kante*, jedoch sind Beziehungen nicht auf diese Darstellung beschränkt (vgl. Abschnitt 3.2). Durch die disjunkt-totale Spezialisierung des *Beziehungstyps* vom *Elementtyp* sind auch Beziehungen zwischen Beziehungen spezifizierbar (z. B. Kanten, welche auf Kanten zeigen). Beziehungen nach diesem Schema sind stets binär; n-äre Beziehungen müssen in mehrere Beziehungstypen zerlegt werden. *Objekttypen* repräsentieren in der Regel existenzunabhängige Elementtypen, die grafisch zumeist – jedoch nicht ausschließlich – als *Knoten* repräsentiert werden.

Objekttypen können ihre Eigenschaften und Beziehungen *vererben* – auf diese Weise ist die Spezifikation *abstrakter* Objekttypen möglich, die eine effiziente Eingabe gemeinsamer Eigenschaften der Subtypen erlaubt. Sämtliche Eigenschaften von Elementtypen, bspw. Attribute, werden wiederum als Objekttypen beschrieben und mit ihren Trägern über entsprechende Beziehungstypen verknüpft. Objekttypen können einen *Datentyp* besitzen, so dass sich als Instanzen Primitive (z. B. Strings, Zahlenwerte etc.) bilden lassen. Besitzt ein Objekttyp den Datentyp „Enumeration“, so werden ihm entsprechend *vordefinierte Werte* zugeordnet. Die *Rolle* ermöglicht die weitere Spezialisierung eines Elementtyps in Relation zu einer Beziehung, die dieser Typ eingeht. So kann bspw. ein Ereignis in einer EPK die Rollen „Vorbedingung“ und „Ergebnis“ annehmen.

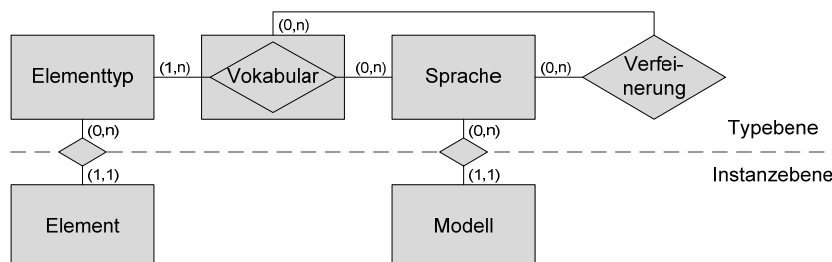


Abbildung 4: Sprachintegration, Modellhierarchien und Instanziierung

Der Aufbau einer methodenorientierten Systemarchitektur erfordert in der Regel die Integration verschiedener Modellierungssprachen (vgl. Abschnitt 1). Die Integration der Sprachen muss dabei über die Sprachspezifikation gewährleistet werden (vgl. Abbildung 4). Elementtypen, die in das *Vokabular* mehrerer Sprachen aufgenommen werden, dienen für diese Sprachen als Schnittstellen und ermöglichen so die horizontale Integration von Modellierungssichten. Zur vertikalen Integration werden *Verfeinerungen* eingesetzt. Auf Typebene wird dabei beschrieben, mit welcher Sprache die nähere Beschreibung der Instanzen eines Elementtyps erfolgen kann (z. B. EPKs für EPK-Funktionen). Elementtypen bzw. Sprachen als *Elemente* bzw. *Modelle* instanziiert. Jedes Modell (bspw. eine konkrete EPK) ist somit eindeutig einer Sprache zugeordnet. Analog gehört jedes Element (bspw. ein konkretes EPK-Ereignis) eindeutig einem Elementtyp an.

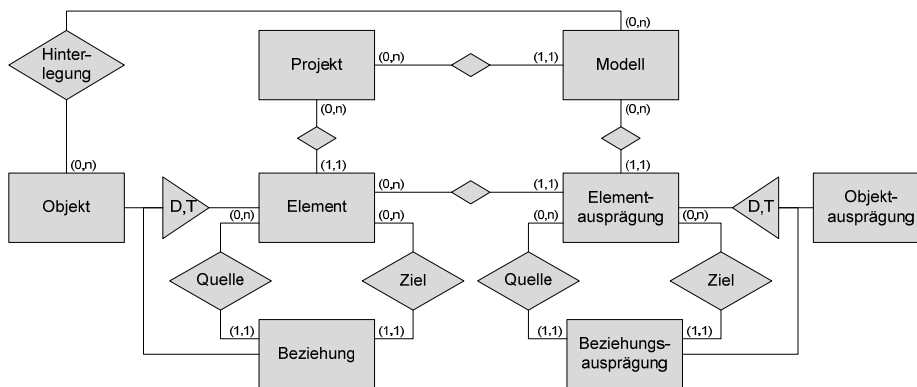


Abbildung 5: Instanzebene des [em]-Datenmodells

Abbildung 5 spezifiziert die Instanzebene des [em]-Datenmodells, auf der die Modelldaten verwaltet werden. Ein Modellierungsvorhaben wird als *Projekt* bezeichnet und fasst alle im Rahmen des Vorhabens erstellten *Modelle* zusammen. Strukturanalog zur Typebene werden *Elemente* in *Objekte* und *Beziehungen* spezialisiert und können untereinander Beziehungen eingehen. Objekte sind mithilfe von *Hinterlegungen* als Modelle verfeinerbar. Elemente sind im gesamten Projekt als Definition einmalig. Von diesen Elementen können *Ausprägungen* erzeugt werden, die Mehrfachverwendungen von gleichen Elementen in Modellen repräsentieren. Jede Elementausprägung ist eindeutig einem Element und eindeutig dem Modell, in dem sie verwendet wird, zugeordnet.

3.2 Repräsentationeller Aspekt

Die Zuordnung der konzeptionellen Typen zu ihren Darstellungsobjekten zeigt der Modellausschnitt in Abbildung 6. *Diagrammtypen* fassen unterschiedliche Darstellungsformen für Sprachen zusammen (für das ERM bspw. (min,max)- vs. Krähenfußnotation). Die Zuordnung der Elementtypen zu einem Diagrammtyp erfolgt über die *Elementtyprepräsentation*. Dem Elementtyp wird so in Abhängigkeit des Diagrammtyps eindeutig eine *Darstellung* zugewiesen. Analog wird einem Objekt, für das eine Verfeinerung angelegt ist, eine Markierung in Form eines *Verfeinerungssymbols* annotiert, das vom Objekt- und Diagrammtyp abhängig ist. Als Visualisierungskonzept für die Elementtyp-

rolle werden auf Darstellungsebene so genannte *Ankerpunkte* eingesetzt. Dabei handelt es sich um Koordinatenpositionen relativ zum grafischen Darstellungsobjekt des Elementtyps, an denen Kanten angeknüpft werden können. Ein Ankerpunkt repräsentiert dabei eine oder mehrere Rollen, die ein Element einnehmen kann. Für jede Rolle können beliebig viele Ankerpunkte definiert werden. Im o. g. EPK-Bespiel würden bspw. für den Objekttyp „Ereignis“ Ankerpunkte der Rolle „Vorbedingung“ an der Unterseite für ausgehende Kontrollflüsse und solche der Rolle „Ergebnis“ an der Oberseite für eingehende Kontrollflüsse angeordnet.

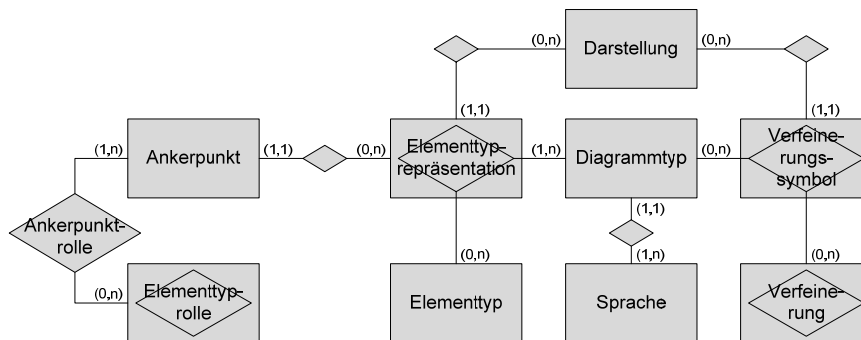


Abbildung 6: Repräsentation konzeptioneller Sprachelemente

Das Konzept der *Darstellung* ist zentral für den repräsentationellen Aspekt der Modellierungssprache. *Objekttypen* werden – sofern ihnen eine sichtbare Darstellung zugeordnet wird – in der Regel durch Knoten dargestellt. Für die Visualisierung von *Beziehungstypen* lassen sich drei verschiedene Muster identifizieren (vgl. Abbildung 7):

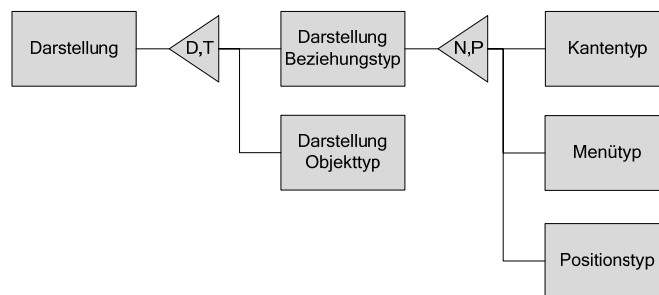


Abbildung 7: Darstellungsarten

- *Kantentyp*: linienartige, grafische Verbinder, die zwei Knoten visuell in Beziehung setzen
- *Positionstyp*: Elemente werden durch die räumliche Anordnung zueinander in Beziehung gesetzt; häufigstes Muster ist dabei die Inklusion (z. B. Subprozesse in BPMN); ein weiteres Muster ist die unmittelbare Verbindung zweier (oder mehrerer) Knoten durch „Aneinanderkleben“.

- *Menütyp*: Das in Bezug genommene Element wird nicht auf dem Zeichenblatt, sondern ausschließlich über ein gesondertes Eigenschaftsfenster oder Kontextmenü des Modellierungswerkzeugs sichtbar. Der Menütyp wird bspw. für Attributierungsbeziehungen eingesetzt.

3.3 Modellvariantenmanagement durch Perspektivenbildung

Die Spezifikation und Umsetzung von Perspektiven ist die konzeptionelle Grundlage für das Sprach- und Modellvariantenmanagement. Perspektiven sind anwendungsspezifische Sichten auf ein Modell, die als Projektion dieses Modells erzeugt werden und nach ihrer Erzeugung eine spezifische Modellvariante darstellen. Abbildung 8 zeigt die Beziehungen, die die Modellierungselemente Perspektiven zuweisen. Jede dieser Beziehungen repräsentiert die datenmäßige Implementierung eines Konfigurationsmechanismus der konfigurativen Referenzmodellierung [BDK04], [De06].

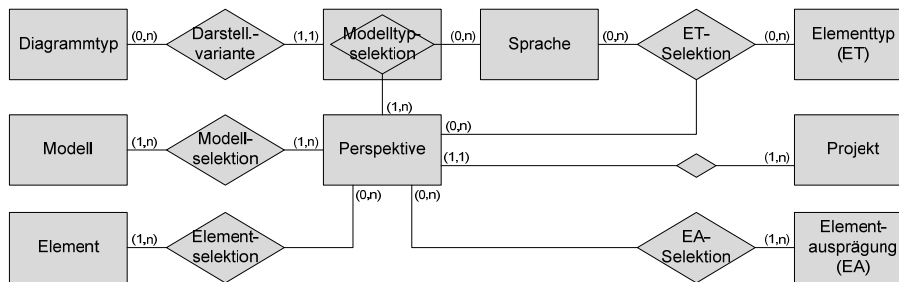


Abbildung 8: Perspektivenmanagement

Jedes Projekt besitzt initial mindestens eine Perspektive. Diese so genannte Standardperspektive ist die Basis für die Generierung weiterer Varianten eines Modellierungsprojekts. Innerhalb der (zusätzlichen) Perspektiven können Modelle modifiziert und erweitert werden. Dabei stehen unterschiedliche Selektions- und Variationsformen zur Verfügung, die determinieren, welche Sprachen mit welchen Elementtypen und Darstellungsformen, welche Modelle, welche Elemente und welche Elementausprägungen in welchen Perspektiven zur Ansicht und Modellierung zur Verfügung stehen. Konkret werden dem Fachkonzept entsprechend die folgenden Konfigurationsmechanismen unterstützt (vgl. zu Konfigurationsmechanismen ausführlich [BDK04], [De06]):

- Mithilfe der *Modelltypselektion* werden die zur Verfügung stehenden Modellierungssprachen je nach Perspektive eingeschränkt. Dies hat zur Folge, dass alle Modelle, die mit einer für die jeweilige Perspektive deselektierte Modellierungssprache erstellt wurden, ausgeblendet werden. Des Weiteren steht die deselektierte Modellierungssprache auch für die Konstruktion neuer Modelle in der entsprechenden Perspektive nicht mehr zur Verfügung.
- *Elementtypselektionen* erlauben die perspektivenabhängige Ein- oder Ausblendung von Sprachbestandteilen. Eine Sprache kann also mithilfe der Elementtypselektion perspektivenabhängig reduziert werden. Als Beispiel sei die Beschneidung der EPK

um organisatorische Objekte genannt, die an Funktionen annotiert werden. Folge dieser Ausblendung ist analog zur Modelltypselektion, dass einerseits alle Modelle in der betrachteten Perspektive um Elemente reduziert werden, die den ausgeblendeten Elementtypen angehören. Andererseits steht die reduzierte Sprache auch zur Modellierung in der Perspektive nunmehr reduziert zur Verfügung.

- Die *Darstellungsvariation* erlaubt die perspektivenspezifische Modifikation des repräsentationellen Aspekts von Modellierungssprachen. Für die Repräsentationsformen der Objekte und Beziehungen – meist Kanten und Symbole – können Varianten angelegt und perspektivenspezifisch ausgewählt werden. Mit der Darstellungsvariation werden insbesondere individuelle Präferenzen der Modellierer adressiert, die sich bspw. aus methodischer oder fachlicher Ausrichtung ergeben können.
- Die Verwaltung von inhaltlichen Modellvarianten wird durch die *Elementselektion* vorgenommen. Einzelne Modellelemente werden als zu einer Perspektive zugehörig oder nicht zugehörig deklariert und werden folglich angezeigt oder ausgeblendet. Die Elementselektion wird dann verwendet, wenn unabhängig von den Ausprägungen des Elements darüber entschieden werden kann, ob es für die Perspektive relevant ist oder nicht.
- Die *Elementausprägungselektion* kommt zur Anwendung, wenn die Entscheidung, ob ein Element relevant ist oder nicht, von seiner konkreten Ausprägung im Modell abhängig ist. So kann das gleiche Element (bspw. ein Entitytyp „Kunde“) für die betrachtete Perspektive ggf. in einem Modell relevant sein, in einem anderen aber nicht.

4. [εm]-Architektur und -Funktionsweise

4.1. [εm]-Softwarearchitektur

Die bereits dargestellten fachkonzeptionellen Anforderungen wurden mit [εm] in Form eines Softwarewerkzeugs implementiert. Dabei wurde insbesondere die gewünschte Praxistauglichkeit sichergestellt, indem auf Benutzerfreundlichkeit bei gleichzeitig hoher Performance geachtet wurde. Die Software basiert auf dem Microsoft .NET-Framework in der aktuellen 3.0-Version und wurde in C# implementiert. Um eine räumlich und zeitlich verteilte Modellierung zu ermöglichen wurde eine mehrschichtige Client/Server-Architektur mit der klassischen Schichtenaufteilung in Datenhaltung, Anwendungslogik und Präsentation zugrunde gelegt. Diese ist um eine vierte Synchronisierungsschicht ergänzt, die für die Client/Server-Kommunikation zuständig ist. Aus Performancegründen ist die Anwendungslogik größtenteils Client-seitig verortet. Teile der eigentlichen Anwendungslogik, die für die Benutzerverwaltung und das Rechtemanagement zuständig sind, sind aus Gründen der Sicherheit Server-seitig positioniert (Abbildung 9).

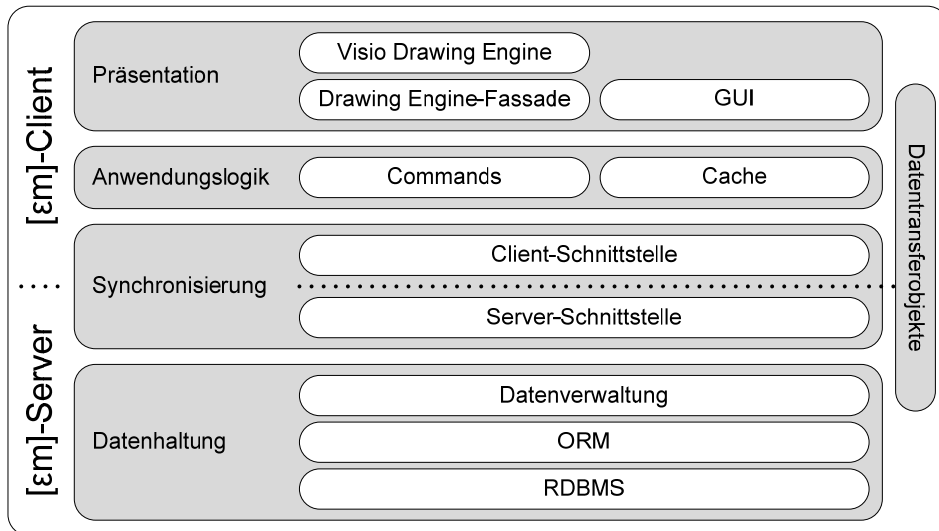


Abbildung 9: [em]-Softwarearchitektur

Die [em]-Architektur folgt grundsätzlich dem Prinzip einer linearen Schichtenordnung, in der von einer Schicht lediglich auf die nächstniedrigere zugegriffen werden kann [Ba00]. Eine Ausnahme hiervon bilden die Datentransferobjekte (DTO), die als Datenaustauschmedium im Rahmen des gesamten Werkzeugs fungieren. Um ihren inneren Aufbau zu kapseln und eine Abstraktion der DTO von den restlichen Komponenten zu erreichen, erfolgen die DTO-Zugriffe ausschließlich über Schnittstellen, die in den jeweiligen Schichten definiert werden. Solange die DTO diese Interfaces implementieren, kann ihre innere Funktionsweise verborgen bleiben, wodurch die Schichtentrennung im Wesentlichen nicht verletzt wird.

Die *Datenhaltungsschicht* basiert auf einem relationalen Datenbankmanagementsystem (RDBMS), welches über das *NHibernate*-Framework angesprochen wird. *NHibernate* sorgt hierbei für das objektrelationale Mapping (ORM), was für zusätzliche Flexibilität bei der Auswahl von RDBMS sorgt und den Implementierungsaufwand verringert. Das zugrunde liegende Datenmodell entspricht dabei dem oben dargestellten Fachkonzept. Die über das ORM übermittelten Daten werden innerhalb der Datenverwaltungskomponente auf die DTO übertragen und gelangen über die *Synchronisierungsschicht* auf den Client. Da die Datenhaltungsschicht für die Benutzerverwaltung und das Rechteverwaltung zuständig ist, werden an den Client nur solche Daten übermittelt, für die der aktuell angemeldete Benutzer eine Berechtigung besitzt. Um Performanceeinbußen zu vermeiden, wird bei der Synchronisierung das *Lazy-Loading-Entwurfsmuster* eingesetzt, welches Daten erst bei einem tatsächlichen Bedarf lädt [Fo02]. In der Synchronisierungsschicht wird die *Windows Communication Foundation (WCF)* eingesetzt, wodurch die gesamte Client/Server-Kommunikation über eine einheitliche Service-orientierte Schnittstelle erfolgen kann.

In der Client-seitigen *Anwendungslogikschicht* wird das *Command-Entwurfsmuster* eingesetzt, infolge dessen alle einzelnen Modellierungsaktivitäten des Benutzers, wie

z. B. das Anlegen eines neuen Modellelements, in sogenannten Command-Objekten in Form einer Warteschlange zwischengespeichert werden. Dies erlaubt eine zeitliche Trennung der Spezifikation, Registrierung und Ausführung der Modelländerungen [Ga95]. Die Command-Warteschlange stellt die vollständige Modellierungshistorie dar und ermöglicht es, zu einem früheren Zustand des Modells zurückzukehren bzw. rückgängig gemachte Schritte wiederherzustellen. Die Benutzeraktivitäten werden dabei solange zwischengespeichert bis der Benutzer mit dem erreichten Modellzustand zufrieden ist und sich für das Speichern der Änderungen entscheidet. In diesem Fall wird die Command-Warteschlange zu einer Transaktion zusammengefasst, über die Synchronisierungsschicht an den Server übermittelt und dort atomar persistiert. Somit werden im RDBMS nur gültige, konsistente Modelle gespeichert.

Die *Präsentationsschicht* stellt eine mittels der *Windows Presentation Foundation (WPF)* implementierte Benutzeroberfläche (GUI) zur Verfügung, in die eine *Drawing-Engine* eingebettet ist. Durch die Verwendung von WPF wurde eine moderne, benutzerfreundliche und performante GUI geschaffen, die für einen praktischen Einsatz von [em] einen kritischen Erfolgsfaktor darstellt. Die *Drawing-Engine* ist für die grafische Darstellung der Modelle während des Modellierens zuständig und wird nach dem Fassaden-Entwurfsmuster über eine eigenständig entwickelte Schnittstelle angesprochen. Somit ist die innere Funktionsweise von der eigentlichen Anwendung gekapselt, was einen eventuellen Austausch der *Drawing-Engine* erleichtert [Ga95]. Aktuell wird in [em] die Zeichenkomponente des Visualisierungswerkzeugs *Microsoft Visio* als *Drawing-Engine* verwendet. Durch die weite Verbreitung dieser Software in der wirtschaftlichen Praxis und den hohen Bekanntheitsgrad bei den Endanwendern wird die Einarbeitungszeit für die Modellierer gesenkt und die Bedienfreundlichkeit des Werkzeugs erhöht. Dabei ist Microsoft Visio in die eigentliche GUI nahtlos integriert, so dass der Eindruck einer einheitlichen Modellierungsoberfläche gegeben wird.

4.2. [em]-Funktionsweise

Der Bedienung von [em] liegt ein dialoggesteuertes Konzept zu Grunde, welches sich in Anlehnung an das in Abschnitt 3 dargelegte Datenmodell in die folgenden Nutzerumgebungen gliedert: Die *Navigationsumgebung* stellt das zentrale Bedienelement zur Verwaltung von Projekten sowie zum perspektivenspezifischen Erzeugen und Anzeigen von Modellen dar (vgl. Abbildung 10). Modelle werden hierbei sowohl projekt- als auch perspektivenbezogen verwaltet. In Abhängigkeit eines Projekts und der ausgewählten Perspektive werden in [em] nur die entsprechend zur Perspektive zugehörigen Modelle bereitgestellt. Die Modelle beinhalten hierbei ausschließlich die Modellelemente, die der entsprechenden Perspektive angehören. Ferner wird durch die perspektivenspezifische Selektion von Modellierungssprachen ebenfalls das Anlegen neuer Modelle auf die verfügbaren Sprachen bzw. Modelltypen eingeschränkt. Die Regelung des generellen Zugriffs auf Projekte, Perspektiven und Sprachdefinitionen erfolgt über eine Rechteverwaltung. Hierbei wird zwischen schreibenden und lesenden Zugriffsrestriktionen unterschieden.

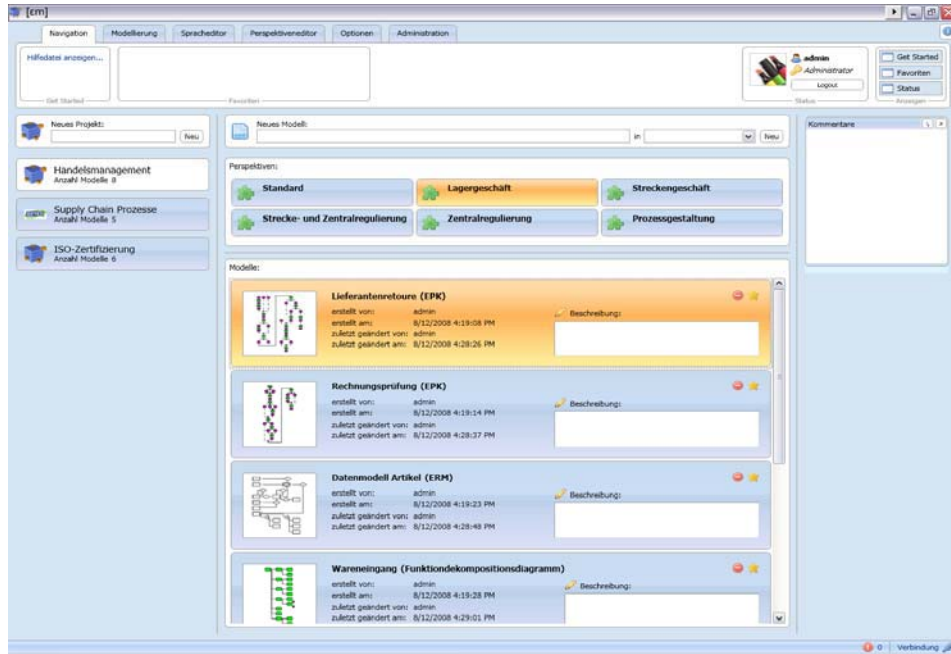


Abbildung 10: Projekt- und Modellverwaltung in [em]

Mit dem Spracheditor ist eine dialoggesteuerte Metamodellierungsumgebung gegeben, welche die Spezifikation von Modellierungssprachen ermöglicht (vgl. Abbildung 11). Die Spezifikation der konzeptionellen Bestandteile einer Sprache stellt den Ausgangspunkt der Sprachkonstruktion dar. Entsprechend der Typebene des Datenmodells sind dazu in einem ersten Schritt die in einer Sprache zu verwendenden Objekttypen zu spezifizieren (1). Im Weiteren sind Rollen zu definieren und den jeweiligen Elementtypen (d. h. sowohl Objekt- als auch Beziehungstypen) zuzuordnen (2). Die Definition von erlaubten Beziehungen zwischen Elementtypen erfolgt in einem nächsten Schritt in Form von Beziehungstypen (3). Zu diesem Zweck werden in [em] die den Elementtypen zugeordneten Rollen untereinander in Beziehung gesetzt. Ferner sind für Beziehungstypen Kardinalitäten zu spezifizieren, die sie hinsichtlich der zu verbindenden Elemente einnehmen, und anzugeben ob sie gerichtet sind. Weiterhin sind Beziehungstypen einer Darstellungsart zuzuordnen (vgl. Abbildung 6). Neben konzeptionellen Bestandteilen gilt es ebenfalls den repräsentationellen Aspekt der Modellierungssprachen zu definieren und den jeweiligen Elementtypen zuzuweisen. (4) Allen Elementtypen, die der Darstellungsart Knoten- oder Kanten typ folgen, sind hierzu eine grafische Repräsentation zuzuordnen. Entsprechende Repräsentationssymbole werden mit jeweiligen Ankerpunkten in Visio erstellt und stehen nach einmaligem Import in einer zentralen Symbolbibliothek zur Verfügung. Abschließend sind die Ankerpunkte zugewiesener Symbole mit Rollen zu belegen (5). Dadurch wird vorgegeben, über welche Punkte Elemente mit anderen Elementen wie bspw. Kanten zu verknüpfen sind.

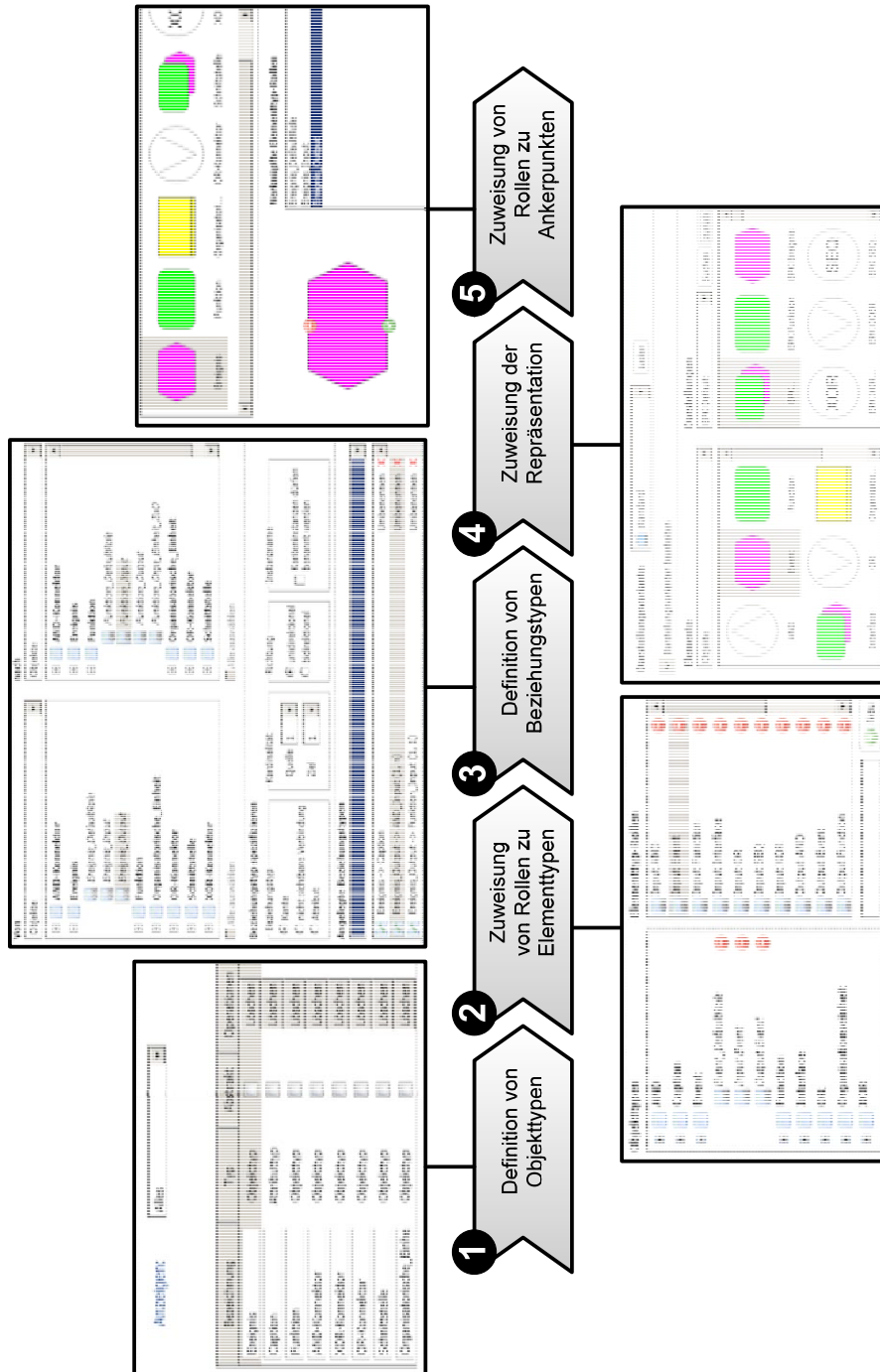


Abbildung 11: Spezifikation von Modellierungssprachen in [em]

Das entsprechend Abschnitt 3.3 zu Grunde liegende Konzept des Variantenmanagements wird über den *Perspektiveneditor* umgesetzt. Dieser dient dem Erstellen und der Spezifikation von Perspektiven. Durch die Zuweisung von *Sprachen* und damit verbundenen *Elementtypen*, *Darstellungsvarianten*, *Modellen* sowie darin enthaltenen *Elementen* und *Elementausprägungen* zu Perspektiven, wird spezifiziert, welche Aspekte in Bezug auf konkrete Modelle in einer Perspektive dargestellt werden bzw. zur Modellierung zur Verfügung stehen (vgl. Abbildung 12). Hierbei werden einer Perspektive nicht zugeordnete Aspekte bei Auswahl dieser ausgeblendet. Ausgehend von der initialen Definition erfolgen weitergehende Modifikationen von Perspektiven ebenfalls über den Perspektiveneditor. Eine Ausnahme hiervon stellt die Selektion von Elementausprägungen dar. Dahingehende Anpassungen können ebenfalls über den Modelleditor direkt im Modell – d. h. auf dem Zeichenblatt – vorgenommen werden. Während der Modellierung kann – abhängig von den jeweiligen Rechten des Modellierers – beliebig zwischen unterschiedlichen Perspektiven hin und her geschaltet werden. Modifikationen an Modellen wirken sich nur auf die aktive Perspektive aus. Eine Ausnahme bildet die Standardperspektive. Änderungen in der Standardperspektive wirken sich auf sämtliche andere Perspektiven aus.



Abbildung 12: Der [em]-Perspektiveneditor

Die Erstellung von Modellen – die eigentliche Modellierung – erfolgt über einen *Modelleditor*. Dieser stellt eine Modellierungsoberfläche zur Verfügung, wie sie in üblichen Modellierungswerkzeugen zu finden ist. Die Erstellung von Modellen im Modelleditor erfolgt grundsätzlich in einer Perspektive. Demgemäß haben Änderungen an einem Modell ausschließlich Konsequenzen für die jeweils selektierte Perspektive. Zur Modellierung werden abhängig von der jeweiligen Perspektive die entsprechend selektierten Objekt- und Beziehungstypen der zugrundeliegenden Sprache über eine Symbolleiste bereitgestellt und können auf dem Zeichenblatt ausgeprägt werden. Die Einhaltung der vorgegebenen Sprachsyntax wird hierbei direkt während der Modellierung überprüft und somit die Erstellung syntaktisch inkorrektur Modelle vermieden. Die perspektivenbezogene Modellierung im Modelleditor wird in Abbildung 13 anhand der Darstellung eines Modells in verschiedenen Perspektiven verdeutlicht.

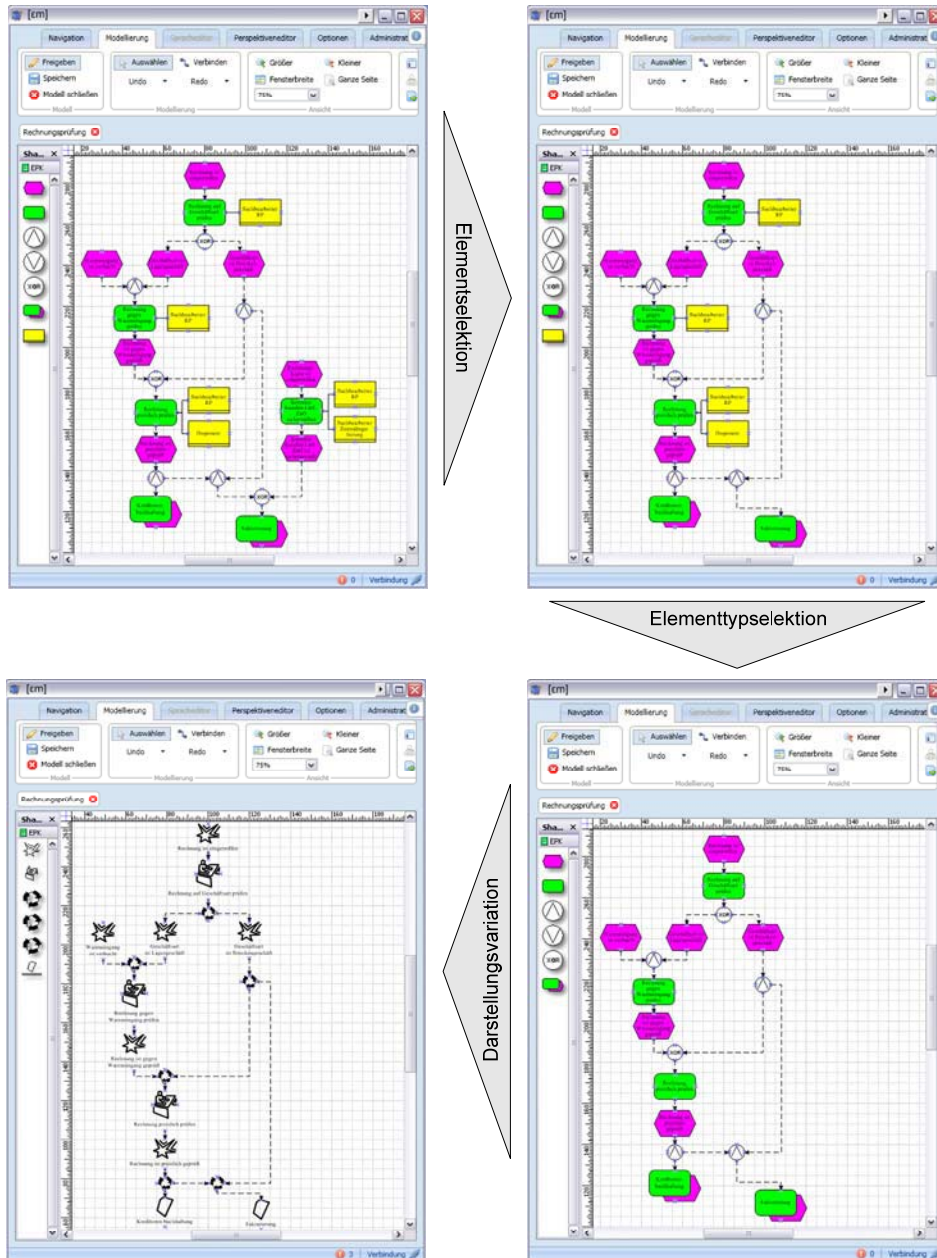


Abbildung 13: Perspektivenspezifische Modellierung in [em]

5. Ergebnis und weiterer Forschungsbedarf

Die Integration einer flexiblen Spezifikation grafischer Modellierungssprachen mit Funktionen des Variantenmanagements eröffnet eine Vielzahl an Anwendungsmöglichkeiten im Kontext der Informations- und Anwendungssystementwicklung. Das vorgestellte Werkzeug [εm] stellt einen entsprechenden Prototypen zur Verfügung, der eine praktische Evaluation der methodischen Konzepte der Referenzmodellierung erlaubt und darüber hinaus die Entwicklung und den Einsatz neuer Beschreibungsmittel vereinfacht.

Die potenzielle Reichweite des Werkzeugs beschränkt sich dabei jedoch auf reine *Beschreibungsmodelle* – Modelle mit unterschiedlichen Zuständen und eigenem Verhalten, wie sie bspw. im Rahmen von *Simulationen* eingesetzt werden, sind zur Zeit nicht mit [εm] abbildbar. Die Ausdrucksmächtigkeit der Sprachspezifikation ist darüber hinaus auf eine lokale Syntax beschränkt, d.h. lediglich *unmittelbare* Beziehungen zwischen Elementtypen sind definierbar. Komplexere syntaktische Restriktionen bspw. über beliebige Pfadlängen sind mit dem vorgestellten Metamodellierungsansatz nicht möglich. Ein kurzfristiges Forschungsziel ist daher, eine entsprechende Constraint-Sprache in das Werkzeug zu integrieren.

Das Konzept des Variantenmanagements durch Projektion hat zur Folge, dass innerhalb perspektivisch angepassten Modellen syntaktische und semantische Inkonsistenzen auftreten können. Diese Inkonsistenzen sind bisher durch den Modellierer im Rahmen der Perspektivendefinition zu behandeln und können ggf. durch Hinzufügen perspektivenspezifischer Elemente wieder hergestellt werden. Eine weitestgehend automatisierte Unterstützung insbesondere der syntaktischen Modellkonsolidierung stellt daher ein aktuelles Forschungsgebiet dar mit hohem Einsatzpotenzial dar. Aktuelle Forschungsarbeiten fokussieren darüber hinaus auf die Unterstützung der fachkonzeptuellen Modellierung durch automatisiert durchgesetzte Bezeichnungskonventionen – hierbei kann über die Syntax der Modellierungssprachen hinaus die Syntax der natürlich-sprachlichen Ausdrücke analysiert und ggf. transformiert werden.

Nicht zuletzt ist das entwickelte Modellierungswerkzeug im Rahmen des gewählten Design-Science-Forschungsansatzes einer umfassenden Evaluation zu unterziehen, indem es sowohl in der Lehre als auch in Praxisprojekten eingesetzt wird. Neben der Benutzerfreundlichkeit, die die Ergonomie und die Verständlichkeit einschließt, ist insbesondere zu messen, in wie weit die Modellierungseffizienz durch die Möglichkeit der individuellen Sprachspezifikation und des Sprach- und Modellvariantenmanagements gesteigert werden kann. Hierzu bieten sich bspw. Vergleichsexperimente mit herkömmlichen Modellierungswerkzeugen an.

Literaturverzeichnis

- [Ba00] Balzert, H.: Lehrbuch der Softwaretechnik, Softwareentwicklung, Heidelberg u. a., 2000.
- [BDK04] Becker, J.; Delfmann, P.; Knackstedt, R.: Konstruktion von Referenzmodellierungssprachen. Ein Ordnungsrahmen zur Spezifikation von Adaptionsmechanismen für Informationsmodelle. *Wirtschaftsinformatik*, 46 (4) 2004, S. 251-264.
- [BDK06] Becker, J.; Delfmann, P.; Knackstedt, R.: Adaptive Reference Modeling: Integrating Configurative and Generic Adaptation Techniques for Information Models. In *Proceedings of the Reference Modeling Conference (RefMod)*, Passau, 2006.
- [Br96] Brinkkemper, S.: Method Engineering: Engineering of Information Systems Development Methods and Tools *Information and Software Technology*, 38 (4) 1996, S. 275-280.
- [Da04] Dalal, N.P., et al.: Toward an integrated framework for modeling enterprise processes. *Communications of the ACM*, 47 (3) 2004, S. 83-87.
- [DB95] Davenport, T.H.; Beers, M.: Managing information about processes. *Journal of Management Information Systems*, 12 (1) 1995, S. 57-80.
- [De06] Delfmann, P.: Adaptive Referenzmodellierung. Methodische Konzepte zur Konstruktion und Anwendung wiederverwendungsorientierter Informationsmodelle. Logos, Berlin, 2006.
- [DK07] Delfmann, P.; Knackstedt, R.: Konfiguration von Informationsmodellen - Untersuchungen zu Bedarf und Werkzeugunterstützung. In *Proceedings of the 8. Internationale Tagung Wirtschaftsinformatik*, Vol. 2 (Oberweis, A.; Weinhardt, C.; Gimpel, H.; Koschmider, A.; Pankratius, V.; Schnitzler, B., Eds.), Karlsruhe, 2007, S. 127-144.
- [Ec91] Eco, U.: *Semiotik: Entwurf einer Theorie der Zeichen*. Fink, München, 1991.
- [FS93] Ferstl, O.K.; Sinz, E.J.: Der Modellierungsansatz des Semantischen Objektmodells (SOM). *Bamberger Beiträge zur Wirtschaftsinformatik*, 1993.
- [Fo02] Fowler, M.: *Patterns of Enterprise Application Architecture*, Reading, 2002.
- [Fr99] Frank, U.: MEMO - Visual Languages for Enterprise Modelling. *Arbeitsberichte des Instituts für Wirtschaftsinformatik*, Koblenz, 1999.
- [Ga95] Gamma, E., et al.: *Design Patterns. Elements of Reusable Object-Oriented Software*, Reading, 1995.
- [He04] Hevner, G., et al.: Design Science in Information Systems Research. *MIS Quarterly*, 28 (1) 2004, S. 75-105.
- [Ka88] Karimi, J.: Requirements and Information Engineering Methods. *Journal of Management Information Systems*, 4 (4) 1988, S. 5-24.
- [KNS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: *Semantische Prozeßmodellierung auf der Grundlage „Ereignisgesteuerter Prozeßketten (EPK)“*. Veröffentlichungen des Instituts für Wirtschaftsinformatik (Scheer, A.-W., Ed.), Saarbrücken, 1992.
- [KK84] Kottmann, J.E.; Kosynski, B.R.: Information Systems Planning and Development: Strategic Postures and Methodologies. *Journal of Management Information Systems*, 1 (2) 1984, S. 45-63.
- [Ro07] La Rosa, M., et al.: Linking Domain Models an Process Models for Reference Model Configuration. In *Proceedings of the 10th International Workshop on Reference Modeling*, Vol. 4928 (ter Hofstede, A.H.M.; Benatallah, B., Eds.), Brisbane, 2007, S. 417-430.
- [Mo88] Morris, C.W.: *Grundlagen der Zeichentheorie*. Fischer-Taschenbuch-Verl., Frankfurt am Main, 1988.
- [Ob03] Object Management Group: MDA Guide Version 1.0.1, 2003. <http://www.omg.org/cgi-bin/apps/doc?omg/03-06-01.pdf>.
- [Ob04] Object Management Group: UML 2.0 Superstructure Specification, 2004. <http://www.omg.org/cgi-bin/doc?formal/05-07-04>

- [Ob06] Object Management Group: BPMN 1.1, 2008. <http://www.bpmn.org/Documents/BPMN%201-1%20Specification.pdf>.
- [Ro97] Rolland, C.: A Primer for Method Engineering. In Proceedings of the INFORSID, Toulouse, 1997.
- [RA07] Rosemann, M.; van der Aalst, W.M.P.: A Configurable Reference Modelling Language. Information Systems, 23 (1) 2007, S. 1-23.
- [Sc00] Scheer, A.-W.: ARIS – Business Process Modelling. 3 Edition, Berlin u.a., 2000.
- [SS83] Schlageter, G.; Stucky, W.: Datenbanksysteme - Konzepte und Modelle. 2 Edition, Stuttgart, 1983.
- [SGD03] Soffer, P.; Golany, B.; Dori, D.: ERP modeling: a comprehensive approach. Information Systems, 28 (9) 2003, S. 673-690.
- [Su02] Sun Microsystems: Java™ Metadata Interface(JMI) Specification, 2002. <http://jcp.org/aboutJava/communityprocess/final/jsr040/index.html>.

SEMAT – Ein Werkzeug zur ontologiebasierten Analyse und zum Vergleich von Prozessmodellen

Martin Kluth¹, Frederik Ahlemann², Frank Teuteberg¹

¹ Institut für Informationsmanagement und Unternehmensführung
Universität Osnabrück
Katharinenstraße 1
49069 Osnabrück
mkluth@uni-osnabrueck.de
Frank.Teuteberg@uos.de

² Institute of Research on Information Systems (IRIS)
European Business School (EBS)
Schloss Reichartshausen
65375 Oestrich-Winkel
Frederik.Ahlemann@ebs.edu

Zusammenfassung: Die automatisierte syntaktische und semantische Analyse von Prozessmodellen ist für viele Anwendungszwecke wie z.B. für das Prozessbenchmarking wünschenswert. Aufgrund der Vielzahl unterschiedlicher Modellierungssprachen und divergierender Verständnisse des modellierten Realitätsausschnitts jedoch meist nicht direkt zu realisieren. In diesem Beitrag wird eine prototypische Umsetzung eines Verfahrens vorgestellt, welches grundsätzlich eine derartige Analyse erlaubt. Die Durchführung von (halb-)automatisierten Prozess-Benchmarking-Initiativen kann damit unter Kosten- und Zeitaspekten effizienter gestaltet werden. Als weitere Anwendungsbereiche, in denen das vorgestellte Verfahren genutzt werden kann, können neben dem Benchmarking unter anderem eine (halb-)automatische Referenzmodellkonstruktion sowie ein „Information Model Mining“ genannt werden.

1 Einleitung

1.1 Motivation

Die Dokumentation von Geschäftsprozessen durch Prozessmodelle schafft Wissen über die Struktur und die Abläufe der unternehmerischen Aktivitäten sowie die Möglichkeit zur Analyse und kontinuierlichen Verbesserung [Sc02, S. 3]. Zur Erstellung solcher Prozessmodelle sind eine Vielzahl an Modellierungssprachen entwickelt worden. Beispielhaft seien hier die *Ereignisgesteuerte Prozesskette* (EPK) und die *Unified Modeling Language* (UML) mit dem UML-Aktivitätsdiagramm genannt. Beim Ver-

gleich und der Interpretation dieser Modelle wird ein einheitliches semantisches und syntaktisches Verständnis aller bei der Modellierung und Interpretation beteiligten Personen vorausgesetzt. Semantisch spiegelt sich dies in einem Konsens über den zu betrachtenden Realitätsausschnitt wieder. Syntaktisch in einem einheitlichen Verständnis bezüglich der Konstrukte der gewählten Modellierungssprache. Sollten zur Prozessmodellierung unterschiedliche Modellierungssprachen eingesetzt werden, so ist für den Vergleich somit zunächst ein Konsens bezüglich der Syntax und Semantik der einzelnen Modelle und Modellkonstrukte zu schaffen. Zum Austausch von Prozessmodellen werden daher Transformationen der Modelle in ein von allen Beteiligten akzeptiertes Modell vorgeschlagen (siehe [VZS05] oder [SMW07]).

Die Transformation von Prozessmodellen in eine einheitliche syntaktische Darstellung löst die Anforderungen, die beispielsweise durch das *Prozess-Benchmarking* (PBM) auf Basis von Prozessmodellen gestellt werden, nur teilweise, da alle am PBM beteiligten Personen ein einheitliches Verständnis über die Domäne besitzen sollten. Ein zentrales Ziel, welches durch das *Prozess-Benchmarking* verfolgt wird, ist die kontinuierliche Steigerung der Leistungsfähigkeit von Arbeitsabläufen (siehe [LÖ99]). Es wird hierbei durch Analyse und Vergleich von Prozessmodellen innerhalb eines Unternehmens oder über Unternehmensgrenzen hinweg versucht, Verbesserungspotenziale der einzelnen Abläufe aufzudecken und umzusetzen. Neben der syntaktischen Vergleichbarkeit muss hierzu auch eine semantische Vergleichbarkeit geschaffen werden. So kann zum Beispiel in einem Modell von „Kunde“ oder „Prozessaktivität“ gesprochen werden, während in einem anderen von „Customer“ und „Prozessschritt“ gesprochen wird. Sollte der Vergleich von Prozessmodellen zusätzlich automatisiert stattfinden, so ist eine computergestützte semantische Interpretation der Modellkonstrukte anzustreben. Hierdurch könnte erkannt werden, dass die Konzepte „Kunde“ und „Customer“ sowie „Prozessaktivität“ und „Prozessschritt“ die gleiche Bedeutung haben. Häufig werden zur semantischen Beschreibung von Prozessmodellen Ontologien genutzt, die semantisches Wissen über eine Domäne in einer maschinenlesbaren Form enthalten. Ansätze, die Ontologien zur weiteren Beschreibung von Prozessmodellen nutzen, sind unter anderem von [He05] und [TF07] vorgestellt worden.

Ein fast identisches Verfahren zur syntaktischen und semantischen Vereinheitlichung von Informationsmodellen wurde von [ATB06] und [Hö07] unabhängig voneinander entwickelt. Im Mittelpunkt dieser Verfahren stehen die Verwendung von Ontologien zur semantischen Anreicherung von Modellkonstrukten sowie die syntaktische Vereinheitlichung von Informationsmodellen über ein Referenz-Metamodell. Die Grundlagen, die in diesem Beitrag vorgestellten prototypischen Umsetzung, wurden jedoch aufgrund des früheren Erscheinungstermins der Arbeit von [ATB06] entnommen. Weiterhin wird in diesem Beitrag die zusätzliche Erfassung von Leistungsinformationen für einzelne Modellkonstrukte und deren Auswertung beschrieben, die ausschließlich in der Arbeit von [ATB06] betrachtet wurde.

1.2 Zielsetzung

Das Ziel dieses Beitrags ist es, die technische Realisierbarkeit des Verfahrens von [ATB06] anhand einer prototypischen Umsetzung aufzuzeigen. Darüber hinaus sollen Erkenntnisse über vorteilhafte Entwurfsentscheidungen als Grundlage für eine spätere Referenzarchitektur gewonnen werden. Der Prototyp soll weiterhin die weitere Evaluation des Konzeptes in Form von praktischen Anwendungen ermöglichen.

Aus Gründen der Komplexitätsreduktion beschränkt sich die hier präsentierte Umsetzung auf Prozessmodelle, jedoch ist eine Überführung der Ausführungen auf das allgemeinere Konzept der Informationsmodelle leicht möglich. Weiterhin wurde keine Evaluation der Nutzung der prototypischen Umsetzung durch fachliche Anwender untersucht.

1.3 Forschungsmethode und Struktur des Beitrags

Die Forschungsmethode, die dem gesamten Forschungsvorhaben zur semantischen und syntaktischen Analyse und des Vergleichs von Informationsmodellen zugrunde liegt, wird als *Design Science* bezeichnet. In dieser Forschungsmethode werden eine zyklische Suche nach einer befriedigenden Lösung eines relevanten Problems und eine rigorose Evaluierung dieser Lösung gefordert ([He04]). Auf die Wirtschaftsinformatik übertragen ist das Ziel dieser Methode somit die Schaffung und Evaluierung von IT-Artefakten, um organisatorische Probleme zu lösen. Die hier vorgestellte prototypische Umsetzung kann somit als Teilergebnis des übergeordneten Forschungsprojektes angesehen werden.

Die Forschungsarbeiten zu diesem Beitrag wurden mit einer eingehenden Literaturanalyse begonnen. Als nächster Schritt folgten die in diesem Beitrag präsentierte prototypische Umsetzung sowie die Ermittlung hieraus resultierender Schlussfolgerungen auf Schwächen des Konzeptes. Die identifizierten Schwachstellen wurden daraufhin behoben, so dass das Konzept weiter verbessert und verfeinert wurde. In den nächsten Schritten soll das Konzept mit Hilfe des hier beschriebenen Prototypen praktisch erprobt werden. Darüber hinaus soll durch eine Kosten-/Nutzenbetrachtung die Wirtschaftlichkeit des Konzeptes untersucht und belegt werden. Im Rahmen dieser Analyse soll auch ermittelt werden, welche organisatorischen Rahmenbedingungen getroffen werden müssen, damit dieses Konzept in einem realen Umfeld effizient eingesetzt werden kann. Diese weitergehenden Evaluationsschritte sind jedoch nicht Teil dieses Beitrags und werden am Ende dieser Arbeit nur kurz skizziert.

Der Beitrag gliedert sich wie folgt: Abschnitt 2 hat die (begrifflichen) Grundlagen zu diesem Artikel zum Gegenstand. Hierzu zählen Prozessmodelle, Ontologien sowie die Beschreibung der bisherigen Forschung im Bereich der semantischen und syntaktischen Analyse von Prozessmodellen. Abschnitt 3 beschreibt das Verfahren von [ATB06] sowie die vorgenommenen Erweiterungen. Die prototypische Umsetzung dieses Verfahrens wird in Abschnitt 4 beschrieben. Nutzeneffekte des Konzeptes im Bereich des Prozess-Benchmarking werden in Abschnitt 5 diskutiert. Der Beitrag schließt in Abschnitt 6 mit einem Fazit sowie einem Ausblick auf weitere Forschung in diesem Bereich.

2 Grundlagen

2.1 Prozessmodelle

Der Anwendungsbereich von Prozessmodellen bzw. von Prozessmodellierungssprachen ist die Abbildung von Prozessen und somit von ablauforganisatorischen Sachverhalten. [BKR05, S. 6] definieren einen Prozess als „... *die inhaltlich abgeschlossene, zeitliche und sachlogische Folge von Aktivitäten, die zur Bearbeitung eines prozessprägenden betriebswirtschaftlichen Objektes notwendig sind.*“ Die mit diesem Prozessverständnis verbundenen dynamischen Aspekte stehen im Zentrum der Modellierung und werden mittels Knoten und Kanten graphisch dargestellt. Die Knoten, welche die eigentlichen Modellelemente, wie zum Beispiel Prozessschritte oder Ereignisse repräsentieren, werden durch gerichtete Kanten in den geforderten zeitlichen und sachlogischen Ablauf eingeordnet. Zur Darstellung dieser Abläufe sind unter anderem die Modellierungssprachen *Ereignisgesteuerte Prozesskette* (EPK) und *Unified Modeling Language* (UML) mit dem UML-Aktivitätsdiagramm entwickelt worden, welche beispielhaft in diesem Beitrag betrachtet werden.

2.2 Ontologien sowie die Ontologiesprache OWL

Die wohl am häufigsten zitierte Definition des Begriffs Ontologie stammt von [Gr93], der eine Ontologie als „*an explicit specification of a conceptualization*“ beschreibt. Er nennt weiterhin drei konstituierende Merkmale von Ontologien:

- **formal:** Ontologien sind maschinenlesbare Modelle, die durch eine Menge von Konzepten (Klassen, Begriffen), Relationen, Instanzen und logischen Aussagen (Axiome) beschrieben werden.
- **explizit:** Die verwendeten Konzepte, deren Randbedingungen sowie Datentypen werden explizit spezifiziert.
- **allgemein:** Der Geltungsbereich einer Ontologie wird von einer Gruppe von menschlichen und maschinellen Akteuren als Kommunikationsgrundlage akzeptiert.

Grundlage vieler Ontologiesprachen, wie der Web Ontology Language (OWL) [OWL], ist die Beschreibungslogik (engl. *Description Logics*), unter der man eine Familie von Sprachen zur Wissensrepräsentation versteht. Merkmal dieser Sprachen ist, dass sie zur Repräsentation von Wissen über eine Anwendungsdomäne in einer strukturierten und formal verständlichen Form genutzt werden können. Hierbei wird das Ergebnis, die Wissensbasis (engl. *knowledge base*), in eine sogenannte T- und A-Box unterteilt. Die T-Box enthält hierbei das Wissen über sämtliche Konzepte einer Domäne, das sogenannte terminologische Wissen, wohingegen die A-Box das Wissen über Instanzen dieser Konzepte, sowie deren Beziehungen untereinander, enthält (siehe [Ba03, S. 50 ff.]). Die Beschreibungslogik ist eine Untermenge der Prädikatenlogik erster Stufe (engl. *first order logic*), ist aber im Gegensatz dazu entscheidbar [Ho05]. Durch die

Entscheidbarkeit sind logische Schlussfolgerungen möglich und es kann so aus vorhandenem Wissen neues Wissen generiert bzw. abgeleitet werden. Als Schlussfolgerung oder Inferenz bezeichnet man hierbei den Vorgang, wenn aus einer Anzahl von Fakten, den Prämissen, ein neues Faktum abgeleitet wird (die Konklusion) ([Ba03]).

Als Ontologiesprache wurde im Rahmen dieses Beitrags OWL gewählt, da sie sich als Standardsprache im Semantic Web etabliert hat, gut dokumentiert ist und eine Reihe von Frameworks zur Bearbeitung und Schlussfolgerung existieren. OWL basiert auf dem *Resource Description Framework* (RDF) und RDF-Schema und nutzt die *Extensible Markup Language* (XML) Syntax von RDF. Sie ist vom *World Wide Web Consortium* (W3C) definiert worden. Der Kompromiss, den diese Sprache eingehen muss, wird von [AV03, S. 69] als Kompromiss zwischen Ausdrucksmächtigkeit und Entscheidbarkeit bzw. leistungsfähiger Schlussfolgerung beschrieben. Da beide Ziele nicht vollständig zu erfüllen erschien, hat sich das W3C dazu entschlossen, die Ontologiesprache OWL in drei Varianten anzubieten, die sich an bestimmte Gruppen von Anbietern und Anwendern richten [AV03, S. 70 ff.]. Diese drei Varianten sind OWL Full, OWL Description Logic (DL) und OWL Lite. Wichtig für das weitere Verfahren ist die Beschränkung auf die Ausdrucksstärken OWL Lite und OWL DL, da die genutzten Ontologien entscheidbar sein sollten.

2.3 Bisherige Forschung im Bereich der semantischen und syntaktischen Analyse von Prozessmodellen

Die Verwendung von Ontologien in Bezug auf Informationsmodelle, insbesondere jedoch Prozessmodelle, wurde in den letzten Jahren in vielen Beiträgen mit unterschiedlichen Zielsetzungen beschrieben und motiviert. So nutzt etwa [Hö07] Ontologien mit dem Ziel der Verbesserung der semantischen Interoperabilität von Prozessmodellen. [He05], [TF07] und [Ce07] verwenden sie zur semantischen Beschreibung der Prozessmodelle und deren automatischer Auswertung. [FL07] nutzen zur ontologischen Evaluierung von Scheers Referenzmodell die Bunge-Wand-Weber (BWW-)Ontologie mit dem Ziel, ontologische Anomalien in diesem Referenzmodell aufzuzeigen. Hingegen nutzen [EKO07] Ontologien im Zusammenhang mit Petri-Netzen zur Messung von Ähnlichkeiten in Prozessmodellen.

In der Literatur lassen sich weiterhin zahlreiche Beiträge zur Verwendung von Metamodellen mit dem Ziel der Vereinheitlichung bzw. Transformation von Prozessmodellen finden. So nutzen [VZS05] Metamodelle mit dem Ziel des interorganisationalen Geschäftsprozessmanagements und [Ka06a] zur transparenten Transformation von Modellen zwischen verschiedenen Modellierungswerkzeugen. [MK07] nutzen ein integriertes Metamodell, das alle Konzepte einer Sprache einer Domäne beinhaltet, um Szenarien wie die Transformation, Integration und Synchronisation in der Geschäftsprozessmodellierung zu verwirklichen.

Weitere Beiträge nutzen Ontologien sowie Metamodelle mit unterschiedlicher Zielsetzung. In ihrem Beitrag beschreiben [RG02] zum Beispiel die Entwicklung eines Metamodells von Konstrukten der BWW-Ontologie mit dem Ziel, den Nutzen dieser

Ontologie Praktikern der Prozessmodellierung näher zu bringen und verständlicher zu machen. [RG02] gehen davon aus, dass Praktikern Metamodelle näher sind als Ontologien. [Ka06b] gehen in ihrem Beitrag den umgekehrten Weg, indem sie Metamodelle in Ontologien überführen und so eine bessere Transformation von Modellen verfolgen. [KH06] nutzen Metamodelle und Ontologien im Zusammenhang mit dem Bereich der semantischen Integration.

3 Ontologiebasierte Analyse und Vergleich von Prozessmodellen

3.1 Syntaktische und semantische Vereinheitlichung von Prozessmodellen

Die Grundlage der syntaktischen Vereinheitlichung von Prozessmodellen, die in unterschiedlichen Modellierungssprachen erstellt worden sind, bildet eine einheitliche Ontologie, die so genannte Basisontologie. Sie ist durch das von [ATB06, S. 9 ff.] beschriebene Referenz-Metamodell motiviert und beinhaltet hier nur einige wenige Konstrukte, die zur Veranschaulichung der Vereinheitlichung von Prozessmodellen benötigt werden (siehe Abbildung 1). So wird beispielsweise in der Basisontologie ein Ereignis (engl. *Event*) eines modellierten Prozesses ganz allgemein als Unterkonzept des Konzepts Prozesselement (engl. *Process Element*) aufgefasst. Ist in einem Prozessmodell ein Ereignis „Projektidee liegt vor“ modelliert, so wird dies als Instanz des Konzeptes Ereignis aus der Basisontologie aufgefasst.

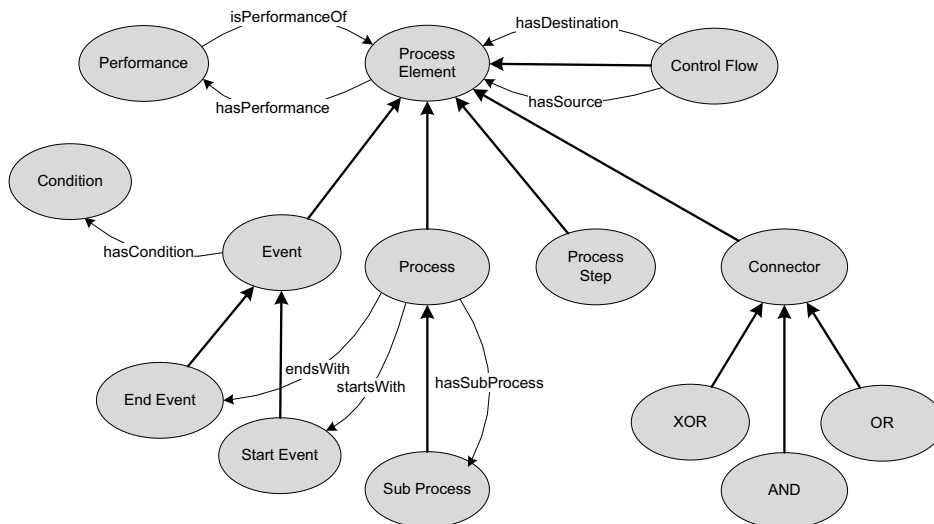


Abbildung 1: Basisontologie in Anlehnung an das Referenz-Metamodell von [ATB06, S. 9 ff.]

Die semantische Vereinheitlichung erfolgt über eine weitere Zuordnung der Instanzen der Basisontologie zu den Instanzen einer Domänenontologie. Diese Domänenontologie ist entweder vorab zu entwickeln oder es kann eine schon entwickelte Ontologie genutzt

werden, wobei in beiden Fällen die Problemangemessenheit beachtet werden sollte. Problemangemessenheit ist in dem hier vorgestellten Verfahren gegeben, wenn alle fachlichen Aspekte der potenziell zu attributierenden Prozessmodelle von der Domänenontologie abgedeckt werden.

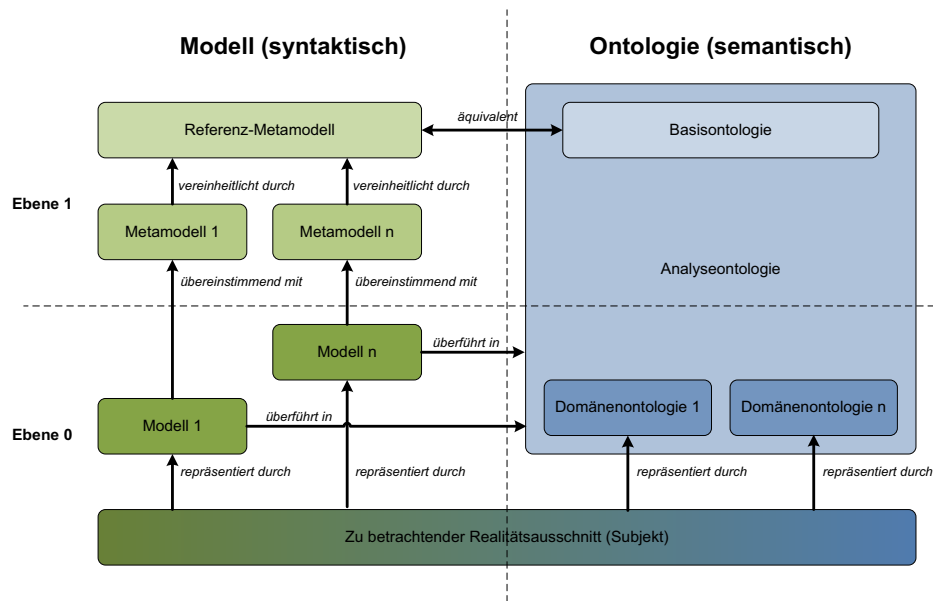


Abbildung 2: Syntaktische und semantische Vereinheitlichung

In Abbildung 2 ist der Zusammenhang zwischen den einzelnen Modellen, Metamodellen und Ontologien dargestellt. Ausgangspunkt ist der zu betrachtende Realitätsausschnitt, der syntaktisch durch 1...n Modelle in unterschiedlichen Modellierungssprachen repräsentiert werden kann. Semantisch kann dieser Realitätsausschnitt durch 1...n Domänenontologien beschrieben werden. Die Modelle, die eine Repräsentation des Realitätsausschnittes in einer Objekt- bzw. Modellierungssprache darstellen, befinden sich nach [St98] auf der untersten Beschreibungsebene. Analog können auch die Domänenontologien zugehörig zu dieser untersten Beschreibungsebene aufgefasst werden, da sie den Realitätsausschnitt durch eine Ontologiesprache semantisch beschreiben. Auf der nächsthöheren Ebene erfolgt eine Beschreibung der einzelnen Objektsprachen durch Metamodelle. Gegenstand der Metamodelle sind somit die Sprachkonstrukte bzw. Modellelementtypen der einzelnen Modelle. Dieses vereinheitlichte Referenz-Metamodell wird in eine äquivalente Basisontologie überführt, so dass in dieser Ontologie Konzepte existieren, welche die vereinheitlichte und für relevant erachteten syntaktischen Modellelementtypen der Metamodelle repräsentieren. Durch Nutzung der Basisontologie kann eine syntaktische Auswertung, wenn auch auf einer höheren Abstraktionsebene, von überführten Modellkonstrukten realisiert werden. Diese Überführung findet durch Erzeugung von Instanzen dieser Basisontologie für jedes Modellkonstrukt statt. [Hö07] spricht in diesem Zusammenhang von „*lifting*“.

Zusammengefasst bildet die Basisontologie, die zur semantischen Beschreibung genutzten Domänenontologie sowie die Instanzen der überführten Modellkonstrukte die sogenannte Analyseontologie. Jegliche Analysen und Vergleiche der Modelle werden ausschließlich auf dieser Ontologie vollzogen.

Die Modelle sowie Domänenontologien der Ebene 0 in Abbildung 2 werden in klassischen Prozessmodellierungs- sowie Ontologiebearbeitungswerkzeugen erstellt. Die Konstrukte dieser Modelle können in den Modellierungswerkzeugen – dem Ansatz des hier vorgestellten Prototypen folgend – auch direkt mit Konzepten einer Domänenontologie verbunden werden. Eine Überführung der verbundenen Konstrukte in die Analyseontologie erfolgt automatisiert durch den Prototypen. Die in der Ebene 1 dargestellten Aspekte liegen dem vorgestellten Prototypen zu Grunde und entziehen sich dem Einfluss des Anwenders.

3.2 Syntaktische und semantische Vereinheitlichung von Prozessmodellen an einem Beispiel

In Abbildung 3 sind das Verfahren sowie die Verwendung der einzelnen Ontologien am Beispiel eines Projektportfolioplanungsprozesses illustriert. Es sind auszugsweise zwei divergierende Prozessmodelle sowie eine Domänenontologie dargestellt. Das erste Modell zeigt eine EPK, welche nur drei Modellelementtypen verwendet: Funktion, Ereignis und Kontrollfluss. Dem gegenüber ist der gleiche Prozessausschnitt nochmals als UML-Aktivitätsdiagramm modelliert. Die genutzten Modellelementtypen sind Aktion, Kontrollfluss und Objektknoten. Durch eine Instanziierung der Modellkonstrukte als Instanzen von Konzepten der Basisontologie wird eine Überführung des Prozessmodells in die so genannte Analyseontologie vollzogen. Im Beispiel kann dies an der Instanziierung bzw. Überführung der Funktion „Prozessantrag schreiben“ der EPK nachvollzogen werden (Pfeil von „Projektantrag schreiben“ der EPK zu „Projektantrag schreiben (EPK)“ in der Analyseontologie in Abbildung 3).

Grundlage der Analyseontologie ist die in Abbildung 1 vorgestellte Basisontologie. Im Beispiel wird deutlich, dass die Modellelementtypen Funktion (EPK) und Aktion (UML) syntaktisch äquivalent sind, da die Ausprägungen dieser Typen in Instanzen des Konzepts *Process Step* überführt werden. Erfolgt nun eine Zuordnung der erzeugten Instanzen der Analyseontologie zu Instanzen der Domänenontologie, dargestellt durch Pfeile mit der Beschriftung „sameAs“ in Abbildung 3, sind weitere Aussagen mittels Inferenz möglich. Es kann zum Beispiel darauf geschlossen werden, dass der Prozessschritt „Projektantrag schreiben“ der EPK identisch mit drei unterschiedlichen Prozessschritten im UML-Aktivitätsdiagramm ist. Diese Information ist über die Eigenschaft „hatTeilaufgabe“ in der Domänenontologie ermittelbar. Sind zusätzlich noch Leistungsinformationen (wie z.B. Durchlaufzeiten oder Ressourcenbedarfe) zu den einzelnen Prozessschritten erfasst worden, so lassen sich weitere Aussagen schlussfolgern. Es lässt sich beispielsweise ableiten, dass die Informationen zu Kosten und Durchlaufzeiten der Konstrukte „Aufwand schätzen“, „Wirtschaftlichkeit analysieren“ und „Projektantrag schreiben“ des UML-Aktivitätsdiagramms entsprechend

zu aggregieren sind, um sie mit dem Konstrukt „Projektantrag schreiben“ der EPK vergleichen zu können.

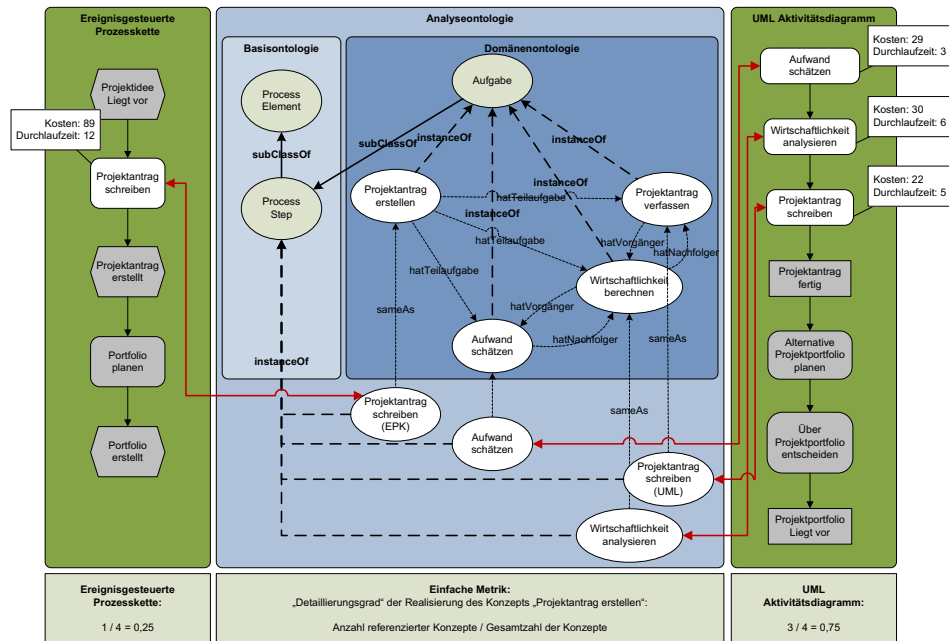


Abbildung 3: Beispiel

4 Software-Komponenten des SEMAT-Werkzeugs

Die prototypische Umsetzung des Verfahrens erfolgt durch mehrere Software-Komponenten, welche für die erforderlichen Teilaufgaben des Verfahrens genutzt werden. Zur Erstellung von Domänenontologien dient der *SEMAT Ontologie-Editor*, zur Modellierung und Attributierung der Informationsmodelle das *SEMAT Modellierungswerkzeug* und zur Analyse der Modelle das *SEMAT Analysewerkzeug*. Das Kürzel SEMAT (SEMantische ATtributierung) wurde vor die Namen der Komponenten gesetzt um die Erweiterung bestehender Software-Komponenten bzw. die Eigenentwicklung dieser zu verdeutlichen.

Für die zentrale Speicherung aller Ergebnisse sowie der Wiederherstellung dieser zur weiteren Bearbeitung wird als vierte Komponente ein Repository in Form eines Datenbanksystems benötigt. Da eine detaillierte Beschreibung aller Komponenten dem Rahmen dieses Beitrags nicht gerecht werden würde, erfolgt eine detaillierte Betrachtung nur für das *SEMAT Analysewerkzeug* und etwas eingeschränkt für das *SEMAT Modellierungswerkzeug*.

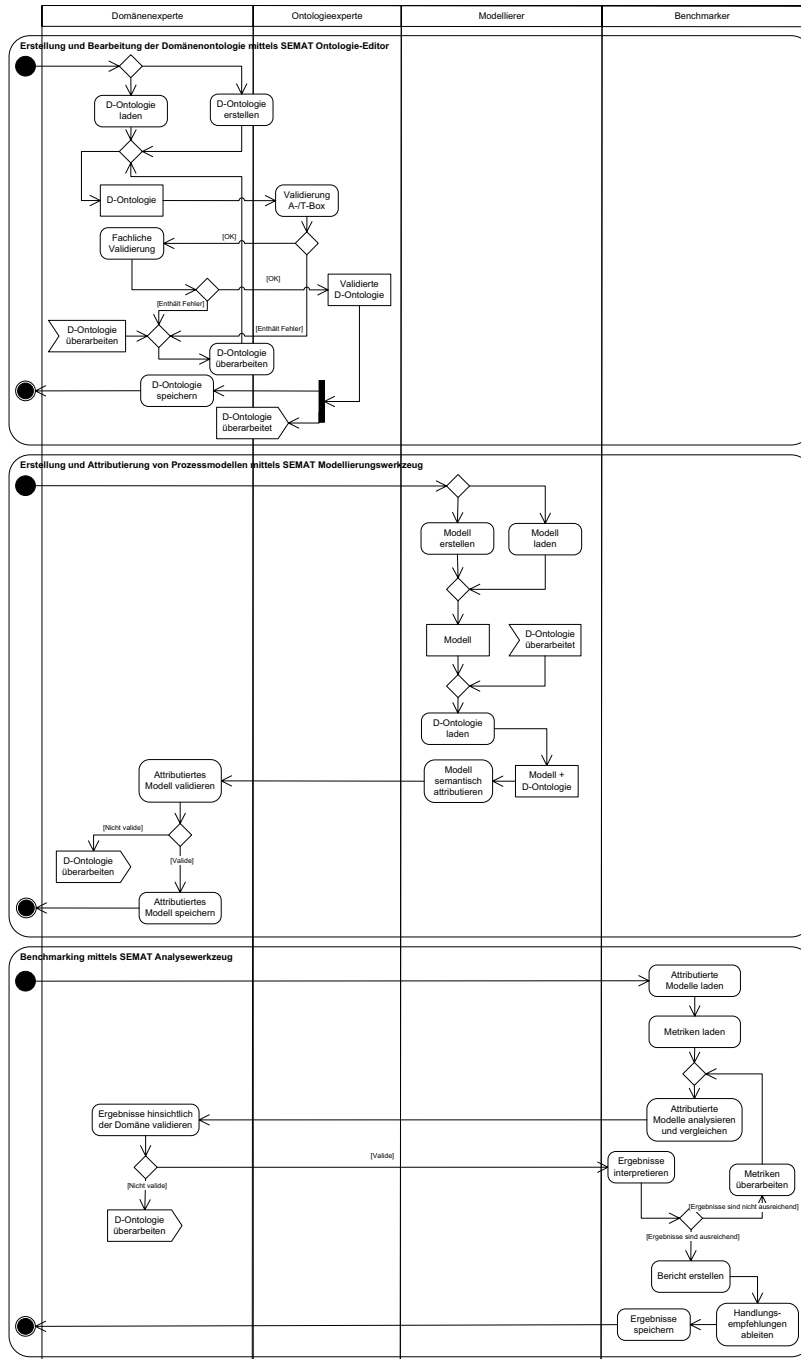


Abbildung 5: Ablauf des Benchmarkings mit dem SEMAT-Werkzeug

In Abbildung 5 ist der Ablauf des Prozessbenchmarking mittels der Software-Komponenten des SEMAT-Werkzeugs anhand von 3 Prozessschritten sowie ihrer möglichen Akteure skizziert:

Im ersten Schritt erfolgt die *Erstellung und Bearbeitung der Domänenontologie mittels des SEMAT Ontologie-Editors* durch die Akteure Domänen- und Ontologieexperte. Sollte bereits eine Domänenontologie existieren, so kann diese aus dem *Datenbanksystem* geladen werden. Sollte noch keine Domänenontologie existieren, so kann diese in Zusammenarbeit der Domänen- und Ontologieexperten mittels des *SEMAT Ontologie-Editors* erstellt und validiert werden. Die Validierung der Ontologie in Bezug auf die A- und T-Box erfolgt durch den Ontologieexperten, während die fachliche Validierung durch den Domänenexperten erfolgt. Sollten durch die Validierungen technische oder fachliche Fehler in der Ontologie ermittelt worden sein oder ein Signal zur Überarbeitung der Domänenontologie empfangen werden, so erfolgt eine Überarbeitung. Bei erfolgreicher fachlicher und technischer Validierung wird die Domänenontologie in dem *Datenbanksystem* gespeichert und ggf. ein Signal gesendet, dass die Ontologie überarbeitet wurde.

Die *Erstellung und Attributierung von Prozessmodellen mittels des SEMAT Modellierungswerkzeugs* erfolgt durch die Akteure Modellierer und Domänenexperte. Der Modellierer erstellt das zu attributierende Prozessmodell oder lädt es aus dem *Datenbanksystem* in das *SEMAT Modellierungswerkzeug*. In einem nächsten Schritt lädt er die den modellierten Realitätsausschnitt repräsentierende Domänenontologie aus dem *Datenbanksystem* und attributiert das Prozessmodell mit den semantischen Informationen aus der Domänenontologie sowie den vorhandenen Leistungsinformationen des Prozesses. Nach der Validierung des Modells durch den Domänenexperten wird das Modell im Erfolgsfall in das *Datenbanksystem* gespeichert oder, falls die Validierung nicht erfolgreich war, ein Signal zur Überarbeitung der Domänenontologie (siehe Abbildung 5) gesendet.

Im letzten Schritt, dem *Benchmarking mittels des SEMAT Analysewerkzeugs*, erfolgen die Analyse und der Vergleich von attribuierten Prozessmodellen. Der Benchmarkler lädt sich hierzu die zu betrachtenden Modelle sowie vorher entwickelte Metriken aus dem *Datenbanksystem*. Danach hat er durch das *SEMAT Analysewerkzeug* die Möglichkeit die attribuierten Modelle zu analysieren und zu vergleichen. Hierbei können auch eigene Abfragen des Benchmarklers zur Analyse genutzt werden. Die Ergebnisse der Analyse und des Vergleichs der Modelle wird durch den Domänenexperten hinsichtlich der Domäne validiert. Sollte dies erfolgreich sein, so können die Ergebnisse durch den Benchmarkler interpretiert werden oder aber das Signal zur Überarbeitung der Domänenontologie gesendet werden. Nach der Interpretation der Ergebnisse durch den Benchmarkler kann bei ausreichendem Erkenntnisgewinn durch die Analyse und den Vergleich ein Bericht erstellt, Handlungsempfehlungen abgeleitet und die Ergebnisse im *Datenbanksystem* gespeichert werden. Anderenfalls erfolgen eine Überarbeitung der Metriken und eine nochmalige Analyse sowie ein Vergleich der Modelle.

4.1 Basisontologie in der Ontologiesprache OWL

Eine Überführung der Basisontologie in OWL ist in Abbildung 6 dargestellt. Im Wesentlichen ist dies eine Überführung der in Abbildung 1 dargestellten Ontologie als T-Box. Die Beziehungen zwischen den Konzepten sind als ObjectProperties dargestellt, die eine Domäne (engl. *domain*) und einen Wertebereich (engl. *range*) besitzen. Detaillierter als in Abbildung 1 sind hier Kardinalitäten dargestellt. So kann zum Beispiel ein Kontrollfluss nur eine Quelle (engl. *source*) und ein Ziel (engl. *destination*) haben.

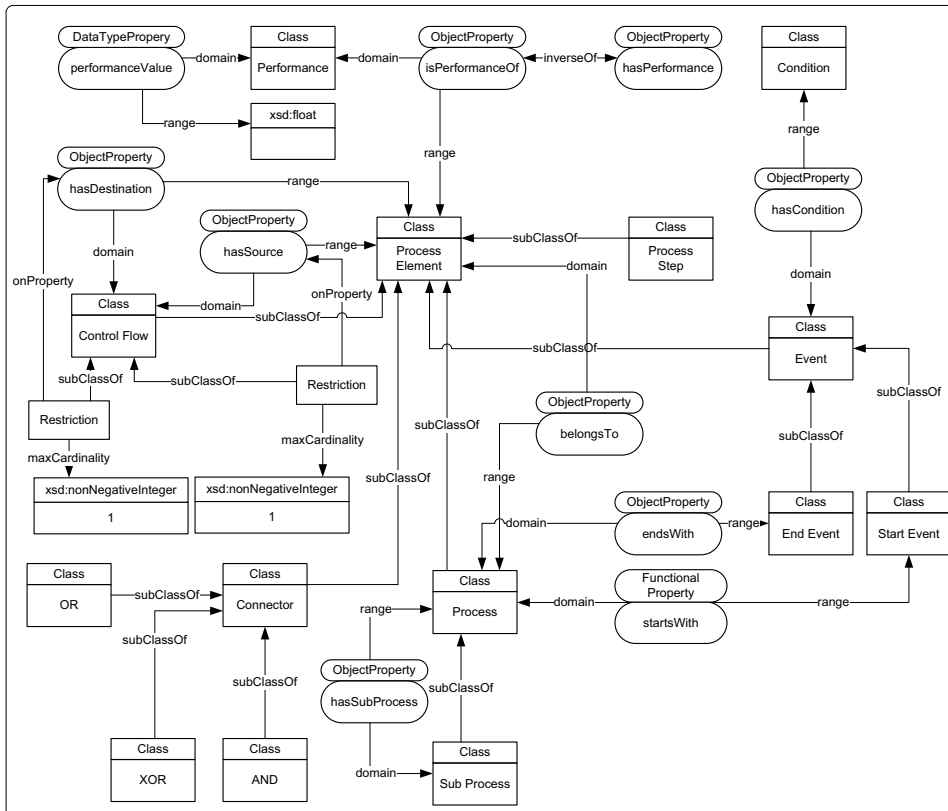


Abbildung 6: Basisontologie in der Ontologiesprache OWL

4.2 SEMAT Modellierungswerkzeug

Zur benutzerfreundlichen Attributierung von Informationsmodellen wurde prototypisch ein Microsoft Visio 2007 Add-In entwickelt. Die Entscheidung, Microsoft Visio 2007 als Modellierungswerkzeug zu nutzen, ist motiviert durch die Erweiterbarkeit mittels Add-Ins sowie der hohen Verbreitung und Akzeptanz von Microsoft Visio. Das Add-In ermöglicht die semantische Attributierung und Erfassung von Leistungsinformationen für UML-Aktivitätsdiagramme und Ereignisgesteuerte Prozessketten. Hierzu kann der Modellierer über ein Kontextmenü zu jedem Modellkonstrukt ein Konzept aus der Domänenontologie suchen und dem Konstrukt zuordnen sowie Leistungsinformationen erfassen, die hier beispielhaft für die Durchlaufzeit und die Kosten abgebildet sind (siehe Abbildung 7). Sollten Instanzen der Domänenontologie in verschiedenen Sprachen beschrieben sein, wie hier in Englisch und Deutsch, so werden diese Instanzen in diesen Sprachen zur Auswahl angeboten. Weiterhin ist das Laden und Speichern der Modelle in das Datenbanksystem möglich.

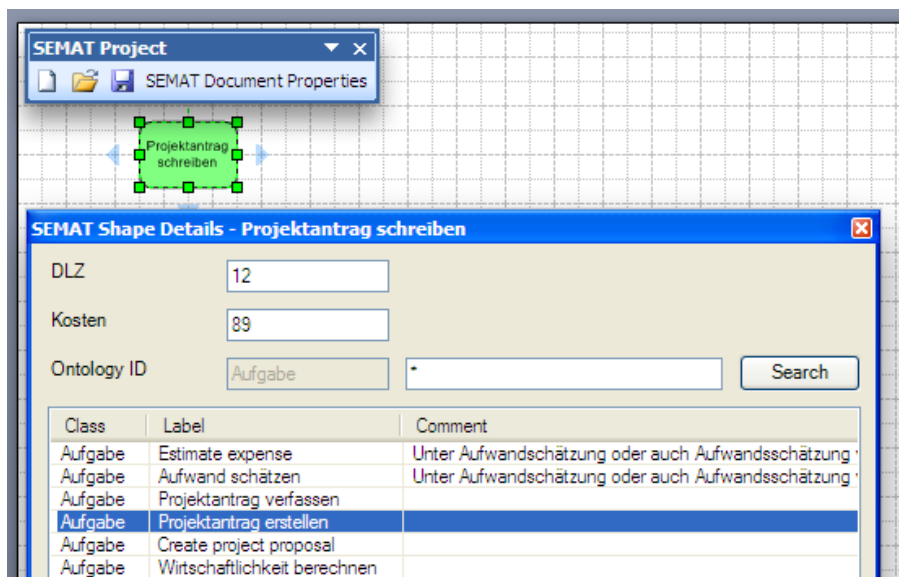


Abbildung 7: SEMAT Microsoft Visio Add-In

Eine Übersicht über die zur Verfügung stehenden Modellkonstrukte, die mittels des Microsoft Visio Add-Ins attribuiert werden können, gibt Tabelle 1.

Bezeichnung im Referenz-Metamodell	UML- Aktivitätsdiagramm	EPK
ProcessStep	Aktion	Funktion
Event	Objektknotenzustand, Entscheidungsbedingung	Ereignis
Start-/End-Event	Startknoten, Endknoten	Ereignis
ControlFlow	Kontrollfluss	Kontrollfluss
Connector	Entscheidungsknoten, Gabelung	OR-, XOR-, AND- Konnektor
Subprocess	Aktivität	Unterprozess

Tabelle 1: Übersicht der Modellkonstrukte

4.3 SEMAT Analysewerkzeug

Zur Analyse und zum Vergleich der Prozessmodelle wird eine prototypisch entwickelte Java-Software genutzt, die in Abbildung 8 als Komponentendiagramm dargestellt ist. Sie besteht aus den Komponenten „Modell-, Metrik- und Ontologieverwaltung“, „Datenbankzugriff“, „Metrikerzeugung und -bearbeitung“ sowie „Ontologieaufbereitung und -analyse“. Über die Komponente „Modell-, Metrik- und Ontologieverwaltung“ ist es möglich, Modelle, Metriken und Ontologien aus der Datenbank zu laden oder zu speichern, wozu die Komponente „Datenbankzugriff“ genutzt wird. Diese Komponente ist ein beliebiger *Object-Relational-Mapper* der jeden Entitätstypen der Datenbank mit einer Klasse verbindet. Somit können sehr leicht Entitäten aus der Datenbank, die ähnlich dem Referenz-Metamodell gespeichert wurden, als Instanzen von Java-Klassen erzeugt werden. Diese Instanzen wiederum werden zur Erzeugung der Analyseontologie genutzt. Durch die Komponente „Ontologieaufbereitung und -analyse“ ist es dem Anwender möglich diese Analyseontologie auszuwerten. Hierzu wird das *Semantic Web Framework* Jena [Jena] und die Inferenzmaschine Pellet [Pellet] genutzt.

Hinsichtlich der Auswertung der Analyseontologien stellt der Prototyp mehrere Möglichkeiten zur Berechnung von Kennzahlen zur Verfügung. Es können alle Möglichkeiten zur Auswertung von Ontologien mittels einer Inferenzmaschine genutzt werden sowie Abfragen mittels SPARQL erfolgen. SPARQL steht für *SPARQL Protocol and RDF Query Language* und wurde vom [W3C] am 18. Januar 2008 als *Recommendation* freigegeben. Es erlaubt Abfragen auf RDF Dokumenten in ähnlicher Weise wie SQL. Diese grundlegenden Auswertungsmöglichkeiten können weiterhin in Skripte zusammengeführt werden.

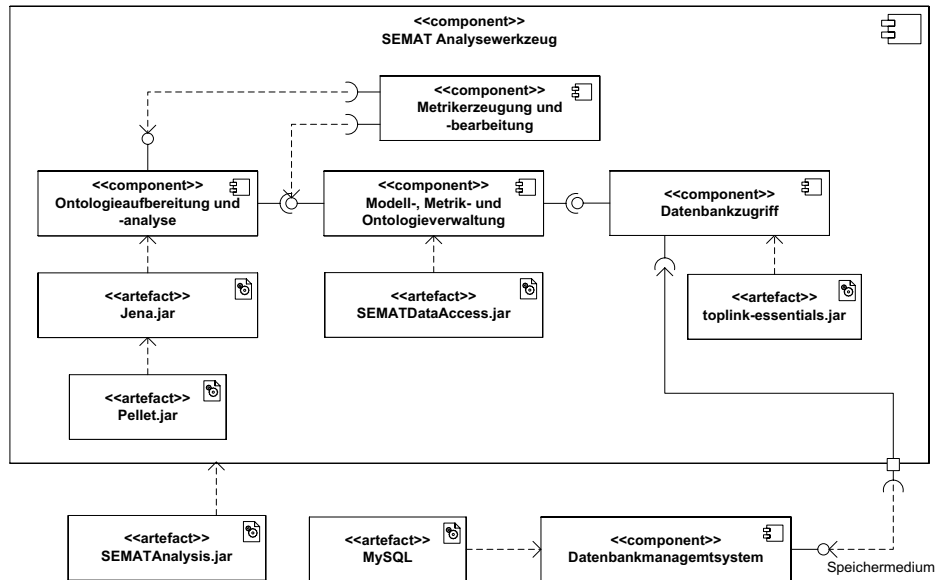


Abbildung 8: Komponenten des SEMAT Analysewerkzeugs

4.4 Analyse und Vergleich mittels SEMAT Analysewerkzeug

Die Analysemöglichkeiten der prototypischen Umsetzung werden anhand des bereits zuvor in Abbildung 3 dargestellten Beispiels demonstriert. Die Analyse einer vorher erzeugten Analyseontologie kann, wie oben erwähnt, auf zwei Arten sowie deren Kombination erfolgen. Zum einen durch die Erzeugung eines Skripts und zum anderen durch die Abfrage mittels SPARQL. Eine SPARQL-Abfrage kann auch in ein Skript mit eingebunden und ausgewertet werden.

In Abbildung 9 ist die Auswertung anhand eines Skripts zur Ermittlung des „Detaillierungsgrad“ der einzelnen Prozessmodelle dargestellt. Diese sehr einfache Metrik kann zum Modellvergleich herangezogen werden, da mit dieser Metrik gemessen werden kann, wie detailliert ein Konzept in einem Prozessmodell repräsentiert ist. Die Metrik besteht aus dem Quotienten aus referenzierten (Sub-)Konzepten und der Anzahl vorhandener (Sub-)Konzepte. In Hinblick auf das Konzept „Projektantrag erstellen“ ergibt sich hier für die ereignisgesteuerte Prozesskette ein Wert von 0,25, weil nur eines von vier Konzepten referenziert wird. In Hinblick auf das Aktivitätsdiagramm ergibt sich ein Wert von 0,75, weil drei von vier Konzepten referenziert werden. Dieses Ergebnis kann mittels Inferenz auf der Analyseontologie des Beispiels automatisch ermittelt und zum Prozessvergleich herangezogen werden. Ein Vergleich ist durch diese einfache Metrik insofern möglich, dass Teilbereiche identifiziert werden können, die in unterschiedlichen Detaillierungsgrad modelliert wurden. Sollten weiterhin Leistungsdaten, wie in Abbildung 3 dargestellt, bei der Analyse genutzt werden,

so können einzelne Teilbereiche auch anhand dieser Informationen miteinander verglichen werden.

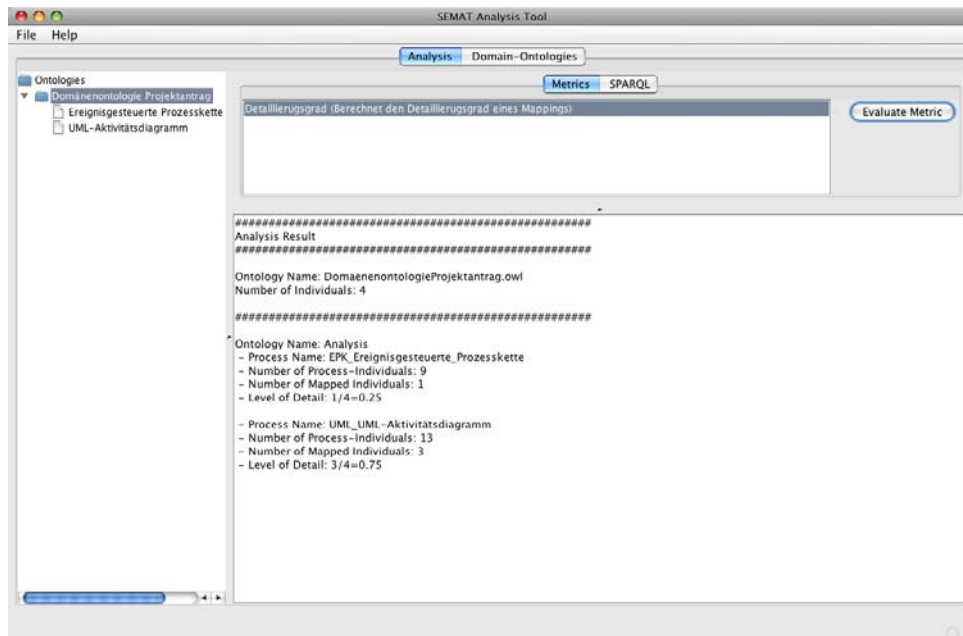


Abbildung 9: SEMAT Analysewerkzeug zur Ermittlung der Metrik "Detaillierungsgrad"

Im Rahmen der SEMAT-Entwicklung wurde ein Katalog von einfachen und fortgeschrittenen Metriken entwickelt, die sowohl den qualitativen als auch einen quantitativen Vergleich von Prozessmodellen erlauben. Zu den ersteren zählen bspw. Kennzahlen zum Detaillierungsgrad und zur „fachlichen Breite“ von Modellen, zu den ersteren bspw. Metriken zu Prozesskosten. Die Konstruktion der Metriken basiert im Wesentlichen auf etablierten Verfahren zur Vermessung von Graphen bzw. Objektmodellen (vgl. [GPC05]) oder auf etablierten betriebswirtschaftlichen Verfahren wie bspw. der Prozesskostenrechnung.

5 Nutzeneffekte im Bereich des Prozess-Benchmarking

Das in diesem Beitrag vorgestellte Verfahren kann dazu eingesetzt werden, die Durchführung von groß angelegten *Prozess-Benchmarking-Initiativen* zu halbautomatisieren und damit effizienter zu gestalten. Während bei einem klassischen *Prozess-Benchmarking* die Analyse und der Leistungsvergleich unterschiedlicher Prozesse durch Fachexperten zu erfolgen hat, können unter Nutzung des skizzierten Prototyps diese Aufgaben (halb-)automatisiert durch Software durchgeführt werden. Als Vorbedingung ist zu fordern, dass die Modelle anhand von semi-formalen Beschreibungssprachen (z.B. EPK oder UML-Aktivitätsdiagramm) dokumentiert sind. Das notwendige Mapping zu Konzepten der Domänenontologie kann entweder manuell oder unter Nutzung von

Mapping-Verfahren (z.B. auf Basis eines Thesaurus) erfolgen (vgl. [NM02]). Danach können die Modelle in Ontologien überführt und schließlich der Analyse unterzogen werden. Sofern entsprechende Metriken vorliegen, kann dann ein vollautomatisierter Vergleich der Modelle realisiert werden, der an Aussagekraft gewinnen kann, wenn zusätzlich Leistungsinformationen wie Durchlaufzeiten, Fehlerquoten oder Kosten (oder andere Leistungskennzahlen) vorliegen.

Mit dem vorgestellten Verfahren kann somit ein deutlicher Zeit- und Kostenvorteil realisiert werden. Allerdings darf dabei nicht außer Acht gelassen werden, dass der Aufbau eines entsprechenden Informationssystems, die Konstruktion (und fortlaufende Wartung) einer Domänenontologie sowie die Annotation der Modelle ebenfalls mit entsprechendem Aufwand verbunden sind. Daher kann von einer Wirtschaftlichkeit des Konzeptes nur dann ausgegangen werden, wenn der relativ aufwändigen Vorbereitung eine *vielfache* Nutzung der Modelle im Rahmen von Benchmarking-Vorhaben gegenübersteht. Eine solche vielfache Nutzung wird nur dann zu erzielen sein, wenn bspw. ein zentrales Benchmarking Clearing Center die dauerhafte Realisierung des Konzeptes übernimmt.

6 Fazit sowie Ausblick auf weitere Forschung

Die derzeitige prototypische Umsetzung des Verfahrens von [ATB06] zeigt die technische Realisierbarkeit. Eine Überführung von in Microsoft Visio 2007 modellierten Prozessmodellen in den Modellierungssprachen UML und EPK sowie deren Leistungsinformationen in eine so genannte Analyseontologie wurde dargestellt. Darauf aufbauend sind unter Zuhilfenahme einer ihr zugeordneten Domänenontologie automatisierte Schlussfolgerungen möglich. Diese Schlussfolgerungen können entweder benutzerdefiniert durch SPARQL-Abfragen oder durch Metriken erfolgen. Nach unserem Kenntnisstand handelt es sich um die erste Implementierung eines Werkzeugs dieser Art.

Der derzeitige Stand des Prototypen unterstützt die Wesentlichen Anforderungen des Verfahrens wie in Abschnitt 4 dargestellt in ausreichender Weise. Allerdings gibt es noch Einschränkungen im Bereich der Benutzungsfreundlichkeit sowie der Konfigurierbarkeit des Systems. Beide Problembereiche sollen in folgenden Versionen des Werkzeugs behoben werden. Insbesondere ist eine Unterstützung auch andere Informationsmodellarten (z.B. Datenmodelle) geplant. Hinzukommen soll weiterhin eine übersichtliche Auswertung von Modellen mit Hilfe von Berichten über verschiedene Metriken. Hierzu soll auch der Metrikkatalog erweitert werden. Auch soll es möglich werden beliebige Arten von Leistungsinformationen zu definieren und die Modelle mit entsprechenden Daten anzureichern. Die in diesem Beitrag vorgestellte Überführung der Metamodelle der EPK sowie des UML-Aktivitätsdiagramms in eine Ontologie, wurde in ähnlicher Weise von [RG02] durch Nutzung der Bunge-Wand-Weber (BWW-)Ontologie vorgestellt. Dies fand im Rahmen dieses Beitrags keine Berücksichtigung und soll daher im Rahmen der weiteren Forschungsarbeiten evaluiert und diskutiert werden.

Neben der technischen Weiterentwicklung des Systems soll eine umfassende Evaluierung des Ansatzes und des Werkzeugs in konkreten Projekten erfolgen. In einem

ersten Schritt soll einem Action Research-Ansatz folgend das Werkzeug in Benchmarking-Projekten eingesetzt und zyklisch verbessert werden. Dann soll eine praktische Evaluierung ohne Interaktion der Autoren durch fachliche Anwender durchgeführt werden. Die Wirtschaftlichkeit des Verfahrens soll mit Hilfe von Simulationsmodellen ermittelt werden.

Im Laufe der Entwicklung des Prototypen wurden Schwachstellen im Verfahren von [ATB06] erkannt und beseitigt. Die wichtigsten Änderungen haben sich bei der Nutzung von Ontologien ergeben. So war ursprünglich eine ausschließliche Nutzung einer Ontologie zur semantischen Beschreibung der Modelle vorgesehen. Im Laufe der Forschung hat sich jedoch gezeigt, dass eine Überführung des Referenz-Metamodells in eine Ontologie sowie eine ausschließliche Analyse auf der Analyseontologie zweckmäßig ist. So können die Möglichkeiten, welche durch Schlussfolgerungen gegeben sind, in einem größeren Ausmaß genutzt werden.

Als Anwendungsbereiche, in denen das vorgestellte Verfahren genutzt werden kann, sind neben dem Benchmarking zum Beispiel auch ein „*Information Model Mining*“ und eine (halb-)automatische Referenzmodellkonstruktion denkbar [ATB06, S. 28]. Unter „*Information Model Mining*“ verstehen [ATB06, S. 28] in Anlehnung an den Begriff des *Data Mining* ein Verfahren, mit denen aus einer Menge von Informationsmodellen automatisch neue Erkenntnisse gewonnen werden können. Bei der (halb-)automatischen Referenzmodellkonstruktion kann eine Informationsmodell-Datenbank eine sehr weitreichende Hilfe sein, indem auf Basis der mittels Datenbank dokumentierten, empirischen Fälle automatisch Vorschläge für Referenzmodellstrukturen generiert werden.

Literaturverzeichnis

- [ATB06] Ahlemann, F.; Teuteberg, F.; Brune, G.: Ontologiebasierte Attributierung von Informationsmodellen: Grundlagen und Anwendungsgebiete. ISPRI-Arbeitsbericht Nr. 01/2006, 2006.
- [AV03] Antoniou, G.; van Harmelen, F.: Web Ontology Language: OWL. In (Staab, S.; Studer, R. (Hrsg.)): Handbook on Ontologies. Springer, Berlin, 2003; S. 67-92.
- [Ba03] Baader F.; Calvanese D.; McGuinness D.; Patel-Schneider P.; Nardi D.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, Cambridge, 2003.
- [BKR05] Becker, J.; Kugeler, M.; Rosemann, M.: Prozessmanagement. Ein Leitfadens zur prozessorientierten Organisationsgestaltung. Springer, Berlin u.a., 2005.
- [Ce07] Celino, I.; de Medeiros, A. K. A.; Zeissler, G.; Oppitz, M.; Facca, F. M.; Zoeller, S.: Semantic Business Process Analysis. In (Hepp, M.; Hinkelmann, K.; Karagiannis, D.; Klein, R.; Stojanovic, N. (Hrsg.)): Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007), 3rd European Semantic Web Conference (ESWC 2007), CEUR Workshop Proceedings, 251, 2007; S. 44-47.
- [EKO07] Ehrig, M.; Koschmider, A.; Oberweis, A.: Measuring Similarity between Semantic Business Process Models. In (Roddick, J. F.; Hinze, A. (Hrsg.)): Proceedings of the Fourth Asia-Pacific Conference on Conceptual Modelling (APCCM 2007), Australian Computer Science Communications, 2007; S. 71-80.

- [FL07] Fettke, P.; Loos, P.: Ontological Evaluation of Scheer's Reference Model for Production Planning and Control Systems. *International Journal of Interoperability in Business Information Systems*, 2007, vol. 2, 1; S. 9-28.
- [GPC05] Genero, M.; Piattini, M.; Calero, C.: A Survey of Metrics for UML Class Diagrams. *Journal of Object Technology*, 2005, 4; 59-92
- [Gr93] Gruber, T. R.: A translation approach to portable ontology specifications. *Knowledge Acquisition*, 1993, vol. 5, 2; S. 199-220.
- [He04] Hevner, A. R.; March, S. T.; Park, J.; Ram, S.: Design Science in Information Systems Research. *MIS Quarterly*, 2004, vol. 28, 1; S. 75-105.
- [He05] Hepp, M.; Leymann, F.; Domingue, J.; Wahler, A.; Fensel, D.: Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management. *ICEBE '05: Proceedings of the IEEE International Conference on e-Business Engineering*, IEEE Computer Society, 2005; S. 535-540.
- [Hö07] Höfferer, P.: Achieving Business Process Model Interoperability Using Metamodels and Ontologies. In *Proceedings of the Fifteenth European Conference on Information Systems*, 2007; S. 1620-1631.
- [Ho05] Horrocks, I.: Applications of Description Logics: State of the Art and Research Challenges. *Proc. of the 13 th Int. Conf. on Conceptual Structures (ICCS '05)*, 2005; S. 78-90.
- [Ka06a] Kappel, G.; Kapsammer, E.; Kargl, H.; Kramler, G.; Reiter, T.; Retschitzegger, W.; Schwinger, W.; Wimmer, M.: On Models and Ontologies – A Semantic Infrastructure Supporting Model Integration. *Modellierung*, Innsbruck, 2006; S. 11-27.
- [Ka06b] Kappel, G.; Kapsammer, E.; Kargl, H.; Kramler, G.; Reiter, T.; Retschitzegger, W.; Schwinger, W.; Wimmer, M.: Lifting metamodels to ontologies – A Step to the Semantic Integration of Modeling Languages. *ACM/IEEE 9th International Conference on Model Driven Engineering Languages and Systems (MODELS '06)*. In (Nierstrasz, O. et al. (Hrsg.)): *LNCS 4199*. Springer, Berlin, Heidelberg, 2006; S. 528-542.
- [KH06] Karagiannis, D.; Höfferer, P.: Metamodels in action: An overview. In (Filipe, J.; Shishkov, B.; Helfert M. (Hrsg.)): *ICSOF 2006, First International Conference on Software and Data Technologies*, Setúbal, Portugal, September 11-14, 2006. INSTICC Press, 2006.
- [LÖ99] Legner, C.; Österle, H.: Prozessbenchmarking – Ein methodischer Ansatz zur Prozessentwicklung mit Standardsoftware. In (Scheer, A.; Nuettgens, M. (Hrsg.)): *4. Internationale Tagung Wirtschaftsinformatik*. Physica, Heidelberg, 1999; S. 331-352.
- [MK07] Murzek, M.; Kramler, G.: The Model Morphing Approach – Horizontal Transformations between Business Process Models. *Proceedings of the 6th International Conference on Perspectives in Business Information Research – BIR 2007*, Department of Computer Sciences, University of Tampere, Tampere, Finland, 2007; S. 88-103.
- [Jena] Jena – A Semantic Web Framework for Java, <http://jena.sourceforge.net/> (Zugriff am 26.05.2008)
- [NM02] Noy, N. F.; Musen, M. A.: Evaluating Ontology-Mapping Tools: Requirements and Experience. *Proceedings of OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management*, Siguenza, Spain, 2002; S. 1-14.
- [OWL] Web Ontology Language. <http://www.w3.org/TR/owl-features/> (Zugriff am 26.05.2008)
- [Pellet] Pellet OWL Reasoner. <http://pellet.owldl.com/> (Zugriff am 26.05.2008)
- [RG02] Rosemann, M.; Green, P.: Developing a meta model for the Bunge-Wand-Weber ontological constructs. *Information Systems*, vol. 27, 2, 2002; S. 75-91.

- [FL07] Fettke, P.; Loos, P.: Ontological Evaluation of Scheer's Reference Model for Production Planning and Control Systems. *International Journal of Interoperability in Business Information Systems*, 2007, vol. 2, 1; S. 9-28.
- [GPC05] Genero, M.; Piattini, M.; Calero, C.: A Survey of Metrics for UML Class Diagrams. *Journal of Object Technology*, 2005, 4; 59-92
- [Gr93] Gruber, T. R.: A translation approach to portable ontology specifications. *Knowledge Acquisition*, 1993, vol. 5, 2; S. 199-220.
- [He04] Hevner, A. R.; March, S. T.; Park, J.; Ram, S.: Design Science in Information Systems Research. *MIS Quarterly*, 2004, vol. 28, 1; S. 75-105.
- [He05] Hepp, M.; Leymann, F.; Domingue, J.; Wahler, A.; Fensel, D.: Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management. *ICEBE '05: Proceedings of the IEEE International Conference on e-Business Engineering*, IEEE Computer Society, 2005; S. 535-540.
- [Hö07] Höfferer, P.: Achieving Business Process Model Interoperability Using Metamodels and Ontologies. In *Proceedings of the Fifteenth European Conference on Information Systems*, 2007; S. 1620-1631.
- [Ho05] Horrocks, I.: Applications of Description Logics: State of the Art and Research Challenges. *Proc. of the 13 th Int. Conf. on Conceptual Structures (ICCS '05)*, 2005; S. 78-90.
- [Ka06a] Kappel, G.; Kapsammer, E.; Kargl, H.; Kramler, G.; Reiter, T.; Retschitzegger, W.; Schwinger, W.; Wimmer, M.: On Models and Ontologies – A Semantic Infrastructure Supporting Model Integration. *Modellierung*, Innsbruck, 2006; S. 11-27.
- [Ka06b] Kappel, G.; Kapsammer, E.; Kargl, H.; Kramler, G.; Reiter, T.; Retschitzegger, W.; Schwinger, W.; Wimmer, M.: Lifting metamodels to ontologies – A Step to the Semantic Integration of Modeling Languages. *ACM/IEEE 9th International Conference on Model Driven Engineering Languages and Systems (MODELS '06)*. In (Nierstrasz, O. et al. (Hrsg.)): *LNCS 4199*. Springer, Berlin, Heidelberg, 2006; S. 528-542.
- [KH06] Karagiannis, D.; Höfferer, P.: Metamodels in action: An overview. In (Filipe, J.; Shishkov, B.; Helfert M. (Hrsg.)): *ICSOF 2006, First International Conference on Software and Data Technologies*, Setúbal, Portugal, September 11-14, 2006. INSTICC Press, 2006.
- [LÖ99] Legner, C.; Österle, H.: Prozessbenchmarking – Ein methodischer Ansatz zur Prozessentwicklung mit Standardsoftware. In (Scheer, A.; Nuettgens, M. (Hrsg.)): *4. Internationale Tagung Wirtschaftsinformatik*. Physica, Heidelberg, 1999; S. 331-352.
- [MK07] Murzek, M.; Kramler, G.: The Model Morphing Approach – Horizontal Transformations between Business Process Models. *Proceedings of the 6th International Conference on Perspectives in Business Information Research – BIR 2007*, Department of Computer Sciences, University of Tampere, Tampere, Finland, 2007; S. 88-103.
- [Jena] Jena – A Semantic Web Framework for Java, <http://jena.sourceforge.net/> (Zugriff am 26.05.2008)
- [NM02] Noy, N. F.; Musen, M. A.: Evaluating Ontology-Mapping Tools: Requirements and Experience. *Proceedings of OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management*, Siguenza, Spain, 2002; S. 1-14.
- [OWL] Web Ontology Language. <http://www.w3.org/TR/owl-features/> (Zugriff am 26.05.2008)
- [Pellet] Pellet OWL Reasoner. <http://pellet.owldl.com/> (Zugriff am 26.05.2008)
- [RG02] Rosemann, M.; Green, P.: Developing a meta model for the Bunge-Wand-Weber ontological constructs. *Information Systems*, vol. 27, 2, 2002; S. 75-91.

- [Sc02] Scheer, A.-W.: ARIS – Vom Geschäftsprozeß zum Anwendungssystem. Springer, Berlin, Aufl. 4, 2002.
- [St98] Strahinger, S.: Ein sprachbasierter Metamodellbegriff und seine Verallgemeinerung durch das Konzept des Metaisierungsprinzips. In (Pohl, K.; Schürr, A.; Vossen, G. (Hrsg.)): Modellierung '98, Proceedings des GI-Workshops in Münster, 11.-13. März, 1998.
- [SMW07] Strommer, M.; Murzek, M.; Wimmer, M.: Applying Model Transformation By-Example on Business Process Modeling Languages Advances. In (Hainaut, J. et al. (Hrsg.)): Advances in Conceptual Modeling - Foundations and Applications, LNCS 4802. Springer, Berlin u.a., 2007; S. 116-125.
- [TF07] Thomas, O.; Fellmann, M.: Semantic Business Process Management: Ontology-Based Process Modeling Using Event-Driven Process Chains. International Journal of Interoperability in Business Information Systems, 2007, vol. 2, 1; S. 29-44.
- [VZS05] Vanderhaeghen, D.; Zang, S.; Scheer, A.-W.: Interorganisationales Geschäftsprozessmanagement durch Modelltransformation. In (Scheer, A.-W. (Hrsg.)): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Nr. 182, Saarbrücken, Universität des Saarlandes, 2005.
- [W3C] SPARQL Query Language for RDF. <http://www.w3.org/TR/rdf-sparql-query/> (Zugriff am 26.05.2008).

Ein Vorschlag zur Messung der Nutzung von Referenzprozessmodellen – Konzept und Anwendung am Beispiel des SCOR-Modells

Peter Fettke

Institut für Wirtschaftsinformatik (IWi)
Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)
Stuhlsatzenhausweg 3
66123 Saarbrücken
peter.fettke@iwi.dfki.de

Abstract: Referenzmodelle gelten als nützliche Instrumente, denen vielfältige Nutzenpotentiale bei der Gestaltung von Informationssystemen nachgesagt werden. Bisher ist allerdings weitgehend unbekannt, in welchem Umfang die Praxis Referenzmodelle tatsächlich einsetzt. Die vorliegende Untersuchung leistet am Beispiel des SCOR-Modells einen ersten Beitrag, um diese Erkenntnislücke zu schließen. Dabei wird zunächst ein Konzept entwickelt, das unterschiedliche Dimensionen der Nutzung eines Referenzprozessmodells in der Praxis herausarbeitet. Anschließend wird in Form einer Querschnittsanalyse ein empirisches Forschungskonzept vorgeschlagen, um den Nutzungsgrad des SCOR-Modells beim Supply Chain Management zu bestimmen.

1 Motivation und Problemstellung

Referenzmodelle werden seit Anfang der 1990er-Jahre in der Theorie und Praxis des Business Engineering als Hilfsmittel für die Entwicklung unternehmensspezifischer Modelle propagiert [FL04, S. 331]. Sie können als allgemeingültige Modelle in unterschiedlichen Kontexten wiederverwendet werden und unterstützen die Analyse, den Entwurf, die Implementierung und die Wartung von Informationssystemen [FB08]. In welchem Umfang Referenzmodelle in der Praxis tatsächlich eingesetzt werden, bleibt bisher weitgehend unbekannt. So finden sich beispielsweise in Untersuchungen, die die generelle Verbreitung von Ansätzen der Unternehmensmodellierung in der Praxis untersuchen, keine weiteren Hinweise zur Nutzung von Referenzmodellen [DGR06; Fe08].

Konkrete Hinweise zur tatsächlichen Nutzung von Referenzmodellen in der Praxis werden in der Literatur nur vereinzelt geäußert: [Sc97, S. V] stellt fest, dass das Y-CIM-Modell „inzwischen vielfach als Referenzmodell genutzt worden“ sei. [BS04, S. V] formulieren mit mehr Nachdruck, dass „das Handels-H-Modell .. seinen festen Platz in Aus- und Weiterbildung, in Softwarehäusern und bei Unternehmensberatern gefunden“

haben. [LK98, S. 2 - 7] halten fest, dass „[d]as Aachener PPS-Modell .. bereits mehrfach erfolgreich in der betrieblichen Praxis eingesetzt werden“ konnte. [St01, S. 471] stellt heraus, „[that the SCOR model] is .. being successfully applied to improve business operations in North America, Latin America, Europe, Asia and Australia/New Zealand.“

Auch wenn die obigen Aussagen Hinweise zur Nutzung verschiedener Referenzmodelle geben, zeigt indes eine nähere Analyse der vorliegenden Arbeiten zwei Defizite. Erstens finden sich keine Begründungen für die vorgelegten Aussagen zur Nutzung der entsprechenden Referenzmodelle. Damit sind die Aussagen für Dritte nicht prüfbar und intersubjektiv zugänglich. Zweitens wird aus den Arbeiten nicht ersichtlich, was unter der Nutzung eines Referenzmodells zu verstehen sei. Mit anderen Worten ist bisher kein Instrument bekannt, mit dem der Umfang bzw. der Grad der Nutzung eines Referenzmodells bestimmt werden kann.

Der vorliegende Beitrag zielt darauf ab, die aufgezeigten Erkenntnisdefizite zu schließen. Im Einzelnen sollen dabei zwei Ziele verfolgt werden:

1. Es soll ein erster Vorschlag zur Messung der Nutzung von Referenzprozessmodellen entwickelt werden.
2. Es wird erläutert, wie der Vorschlag zur Messung der Nutzung von Referenzprozessmodellen anhand des SCOR-Referenzmodells exemplarisch anzuwenden ist.

Der Beitrag besitzt sowohl eine theoretische als auch eine praktische Relevanz. Bisher wird zumeist in der Literatur davon ausgegangen, dass ein Referenzmodell entweder genutzt oder nicht genutzt wird. Es wird also von einem klassifikatorischen Nutzungsbegriff ausgegangen (siehe zur Unterscheidung zwischen klassifikatorischen und metrischen Begriffen [St74, S. 16f.]). Allerdings ist es denkbar, dass Referenzmodelle nicht vollständig, sondern nur zum Teil in einem Unternehmen eingesetzt werden. Um unterschiedliche Nutzungsgrade eines Referenzmodells unterscheiden zu können, ist der Begriff der Nutzung eines Referenzmodells nicht klassifikatorisch, sondern metrisch einzuführen. Der Übergang von klassifikatorischen zu metrischen Begriffen ist verbunden mit einem höheren Informationsgehalt und bildet damit einen theoretischen Fortschritt [St74, S. 16f.]. Präzisere Begriffe können dazu genutzt werden, um präzisere theoretische Aussagen zu formulieren.

Präzisere Begriffe dienen aber auch der Praxis. Modellierer können formulieren, in welchem Umfang ein Referenzmodell genutzt wird. Damit ist eine präzisere Kommunikation zwischen den an der Modellierung beteiligten Personen möglich. Die Planung, Steuerung und Überwachung von Modellierungsprozessen erhält wertvolle Informationen, in welchem Umfang ein Referenzmodell genutzt wird oder werden soll. So kann beispielsweise formuliert werden, dass in einem Unternehmen ein bestimmter Nutzungsgrad eines Referenzmodells erreicht werden soll. Zu einem späteren Zeitpunkt kann überprüft werden, ob der angestrebte Nutzungsgrad tatsächlich erreicht worden ist. Wenn dagegen nur ein klassifikatorischer Begriff der Nutzung vorliegt, dann lassen sich unterschiedliche Nutzungsgrade eines Referenzmodells nicht unterscheiden. Model-

lierungsprozesse können dann nur auf einer groben Informationsbasis geplant, gesteuert und überwacht werden.

Forschungsmethodisch orientiert sich die vorliegende Arbeit an einem ingenieurwissenschaftlichem Vorgehen und basiert auf einem pluralistischen Forschungsansatz [Fr06]. Ausgehend von einem theoretisch wie praktisch relevanten Problem, für das noch keine befriedigende Lösung vorliegt, wird ein neuer Lösungsvorschlag unterbreitet. Die Anwendbarkeit und Nützlichkeit der neuen Problemlösung werden anhand einer konkreten Anwendung aufgezeigt. Im Rahmen der Anwendung kommen empirische Forschungsmethoden zum Einsatz.

Die Arbeit ist wie folgt gegliedert: Nach dieser Einführung wird im nächsten Abschnitt das SCOR-Modell als Anwendungsbeispiel kurz erläutert. Der Hauptteil der Untersuchung besteht aus den Abschnitten 3 und 4. Dabei beschreibt Abschnitt 3 unterschiedliche Dimensionen der Nutzung eines Referenzprozessmodells. Abschnitt 4 beschreibt eine mögliche Untersuchungskonzeption zur Anwendung des Vorschlags. Der Beitrag schließt mit einem Ausblick auf sich anschließende Forschungsfragen in Abschnitt 5.

2 Einführung in das SCOR-Modell als Anwendungsbeispiel

Der entwickelte Vorschlag zur Messung der Nutzung eines Referenzmodells soll in diesem Beitrag nicht nur abstrakt entwickelt werden, sondern auch anhand eines konkreten Referenzmodells exemplarisch angewendet werden. Dabei basiert die Anwendung auf dem SCOR-Modell. Dieses Modell wurde als Anwendungsbeispiel aufgrund seiner relativ hohen Reife und Verbreitung ausgewählt.

Das SCOR-Modell definiert fünf unterschiedliche Prozesstypen, deren Zusammenhang anhand einer mehrstufigen Supply Chain in Abbildung 1 visualisiert wird [Sc08]:

- Planen (*plan*): Dieser Prozess umfasst die Planung und das Management des Angebots an und der Nachfrage nach Gütern.
- Beschaffen (*source*): Dieser Prozess umfasst den Bezug von Waren, den Wareneingang, die Eingangskontrolle, die Lagerung und die Zahlungsanweisung für sämtliche Güter.
- Herstellen (*make*): Dieser Prozess umfasst sämtliche Schritte der Produktionsdurchführung.
- Liefern (*deliver*): Dieser Prozess umfasst sämtliche Schritte der Bestellung und Auslieferung von Gütern zum Kunden.
- Rückliefern (*return*): Dieser Prozess umfasst sämtliche Schritte zur Abwicklung von zurückgesendeten Gütern, wobei sowohl Reparaturen als auch Wartungen berücksichtigt werden.

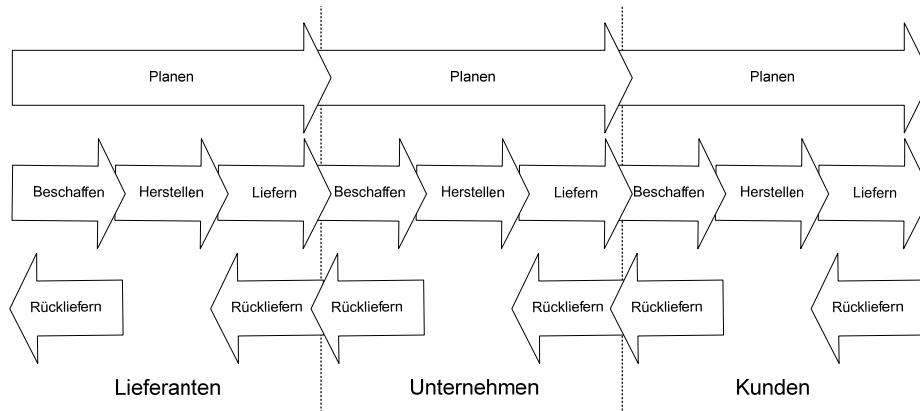


Abbildung 1: Zusammenhang zwischen den Prozessstufen des SCOR-Modells

Die fünf Prozessstufen werden im SCOR-Modell auf insgesamt vier Modellierungsebenen beschrieben, die unterschiedliche Detaillierungsgrade von Supply Chains erfassen [Sc08]:

1. Ebene: Die oberste Modellierungsebene umfasst die fünf Prozessstufen.
2. Ebene: Jeder Prozessstyp wird auf der zweiten Ebene in unterschiedliche Prozesskategorien verfeinert. Hierbei werden in Abhängigkeit des Prozessstyps unterschiedliche Kriterien verwendet. Beispielsweise werden die Prozessstufen „Beschaffen“, „Herstellen“ und „Liefern“ im Hinblick auf Lager- und Kundenauftragsfertigung unterschieden.
3. Ebene: Jede Prozesskategorie wiederum wird auf dieser Ebene in einzelne Prozesselemente untergliedert. Zwischen den Prozesselementen werden Ablaufreihenfolgen und Datenflussbeziehungen festgelegt.
4. Ebene: Zu jedem Prozesselement kann auf dieser Ebene eine Menge von durchzuführenden Aktivitäten definiert werden. Diese Aktivitäten sind nicht vom SCOR-Modell standardisiert, sondern können beziehungsweise müssen unternehmensspezifisch festgelegt werden.

Im Rahmen des Anwendungsbeispiels soll untersucht werden, inwieweit das SCOR-Modell in der betrieblichen Praxis zum Supply Chain Management genutzt wird. Für diesen Zweck wird zunächst ein Vorschlag zur Messung der Nutzung eines Referenzprozessmodells entwickelt.

3 Konzept zur Messung der Nutzung eines Referenzprozessmodells

Der Begriff der Nutzung oder Anwendung eines Referenzmodells wird in der vorliegenden Literatur zur Referenzmodellierung implizit vorausgesetzt und in einem intuitiven

Sinn verstanden. So lassen sich keine Aussagen darüber finden, was genau unter der Nutzung eines Referenzmodells zu verstehen ist und wie diese theoretisch oder empirisch erfasst werden kann. Zwar finden sich insbesondere in der englischsprachigen Literatur vielfältige Ausführungen zur allgemeinen Nutzung von Informationssystemen. Diese sind allerdings noch nicht auf den Kontext der Referenzmodellierung übertragen worden und es bleibt offen, wieweit dies auch möglich ist [VDR07; VMD03]. Für die vorliegende Untersuchung ist daher ein Konzept zu entwickeln, das festlegt, wie die Nutzung eines Referenzmodells zu erfassen ist.

Vordergründig kann die Nutzung eines Referenzmodells als ein klassifikatorischer Begriff eingeführt werden. Demnach wird ein Referenzmodell entweder genutzt oder nicht genutzt. Wenn dieser Auffassung gefolgt wird, muss eine Reihe von Indikatoren identifiziert werden, die anzeigen, wann von einer Referenzmodellnutzung auszugehen ist.

Der Informationsgehalt eines klassifikatorischen Nutzungsbegriffs ist relativ gering, da nicht unterschieden werden kann, ob ein Unternehmen alle oder nur ausgewählte Aspekte eines Referenzmodells nutzt. Hierzu ein Beispiel: Es sei angenommen, ein Referenzmodell umfasse 20 Prozesse, von denen in einem Unternehmen nur die Hälfte berücksichtigt werden. Wenn dagegen in einem anderen Unternehmen sämtliche 20 Prozesse des Referenzmodells genutzt werden, ist der Nutzungsgrad des Referenzmodells in diesem Unternehmen höher einzustufen. Derartige Unterschiede hinsichtlich der Nutzung eines Referenzmodells können von einem klassifikatorischen Nutzungsbegriff nicht erfasst werden.

Folglich erscheint es sinnvoll, den Begriff der Nutzung eines Referenzmodells nicht klassifikatorisch, sondern metrisch einzuführen, um unterschiedliche Nutzungsgrade theoretisch unterscheiden zu können. Der Übergang von klassifikatorischen zu metrischen Begriffen ist verbunden mit einem höheren Informationsgehalt und stellt einen theoretischen Fortschritt in der Wissenschaft dar [St74, S. 16f.]. Auf diese Weise ergibt sich die Möglichkeit, den Nutzungsgrad eines Referenzmodells detaillierter zu erfassen.

Im Folgenden ist daher eine metrische Konzeptualisierung der Nutzung eines Referenzmodells zu entwickeln. Die unten vorgestellte Konzeptualisierung wurde von einer Untersuchung zum *Electronic Data Interchange* inspiriert [MZ96, siehe insbesondere S. 335]. Ausgehend vom dort unterbreiteten Vorschlag zur Konzeptualisierung der Nutzung von *Electronic Data Interchange* wird die Nutzung eines Referenzmodells durch folgende Dimensionen definiert:

Breite der Nutzung

Die Breite der Nutzung eines Referenzmodells wird definiert als der Grad der Nutzung der einzelnen Geschäftsprozesse, die durch das Referenzmodell bestimmt werden. Beispielsweise wird im Kontext des SCOR-Modells die Breite in Abhängigkeit davon definiert, in welchem Umfang in einem Unternehmen die durch das SCOR-Modell spezifizierten Geschäftsprozesse „Planen“, „Beschaffen“, „Herstellen“, „Lieferten“ und „Rückliefern“ genutzt werden.

Detaillierung der Nutzung

Referenzmodelle verfügen meist über unterschiedliche Abstraktionsebenen, die jeweils eine unterschiedliche Detaillierung besitzen. Die Detaillierung beschreibt, in welchem Umfang die unterschiedlichen Abstraktionsebenen bei der Anwendung des Referenzmodells berücksichtigt werden. Beispielsweise definiert das SCOR-Modell Geschäftsprozesse auf unterschiedlichen Abstraktionsstufen, wobei die ersten drei Ebenen durch das Referenzmodell vorgegeben sind und die vierte Ebene für unternehmensspezifische Erweiterungen vorgesehen ist. Mit zunehmender Nutzung weiterer Modellierungsebenen wird von einer höheren Detaillierung der Nutzung des SCOR-Modells ausgegangen.

Tiefe der Nutzung

Mit der Tiefe der Modellnutzung wird beschrieben, wie viele Geschäftspartner bei der Nutzung des Referenzmodells eingebunden sind. Werden keine weiteren Geschäftspartner bei der Anwendung des Referenzmodells berücksichtigt, wird von einer geringen Nutzungstiefe gesprochen. Mit steigender Anzahl der Geschäftspartner ist von einer höheren Nutzungstiefe des Referenzmodells auszugehen. Im Zusammenhang mit dem SCOR-Modell beschreibt die Tiefe der Nutzung des SCOR-Modells, wie viele Geschäftspartner und Kunden in der Nutzung des SCOR-Modells eingebunden sind. Wenn in dem Modell keine Geschäftspartner eingebunden sind, ist die Nutzungstiefe des Referenzmodells gering. Mit zunehmender Berücksichtigung weiterer Geschäftspartner entlang der Supply Chain steigt die Nutzungstiefe des SCOR-Modells.

Volumen der Nutzung

Ein Referenzmodell kann für verschiedene Aufgaben im Unternehmen Gestaltungsvorschläge unterbreiten. Die Gestaltungsvorschläge müssen aber letztlich nicht eingehalten werden, sondern können aufgrund von Ad-hoc-Entscheidungen umgangen werden. Das Volumen beschreibt, in welchem Umfang sämtliche Geschäftsvorfälle tatsächlich nach den Vorgaben des Referenzmodells abgewickelt werden. Ein hohes Volumen ergibt sich dann, wenn sämtliche Geschäftsvorfälle gemäß den Prozessvorgaben des Referenzmodells abgewickelt werden. Dies bedeutet beispielsweise beim SCOR-Modell, dass sämtliche Lieferantenanfragen gemäß den Modellvorgaben realisiert werden.

Sprachgebrauch der Nutzung des SCOR-Modells

Ein Referenzmodell umfasst einen begrifflichen Rahmen, der sich nicht nur in den implementierten Abläufen und Handlungsvorgaben zeigt, sondern auch die im Unternehmen verwendeten Begrifflichkeiten prägt [Fr07, S. 125, 129, 131]. Folglich beeinflusst die Nutzung eines Referenzmodells das Kommunikationsverhalten und den Sprachgebrauch der Mitarbeiter. Der Sprachgebrauch gibt Auskunft, wie intensiv die Begrifflichkeiten des Referenzmodells von den Mitarbeitern in einem Unternehmen tatsächlich verwendet werden. Hierbei werden sowohl die (fern-)mündliche als auch die elektronische Kommunikation berücksichtigt. Ebenso spielt auch die Kommunikation mit den Geschäftspartnern eine Rolle. Im Kontext des SCOR-Modells bedeutet beispielsweise ein hoher Sprachgebrauch, dass die Begriffe „Planen“, „Beschaffen“, „Herstellen“, „Lieferten“ und „Rückliefern“ von allen Mitarbeitern gemäß den Vorgaben

des SCOR-Modells gebraucht werden. Das Vorhandensein einer gemeinsamen Terminologie ist im Supply Chain Management bei der Nutzung des SCOR-Modells von Bedeutung [Ku05, S. 353].

4 Untersuchungskonzeption einer möglichen Anwendung

Im Folgenden wird aufgezeigt, wie das zuvor eingeführte Konzept zur Messung der Nutzung des SCOR-Modells angewendet werden kann. Mit Hilfe der exemplarischen Anwendung wird gezeigt, in welchem Umfang das SCOR-Modell in der Praxis genutzt wird. Dabei ist ausschließlich die Nutzung des SCOR-Modells zum Supply Chain Management im Unternehmen von Interesse. Andere Anwendungsgebiete des SCOR-Modells wie die Entwicklung von Software-Systemen für das Supply Chain Management, die Beratung anderer Unternehmen oder Nutzungsmöglichkeiten des Modells für die Forschung und Lehre sind explizit von der vorliegenden Untersuchung ausgeschlossen.

Im Rahmen der folgenden Darstellung der Untersuchungskonzeption werden Forschungsdesign, Datenbasis, Art der Datenerhebung und Operationalisierung der Variablen erläutert. Zum allgemeinen Vorgehen möglicher empirischer Untersuchungskonzeptionen siehe beispielsweise [SHE05].

4.1 Forschungsdesign

Um den realen Nutzungsgrad des SCOR-Modells zu erheben ist eine deskriptive Felduntersuchung zu wählen. Dieses Untersuchungsdesign kann hinsichtlich der Anzahl der in der Untersuchung betrachteten Unternehmen und hinsichtlich der Anzahl der Messpunkte weiter differenziert werden (siehe Abbildung 2). In der vorliegenden Untersuchung wird eine Querschnittanalyse vorgeschlagen, bei der bei mehreren Unternehmen die Nutzung des SCOR-Modells zu einem Zeitpunkt erfasst wird. Bei diesem Forschungsdesign können folglich die Nutzungsgrade des SCOR-Modells in mehreren Unternehmen vergleichend analysiert werden, eine zeitliche Analyse der Nutzung des SCOR-Modells ist allerdings nicht möglich.

		Anzahl der Unternehmen	
		ein Unternehmen	mehrere Unternehmen
zeitlicher Umfang	ein Zeitpunkt	Einzelfallstudie	<i>Querschnittanalyse</i>
	mehrere Zeitpunkte	singuläre Längsschnittanalyse	multiple Längsschnittanalyse

Abbildung 2: Abgrenzung des vorgeschlagenen Forschungsdesigns (Darstellung in Anlehnung an [Ku75, S. 62])

4.2 Datenbasis

Gegenstand der Untersuchung sind Unternehmen, die das SCOR-Modell im Rahmen des Supply Chain Management in der eigenen Organisation einsetzen. Die Bestimmung der Größe der Grundgesamtheit stößt auf Schwierigkeiten, da nach dem Kenntnisstand des Verfassers keine (amtliche) Statistik existiert, aus der die Anzahl der Nutzer des SCOR-Modells hervorgeht.

Da es kein Verzeichnis aller Nutzer des SCOR-Modells gibt, die Grundgesamtheit somit nicht bekannt ist, bereitet auch das Ziehen einer repräsentativen Stichprobe Schwierigkeiten. Folgende Überlegungen zeigen einen Lösungsweg für dieses Problem auf. Grundsätzlich ist davon auszugehen, dass nur diejenigen Unternehmen eine ernsthafte Anwendung des SCOR-Modells vornehmen können, denen das SCOR-Modell bekannt ist. Unternehmen, denen nicht das SCOR-Modell bekannt ist, werden dieses nicht anwenden können. Des Weiteren erhält ein Unternehmen Zugriff auf das SCOR-Modell, wenn es Mitglied des *Supply-Chain Council* ist. Damit bilden die Mitglieder des *Supply-Chain Council* eine empirische Basis für eine mögliche Untersuchung.

4.3 Datenerhebung

Für die Datenerhebung stehen unterschiedliche Erhebungsverfahren zur Verfügung [SHE05, S. 321]. Bei einer Beobachtung ist fraglich, ob die ausgewählten Unternehmen einen Zugang zur Organisation gewähren würden. Ferner wäre eine intensive Beobachtung in größeren Fallzahlen nicht realistisch. Beispielsweise wäre es notwendig, Beobachtungen in Organisationen in verschiedenen Ländern vorzunehmen.

Eine Inhaltsanalyse erscheint ebenso nicht angemessen. Abgesehen von der Frage, ob die notwendigen Informationen überhaupt aus Dokumenten erschlossen werden können, erfordert diese Erhebungsart einen Zugriff auf eine Vielzahl unternehmensspezifischer Dokumente. Es ist davon auszugehen, dass in vielen Fällen die untersuchten Unternehmen Dritten keinen Zugriff auf die erforderlichen Dokumente einräumen. Folglich könnten in vielen Fällen die benötigten Daten nicht erhoben werden.

Es verbleibt die Befragung, die sich hinsichtlich der Art ihrer Durchführung unter anderem in eine mündliche und schriftliche Form unterscheiden lässt. Für eine mögliche Untersuchung wird eine schriftliche Befragung vorgeschlagen.

4.4 Operationalisierung des Nutzungsbegriffs

Bisher wurde der Nutzungsbegriff von Referenzmodellen implizit klassifikatorisch gebraucht. In dieser Untersuchung wird dagegen ein metrischer Nutzungsbegriff motiviert, wobei der Nutzungsgrad des SCOR-Modells anhand der fünf Dimensionen „Breite“, „Detaillierung“, „Tiefe“, „Volumen“ und „Sprachgebrauch“ konzeptualisiert worden ist. Eine Operationalisierung dieser Dimensionen liegt bisher noch nicht vor. Im Folgenden wird folgende Operationalisierung vorgeschlagen.

Breite der Nutzung des SCOR-Modells

Die Breite der Nutzung beschreibt die Nutzung der einzelnen im Referenzmodell definierten Geschäftsprozesse. Im SCOR-Modell sind dies die fünf Geschäftsprozesse „Planen“, „Beschaffen“, „Herstellen“, „Liefiern“ und „Rückliefern“. Der Implementierungsgrad der einzelnen Geschäftsprozesse wird auf einer 7-Punkt-Likert-Skala bestimmt, wobei der Skalenpunkt „1“ gar keine und der Skalenpunkt „7“ eine volle Implementierung des jeweiligen Geschäftsprozesses repräsentiert.

Detaillierung der Nutzung des SCOR-Modells

Die Detaillierung der Nutzung des SCOR-Modells beinhaltet, wie weit die unterschiedlichen Abstraktionsebenen des SCOR-Modells bei der Anwendung genutzt werden. Im SCOR-Modell werden vier Detaillierungsebenen unterschieden, wobei die erste bis dritte Ebene vom Referenzmodell festgelegt sind und die vierte Ebene zur unternehmensspezifischen Modellierung genutzt werden kann. Da letztlich jeder der fünf Geschäftsprozesse „Planen“, „Beschaffen“, „Herstellen“, „Liefiern“ und „Zurückliefern“ unterschiedlich detailliert implementiert werden kann, wird eine separate Operationalisierung gewählt. Dabei wird jeweils direkt ermittelt, bis zu welcher Modellebene eine Implementierung des SCOR-Modells im Unternehmen vorgenommen wurde. Es wird eine 5-Punkt-Likert-Skala verwendet, wobei der Skalenpunkt „1“ gar keine Implementierung und der Skalenpunkt „5“ eine Implementierung bis zur vierten Modellebene bedeutet. Dieser Aspekt wird für jeden der fünf Geschäftsprozesse separat ermittelt.

Tiefe der Nutzung des SCOR-Modells

Die Tiefe der Nutzung des SCOR-Modells beschreibt, wie weit verschiedene Geschäftspartner bei der Nutzung des SCOR-Modells involviert sind. Konzeptionell können Geschäftspartner in die drei Gruppen Lieferanten, Kunden und Logistikdienstleistern unterschieden werden. Auf einer 7-Punkt-Likert-Skala wird erfasst, wie weit die Geschäftspartner bei der Implementierung des SCOR-Modells berücksichtigt werden, wobei der Skalenpunkt „1“ keine Berücksichtigung und der Skalenpunkt „7“ eine sehr hohe Berücksichtigung bedeutet.

Des Weiteren ist zu berücksichtigen, dass im Supply Chain Management nicht nur die direkten Geschäftspartner, sondern auch die Geschäftspartner der Geschäftspartner et cetera eine Rolle spielen. Da bisherige empirische Befunde zeigen, dass Geschäftspartner, die innerhalb der Supply Chain ausgehend vom betrachteten Unternehmen in zweiter oder höherer Stufe stehen, im Supply Chain Management weniger intensiv oder gar nicht betrachtet werden [Fe07, S. 425], wird die Operationalisierung in der vorliegenden Untersuchung ab der dritten und höheren Ebene nicht detailliert aufgeschlüsselt.

Volumen der Nutzung des SCOR-Modells

Das Volumen der Nutzung des SCOR-Modells beschreibt, in welchem Umfang verschiedene Geschäftsprozesse gemäß den Vorgaben des Referenzmodells bearbeitet werden. Betrachtet wird jeweils der prozentuale Anteil der Geschäftsvorfälle pro Geschäftsprozess, die gemäß den Vorgaben des SCOR-Modells bearbeitet werden. Da eine genaue Quantifizierung schwierig ist, wird eine 5-Punkt-Likert-Skala gewählt, wobei der Skalenpunkt „1“ bedeutet, dass 0 Prozent der entsprechenden Geschäftsvorfälle gemäß den Vorgaben des SCOR-Modells, und der Skalenpunkt „5“ bedeutet, dass 100 Prozent der Geschäftsvorfälle gemäß den Vorgaben des SCOR-Modells abgewickelt werden. Dieser Aspekt wird für jeden der fünf Prozesse des SCOR-Modells separat erfasst.

Sprachgebrauch der Nutzung des SCOR-Modells

Bisher liegen keine Operationalisierungen für den Sprachgebrauch eines Referenzmodells vor. Die in der Untersuchung gewählte Erhebung betrachtet die subjektive Einschätzung des Befragten. Der Sprachgebrauch wird über drei Aspekte bestimmt, die sowohl die innerbetriebliche wie überbetriebliche Kommunikation als auch die Mensch-Maschine-Kommunikation betreffen. Hierbei wird jeweils auf einer 7-Punkt-Likert-Skala ermittelt, in welchem Umfang typische Begrifflichkeiten des SCOR-Modells bei der Kommunikation zwischen Mitarbeitern, bei der Kommunikation mit Geschäftspartnern und bei der Nutzung von Softwaresystemen gebraucht werden. Der Skalenpunkt „1“ bedeutet gar kein Gebrauch und der Skalenpunkt „7“ bedeutet einen ausschließlichen Gebrauch der Terminologie des SCOR-Modells.

5 Ausblick

In diesem Beitrag wird ein Konzept zur Messung der Nutzung von Referenzprozessmodellen vorgeschlagen und seine Anwendung am Beispiel des SCOR-Modells demonstriert. Im Anschluss an die vorliegende Untersuchung stellt sich eine Reihe weiterer Fragen, die in zukünftigen Arbeiten zu beantworten sind:

- Welche Ergebnisse können im Rahmen der vorgeschlagenen Forschungskonzeption erzielt werden?
- Existieren weitere Facetten, die die Nutzung eines Referenzprozessmodells ausmachen? Wie kann die vorgeschlagene Konzeptualisierung validiert werden?

- Inwiefern kann der vorgestellte Vorschlag zur Messung der Nutzung eines Referenzprozessmodells auf andere Referenzmodellarten, insbesondere Referenzdatenmodelle übertragen werden?
- Wie kann die Nutzung von Modellen im Allgemeinen bestimmt werden?
- Inwieweit gelingt es, Zusammenhänge zwischen dem Grad der Nutzung eines Referenzmodells und anderen wichtigen Konstrukten der Modellierung herzustellen? Besteht beispielsweise ein Zusammenhang zwischen der Nutzung und dem Erfolg der Referenzmodellierung?

Literaturverzeichnis

- [BS04] Becker, J.; Schütte, R.: Handelsinformationssysteme – Domänenorientierte Einführung in die Wirtschaftsinformatik. 2. Aufl., Frankfurt a. M. 2004.
- [DGR06] Davies, I.; Green, P.; Rosemann, M.; Indulska, M.; Gallo, S.: How do practitioners use conceptual modeling in practice? In: Data & Knowledge Engineering 58 (2006), S. 358-380.
- [FB08] Fettke, P.; vom Brocke, J.: Referenzmodell. In: K. Kurbel; J. Becker; N. Gronau; E. J. Sinz; L. Suhl (Hrsg.): Enzyklopädie der Wirtschaftsinformatik – Online-Lexikon. <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/>. München 2008.
- [Fe07] Fettke, P.: Supply Chain Management: Stand der empirischen Forschung. In: Zeitschrift für Betriebswirtschaft 77 (2007) 4, S. 417-461.
- [Fe08] Fettke, P.: Ansätze der Informationsmodellierung und ihre betriebswirtschaftliche Bedeutung: Eine Untersuchung der Modellierungspraxis in Deutschland. In: Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung (2008), im Druck.
- [FL04] Fettke, P.; Loos, P.: Referenzmodellierungsforschung. In: Wirtschaftsinformatik 46 (2004) 5, S. 331-340.
- [Fr06] Frank, U.: Towards a Pluralistic Conception of Research Methods in Information Systems Research. Institut für Informatik und Wirtschaftsinformatik (ICB) der Universität Duisburg-Essen, ICB-Research Report No. 7. Essen 2006.
- [Fr07] Frank, U.: Evaluation of Reference Models. In: P. Fettke; P. Loos (Hrsg.): Reference Modeling for Business Systems Analysis. Hershey, PA, USA, et al. 2007, S. 118-140.
- [Ku05] Kurbel, K.: Produktionsplanung und -steuerung im Enterprise Resource Planning und Supply Chain Management. 6. Aufl., München, Wien 2005.
- [Ku75] Kubicek, H.: Empirische Organisationsforschung – Konzeption und Methodik. Stuttgart 1975.
- [LK98] Luczak, H.; Kees, A.: Das Aachner PPS-Modell. In: J. Becker; W. Eversheim; H. Luczak; P. Mertens (Hrsg.): Referenzmodellierung '98 – Anwendungsfelder in Theorie und Praxis, 14. Juli 1998, RWTH Aachen. Aachen 1998, S. 2-1 bis 2-9.
- [MZ96] Massetti, B.; Zmud, R.: Measuring the Extent of EDI Usage in Complex Organizations: Strategies and Illustrative Examples. In: MIS Quarterly (1996), S. 331-345.
- [Sc08] Supply-Chain Council Inc.: Supply Chain Operations Reference Model (SCOR), Version 8.0. www.supply-chain.org, Abruf 2008-04-06 (Dokument nur für Mitglieder zugänglich).
- [Sc97] Scheer, A.-W.: Wirtschaftsinformatik – Referenzmodelle für industrielle Geschäftsprozesse. 7. Aufl., Berlin et al. 1997.
- [SHE05] Schnell, R.; Hill, P. B.; Esser, E.: Methoden der empirischen Sozialforschung. 7. Aufl., München, Wien 2005.

- [St01] Stephens, S.: Supply Chain Operations Reference Model Version 5.0: A New Tool to Improve Supply Chain Efficiency and Achieve Best Practices. In: Information Systems Frontiers 3 (2001) 4, S. 471-476.
- [St74] Stegmüller, W.: Probleme und Resultate der Wissenschaftstheorie und Analytischen Philosophie, Band II, Theorie und Erfahrung, Erster Teilband, Begriffsformen, Wissenschaftssprache, empirische Signifikanz und theoretische Begriffe. 2. Aufl., Berlin et al. 1974.
- [VDR07] Venkatesh, V.; Davis, F. D.; Morris, M. G.: Dead Or Alive? The Development, Trajectory And Future Of Technology Adoption Research. In: Journal of the Association for Information Systems 8 (2007) 4, S. 267-286.
- [VMD03] Venkatesh, V.; Morris, M. G.; Davis, G. B.; Davis, F. D.: User Acceptance of Information Technology: Toward a Unified View. In: MIS Quarterly 27 (2003) 3, S. 425-478.

Integration der Finanz- und Produktionsdomäne nach ANSI/ISA-95 in einem REA-System

Walter S.A. Schwaiger , Robert Ranzi

Institut für Managementwissenschaften - Abteilung für Finanzwirtschaft und Controlling
Technische Universität Wien
Favoritenstrasse
A-1040 Wien
schwaiger@imw.tuwien.ac.at
ranzi@imw.tuwien.ac.at

Abstract: Das Ziel dieser Arbeit ist die Präsentation einer Erweiterung des ANSI/ISA-95 Modells, mit deren Hilfe der Standard als Grundlage eines integrierten betrieblichen Informationssystem in Anlehnung an das REA-Modell verwendet werden kann. Der Standard ist in seiner Grundfassung auf die Modellierung der Produktionssteuerung und –kontrolle eines Industriebetriebes beschränkt. Wir präsentieren eine Erweiterung des Standards um finanzielle Aspekte. Diese Erweiterung ermöglicht es die finanziellen Bestände und Flüsse, die bei Ereignissen im *acquire-convert-market*-Ablauf eine Rolle spielen, zu thematisieren.

1 Einleitung

Die Modellierung und Entwicklung von betrieblichen Informationssystemen erfordert für produzierende Unternehmen die Berücksichtigung sowohl des Liquiditäts-Finanzbereichs als auch des technisch-organisatorischen Bereichs. Über die Aspekte der einfachen Modellierung hinaus muss der Verbindung dieser zwei Bereiche Aufmerksamkeit geschenkt werden.

Der ANSI/ISA-95¹ Standard bildet eine umfassende Grundlage zur prozessorientierten Modellierung der Datenstruktur eines Industriebetriebes². Die Orientierung an der Produktionssteuerung und –kontrolle führt dazu, dass finanzielle Aspekte im ANSI/ISA-95 Standard nicht betrachtet werden.

¹ www.isa-95.com

² Siehe dazu die Dokumente des Standards [IS00], [IS01] und [IS05].

Dies bringt es mit sich, dass Daten aus der Produktion, d.h. Mengen- und Preisinformation, nicht monetären Größen gegenübergestellt werden können. So ist es nicht möglich, die Wirkung von Ereignissen zu abbilden, die nicht mit der Produktion zusammenhängen: zum Beispiel sind der erste und der letzte Teil im *acquire-convert-market*-Ablauf (d.h. Einkauf und Verkauf) im Grundmodell des Standards nicht abbildbar. Wir präsentieren ein Modell finanzieller Ressourcen, welches die Bestände und Flüsse monetären Größen im Unternehmen modelliert.

2 Unternehmensmodellierung nach REA mit ANSI/ISA-95

Unsere Modellierung folgt den Konzepten des REA-Modells von McCarthy [Mc82] und des erweiterten REA-Modells von Geerts und McCarthy [GM00]. Die Elemente, mit denen wir die REA-Struktur befüllen, sind die Modelle des ANSI/ISA-95 Standards.

2.1 Das REA-Modell

Das Ressourcen-Ereignisse-Agenten-Modell (REA-Modell) beschreibt die unternehmerische Tätigkeit als Abfolge von ökonomischen Tauschhandlungen. Es wurde von McCarthy 1982 publiziert [Mc82] und in den folgenden Jahren weiterentwickelt.³

Die im REA-Framework betrachteten Tauschhandlungen bestehen, abstrakt betrachtet, aus folgenden wesentlichen Komponenten.

- Ereignisse: ökonomische Unternehmensereignisse, die dem Tausch zugrunde liegen.
- Agenten: Zu jedem Ereignis gehören (interne und/oder externe) Agenten, welche die zum Ereignis gehörende Tätigkeit ausführen.
- Doppik: Bei jedem ökonomischen Ereignis kommt es zu einem Austausch von ökonomischen Ressourcen.

Interne Ereignisse

Als interne Ereignisse bezeichnen wir die Umwandlung von Inputfaktoren zu Output im Zuge eines Produktionsprozesses. Dabei werden durch Verbrauch von Ressourcen (benötigte Arbeitszeit, verbrauchte Rohstoffe, maschinelle Abnutzung) neue Ressourcen geschaffen (Endprodukt). Es werden also durchwegs reale Ressourcen (im Gegensatz zu finanziellen Ressourcen) in reale Ressourcen umgewandelt (Feld a in Abbildung 1).

³ Siehe z.B. [GM99] sowie [GM00] und [GM02].

Externe Ereignisse

Externe Ereignisse sind im Allgemeinen durch den Tausch von realen Ressourcen gegen Finanzressourcen gekennzeichnet. Solche Ereignisse sind zum Beispiel der Einkauf von Ressourcen (Feld c in Abbildung 1) oder der Verkauf von Produkten (Feld b in Abbildung 1).

Bei der Durchführung von Finanztransaktionen kommt es zu einem Tausch von Finanzressourcen (z.B. Kreditaufnahme oder Veranlagung in einem Zerobond). Solche reinen Finanztransaktionen werden in Abbildung 1 durch das Feld d dargestellt.

	Reale Ressourcen	Finanz-Ressourcen
Reale Ressourcen	a	b
Finanz-Ressourcen	c	d

Abbildung 1: Doppik aus Ressourcensicht

2.2 ANSI/ISA-95 und REA

ANSI/ISA-95 ist ein Standard zur Integration von Enterprise- und Ausführungssystemen. Der Standard bietet deshalb nur die Möglichkeit, die Effekte von Ereignissen abzubilden, welche reale Ressource betreffen. Zum Beispiel ist es mit dem ANSI/ISA-95 Standard möglich, die Ressourcentransformation von Input zu Output zu dokumentieren (Feld a in Abbildung 1). Betriebliche Ereignisse, die sowohl reale als auch finanzielle Ressourcen verändern (Felder b, c und d in Abbildung 1), können mit dem Standard nicht abgebildet werden. Wir stellen ein Modell finanzieller Ressourcen vor, das in seiner Modellierung an den ANSI/ISA-Standard angelehnt ist. Unser Modell kann als Erweiterung des ANSI/ISA-Standards gesehen werden, mit dessen Hilfe auch Ereignisse dokumentiert werden können, die finanzielle Ressourcen verändern.

Datenmodelle, welche eine Unternehmensrealität abstrakt darstellen, bilden die Grundlage für betriebliche Informationssysteme. Dabei werden die Daten im Allgemeinen durch die erweiterte unternehmerische Tätigkeit erzeugt und konsumiert. Die erweiterte unternehmerische Tätigkeit umfasst neben den eigentlichen Produktionsprozessen auch die Prozesse, welche unmittelbar mit der Produktion zusammenhängen (z.B. Planungs- und Auswertungsprozesse) sowie Prozesse, welche die gesammelten Informationen verarbeiten und/oder in der Folge steuernd auf das Gesamtunternehmen einwirken:

- [Sc02] Scheer, A.-W.: ARIS – Vom Geschäftsprozeß zum Anwendungssystem. Springer, Berlin, Aufl. 4, 2002.
- [St98] Strahinger, S.: Ein sprachbasierter Metamodellbegriff und seine Verallgemeinerung durch das Konzept des Metaisierungsprinzips. In (Pohl, K.; Schürr, A.; Vossen, G. (Hrsg.)): Modellierung '98, Proceedings des GI-Workshops in Münster, 11.-13. März, 1998.
- [SMW07] Strommer, M.; Murzek, M.; Wimmer, M.: Applying Model Transformation By-Example on Business Process Modeling Languages Advances. In (Hainaut, J. et al. (Hrsg.)): Advances in Conceptual Modeling - Foundations and Applications, LNCS 4802. Springer, Berlin u.a., 2007; S. 116-125.
- [TF07] Thomas, O.; Fellmann, M.: Semantic Business Process Management: Ontology-Based Process Modeling Using Event-Driven Process Chains. International Journal of Interoperability in Business Information Systems, 2007, vol. 2, 1; S. 29-44.
- [VZS05] Vanderhaeghen, D.; Zang, S.; Scheer, A.-W.: Interorganisationales Geschäftsprozessmanagement durch Modelltransformation. In (Scheer, A.-W. (Hrsg.)): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Nr. 182, Saarbrücken, Universität des Saarlandes, 2005.
- [W3C] SPARQL Query Language for RDF. <http://www.w3.org/TR/rdf-sparql-query/> (Zugriff am 26.05.2008).

Semantic Business Process Management: An Empirical Case Study

Sebastian Stein^{1*}, Christian Stamber^{1*}, Marwane El Kharbili¹, Paweł Rubach²

¹ IDS Scheer AG

Altenkesseler Str. 17

66115 Saarbrücken, Germany

(sebastian.stein,christian.stamber,marwane.elkharbili)@ids-scheer.com

² Telekomunikacja Polska S. A., Research & Development Branch

ul. Obrzeźna 7

02-691 Warszawa, Poland

pawel.rubach@telekomunikacja.pl

* Both authors contributed equally to this work.

Abstract: Semantic technologies promise fully leveraging the content of enterprise and business process models by applying reasoning techniques to query the process space or using ontological mappings for bridging the business-IT divide. Such an approach is known as semantic business process management.

We conducted an empirical case study to explore semantic business process management. The case study was replicated 13 times with 17 participants from 8 different industrial and scientific organisations following a strict case study methodology to ensure validity and reliability of the results. The results show that participants embrace the possibilities of semantic technologies, but there are still open problems. In this paper, we present the case study design and discuss the results achieved.

1 Introduction and Related Work

Business process management (BPM) [SS08] comprises defining, documenting, communicating, designing, implementing, and monitoring business processes. A business process associates value-adding activities across internal and external organisational and functional boundaries [SS08, pp. 63]. Business processes are derived from a company's strategy and are aligned to customer needs and market conditions.

Even though the modelling languages used in BPM are public standards, the meaning of the models created with those standards is not available to machines. Here, the use of semantic technologies like ontologies, reasoners, and semantic web services as envisioned by Hepp et al. [HLD⁺05] promises enabling machines to understand business process models. This is known as semantic business process management (sBPM).

The idea of sBPM has gained momentum in recent years. Hepp and Roman [HR07a] suggest a stack of ontologies to represent the various aspects of business processes and enterprise models. However, sBPM will only be adopted in industry if it builds on existing

models and languages allowing an easy transition. Following this idea, Abramowicz et al. [AFKK07] create a semantic representation of the BPMN modelling language, which allows relating BPMN activities to WSMO goals [FLP⁺06]. In a similar effort, Nitzsche et al. [NWvL07] provide an ontologised version of BPEL. Again, BPEL activities can be related to WSMO goals enabling semantic web service discovery during process execution. Dimitrov et al. [DSSK07] implemented a semantic business process modelling tool based on WSMO Studio, which allows modelling semantic business processes using semantic BPMN. Besides process design and execution, semantics are also used for process analysis [CAmZ⁺07] and process mining [AdMPvdA⁺07]. Semantic compliance management [EKSMP08] is a more holistic approach ensuring compliance of semantic enterprise models with laws and regulations.

This overview shows sBPM is an active research area. So far, most efforts are fundamental research either defining the necessary languages [HR07a, AFKK07, NWvL07] or overall approaches [HLD⁺05, CAmZ⁺07, AdMPvdA⁺07, EKSMP08]. However, empirical evaluations of the proposed technologies to validate the practical relevance of sBPM are still missing. Such work would provide feedback, which could be incorporated in research agendas to realign the research efforts with requirements from the industrial field.

In this paper we¹ present first work providing this missing feedback through an empirical case study conducted in the telecommunication domain. A prototype of a sBPM suite was provided to practitioners, consultants, and researchers not involved in sBPM research. In addition, a tutorial describing step-by-step how to use the prototype was provided. After conducting the tutorial, participants were interviewed and asked to reflect on the usage of semantics in BPM. This paper describes and discusses the case study's results.

2 Research Design

2.1 Research Method and Research Question

While designing our research, we first defined our research question as follows: Which benefits do today's sBPM technologies provide to BPM experts and what obstacles exist hindering the adoption in industry?

Our research interests were of explorative nature. According to Yin [Yin03], controlled experiments as well as case studies are possible research methods to answer *how* and *why* research questions. Kitchenham et al. [KPP95] add that experiments are usually used for *research-in-the-small* and case studies for *research-in-the-typical*. We were clearly focused on research-in-the-typical, since our intention was to investigate the usage of sBPM in real-world settings. Kitchenham et al. also state that case study research is often used to evaluate new technologies. This also applied to our case.

We decided to conduct an empirical case study following the case study research methodology defined by Yin [Yin03]. We augmented Yin's methodology with ideas taken from

¹Research is part of the integrated research project SUPER financed by the EU Commission.

Kitchenham et al. [KPP95], because they describe specific practices for case study research in software engineering. According to Yin's methodology, after defining the research question one has to identify the case study propositions.

2.2 Case Study Propositions

Research questions are usually too abstract and too broad to be answered in a case study [Yin03]. Therefore, case study propositions (i. e. hypothesis) are defined to exactly know what to look at, to be aware of own preconceptions, and to prevent biased research.

We focused on the top-down approach of business process automation. The use of semantics in this area promises a simplified generation of executable business process models, because an ontological mapping between the business and IT domain exists on a meta-level. Using data mediators suggests simplifying data handling, because semantically equal but structural different data objects can be transformed automatically. We expected that identifying appropriate services to automate functions is easier using semantics instead of selecting services manually. In contrast, we expected that business experts currently cannot motivate this use case. Services represent business partners, who are contracted carefully and not dynamically allocated at runtime. Another obstacle concerns the definition of the needed ontologies, because the business environment might evolve faster than ontologies can be adapted. Also, adding semantics to the enterprise computing stack means an increased learning curve and additional middleware to be supported.

Abstracting from the different concrete propositions shows that we were sceptical about whether the use of semantics in BPM will pay off. Adding semantics will certainly simplify or even remove certain steps, but new steps are added. We estimated that an investment into semantics cannot be justified economically.

2.3 Unit of Analysis and Analysis of Results

Yin [Yin03] requests to clearly define the unit of analysis meaning to define the case study's case. As defined in the research question, we were interested in the benefits of semantics in BPM. In our case the unit of analysis was business process automation. We prepared two tutorials guiding a user through the process of implementing a business process. One tutorial covered the approach taken today without semantics. We call this the non-semantic tutorial or approach. The second tutorial used semantics and is therefore called semantic tutorial or approach. Both tutorials are based on business processes taken from real-world projects. The case of our case study comprises conducting at least the semantic tutorial if the participant is already familiar with the non-semantic approach.

We gathered experience gained by the participants through semi-structured interviews based on 18 open-ended questions. The interviewees were asked to describe what they have done, how non-semantic and semantic approach differ, and to reflect on the usage of semantics. At the beginning of each interview, we elaborated on the background of our

research effort and explained that the interview results are made anonymous and not publicly available assuring privacy. We emphasised that we are not trying to prove or disprove semantics as beneficial. We allowed participants to ask questions as well. The interviews were not recorded but instead conducted by two researchers. One researcher led the interview and the other researcher focused on taking notes. Each interviewer wrote a small summary immediately after the interview and both summaries were then exchanged. We also collected work artefacts such as the semantically annotated business process models to have multiple sources of evidence [Yin03].

3 Case Study Process

3.1 Overview

In this section we describe the different components of the case study. Before we started the case study, we prepared the semantic tutorial, the example business process, and semantically enabled prototypes. To ensure that all replications of the case study and all interviews are run and analysed in the same way, we created a case study protocol [Yin03]. The case study protocol defines the overall research goal, case study propositions, and unit of analysis. It also defines the questions to be asked during the interview, it describes how to conduct the study, and how to analyse the results. It also names possible participants and outlines the structure of the result report. To further increase the validity of our case study, we applied the case study checklist described in [HR07b].

3.2 Non-Semantic and Semantic Tutorial

The case study is based on a non-semantic and a semantic tutorial. The non-semantic tutorial is available in two versions featuring two different business processes. One business process was taken from the automotive domain and the other one belongs to the e-government domain. The tutorial uses ARIS to model the business process with the EPC notation. Each function is annotated with a software service using ARIS' service discovery functionality. Afterwards, the business process is transformed by the tool into BPEL. Additional manual refinements are needed before the BPEL process can be deployed and executed on Oracle BPEL Process Server. The tutorial package consists of a database for ARIS, a set of implemented web services, and a detailed manual. The non-semantic tutorial was not prepared for the case study, but existed already before [SKD⁺08].

The semantic tutorial was prepared for the case study. It consists of semantically extended version of ARIS and uses Oracle BPEL Process Server (see section 3.4). A detailed step-by-step instruction explaining how to install the necessary extensions and how to conduct the tutorial was provided to the participants. The different parts of the semantic tutorial are described in the following subsections.

3.3 VoIP Activation Process of Telekomunikacja Polska

The business process used in the semantic tutorial was contributed by Telekomunikacja Polska (TP). TP is the dominant player in the Polish telecommunications market serving 10.6 million fixed-line subscribers and over 12 million mobile customers, employing about 28.000 people (as of Q1/2007).

The voice-over-IP (VoIP) ordering process was used in the tutorial, because it is a rather complex one involving internal and external parties. The business process is illustrated in Fig. 1 using a simplified EPC notation. Most events and all semantic annotations were removed from the process model so that it fits on one page. The VoIP ordering process allows TP's customers to order the VoIP service for an existing contract. The process is initiated by the customer through TP's web portal. After identifying the customer, the process first checks if all technical and formal requirements are fulfilled. A new order is created, which must be confirmed by the customer. A check is run to see if the customer already has the necessary hardware. If not, the hardware is sent together with the contract to the customer. After TP receives the signed contract, the contract is archived, the billing system is activated, and finally the VoIP service is activated. An ontology and belonging semantic web services existed already and were reused [FRS07].

3.4 Semantically Enabled Prototypes

ARIS was used for modelling the semantic business process. The tool was extended as described in [SSEK08b]. Each function is annotated by a WSMO goal using an prototypical graphical user interface. On the left side of the screen, the user selects a goal. On the right side the belonging WSML description is shown. The WSML files on the left have no speaking names in order to force the participant to look at the WSML code. After confirming the selection of the goal, the WSMO goal and the ontological input/output instances are added to the function. The WSMO goals are later used during process execution to discover matching services. Another extension was developed allowing users to complete the data flow in the process model by mapping ontological instances produced by a function to the input of a later occurring function. The existing EPC to BPEL transformation was adapted, too. In case of a function annotated with a WSMO goal, a BPEL variable with the WSMO goal as value is created and a proxy service is invoked.

Today's process execution environments are not able to use semantic descriptions like WSMO goals to discover services during runtime. To still allow using semantic descriptions in the executed BPEL process, we implemented a proxy service. This proxy service consumes the semantic description through a standard web service interface and invokes a semantic execution environment to do semantic service discovery. The discovered semantic web service is invoked by the proxy service as well. A detailed description of our approach to service discovery during process execution can be found in [SSEK08a].

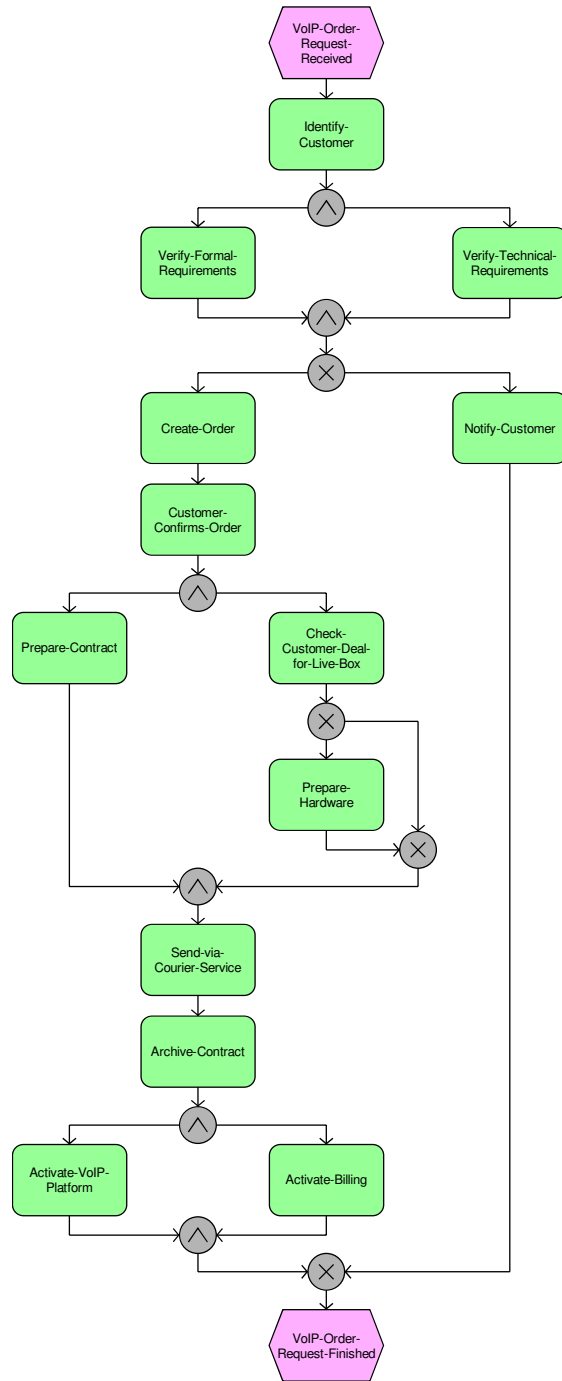


Figure 1: VoIP Ordering Process of Telekomunikacja Polska

Table 1: Participants

Type of Organisation	Organisations	Interviews	Participants
Research Consulting Institute	2	2	3
University	1	2	2
University of Applied Sciences	2	4	4
Company	3	5	8
Sum:	8	13	17

3.5 Participants

There were participants from different organisations (see table 1). Research consulting institutes are research institutes, which are not solely financed through the public, but also offer commercial consulting services like the German Fraunhofer institutes. We distinguish between university and university of applied sciences, because the latter one focuses on practical application in contrast to theoretical education. None of the participants was part of the case study’s research team and most of them had no prior knowledge of semantic technologies. We provided no additional material for background reading to the participants besides the semantic tutorial. We paid special attention that the participants do not get aware of our own preconception of sBPM and the case study propositions.

Participants were selected by the authors based on their BPM experience and their availability. At least 75% of the participants have more than 5 years of experience in BPM by either working as BPM experts or teaching advanced classes on BPM.

4 Results and Discussion

4.1 Overview

This section presents the case study results and a discussion of the outcomes. The results must be viewed as hypothesis to be investigated by future research activities. This section is structured around the main interview points and the most interesting discussions we had. The answers between participants from the different types of organisation were consistent if not discussed otherwise.

4.2 Understanding Semantics

At the beginning of each interview, we asked participants to summarise the different steps of the tutorial. Most participants were able to name the steps and their order. During this summary, most of them used the word “semantic”. We asked them how they understand

semantics and how they define the term ontology.

All participants said a semantic description is not a technical one, but instead business oriented. Interestingly, some of the participants said that a semantic description defines only what needs to be done but not how to achieve it. None of the participants provided one of the popular definitions of ontology like “shared conceptualisation”. Instead, all participants tried to describe what an ontology is like pointed out that an ontology is a collection of terms, concepts or classes. Some of them used the term “glossary” or “taxonomy”, but only a few called an ontology a “namespace”, a “domain”, a “classification” or a “domain specific language”. Some participants said an ontology not only defines terms, but also relationships between them. For example, one participant said an ontology describes “what exists and how everything is related to each other” and another said it is a “model of the world”. One participant pointed out that an ontology standardises the vocabulary used. Interestingly, some participants also talked about “business cases” while actually referring to concepts. However, only a few pointed out that an ontology is processable by machines.

None of the participants seemed to be comfortable with the term “ontology”, because the term is not known from daily language usage, and seems artificial to them. *We postulate that one should not use the term ontology while talking to business experts, but instead talk about “semantics” or “semantic descriptions”.* To give a more detailed definition, one should talk about a glossary of business terms, which also has detailed relationships between terms in contrast to ordinary glossaries. One should also point out that semantic descriptions of services are business oriented, processable by computers, and used to describe what needs to be done, and not how it should be implemented.

4.3 Getting Familiar With the Domain

The domain ontology developed by TP was presented to the reader. All participants confirmed to have studied it at the beginning of the tutorial, but only a few of them used it later. The example process was still simple enough to understand and the terminology used was also known to the participants. Many participants pointed out that it is unclear to them where the domain ontology comes from and who creates it. The domain ontology was only available in the printed tutorial, but it was not part of the tool. This was confusing for some participants, because they expected the ontology to be present in the tool, too. Many of them pointed out that for more complex ontologies an “ontology browser” is required to allow easy navigation between the different concepts. *We postulate that having a domain ontology is useful even if no other semantic technologies are used. Such an ontology must be integrated in the business process modelling tool allowing easy access and navigation.*

4.4 Visualisation of Ontologies

The domain ontology was presented in three different ways to the participants: a “star” of the main concepts generated by WSMO Studio, an UML class diagram with a class

for each concept plus the belonging attributes and the main relationships, and the WSML code.

If participants were familiar with UML modelling like the participants not working in a company, they found the UML class diagram most useful. Participants said that the UML class diagram contains far more information compared to the star diagram. If participants were not familiar with UML, they preferred the star, because it provides an easy to understand overview of the domain ontology. All participants said the WSML code is not useful and readable. Some of them noted that it might be possible to understand the WSML syntax after training, but that it is definitely not useful for business experts. One participating business expert confirmed that by saying he refuses to look at “something” like the WSML code.

We postulate that a graphical representation along with a textual description of a domain ontology is required. Probably several graphical representations are necessary allowing the user to select the preferred one.

4.5 Selecting WSMO Goals

One important step of the tutorial was selecting a goal for each function. It turned out that all participants just identified the name of the WSMO Goal in the WSML code and based their decision mostly on the name. Only a few of them looked at additional details of the goal description such as pre-/postconditions. However, most participants recognised they must use the pre-/postconditions when goal selection is ambiguous.

Most participants were not satisfied with goal selection. Many pointed out that browsing a list of goals does not scale and more advanced search mechanisms are required. A participant suggested that it must be possible to filter the list of goals based on concepts taken from the ontology. Another participant proposed using the pre-/postconditions as filter criteria. It was also suggested to add a graphical representation for each goal. One participant suggested the visualisation shown in Fig. 2.

We postulate that goal selection is an important part and must be supported by a sophisticated tool. This requirement is amplified, because some participants pointed out that they cannot see any advantage compared to selecting a web service directly. Therefore, *research should focus on ways to graphically visualise goals and semantic descriptions and evaluate the usefulness through empirical experiments.*

4.6 Completing the Data Flow

After selecting a goal, participants completed the data flow by mapping output instances of a function to input instances of a later function. Even though all participants were able to complete this step, some concerns were raised. Participants pointed out that input/output instances are similar to variables, whereas business objects are normally used in business

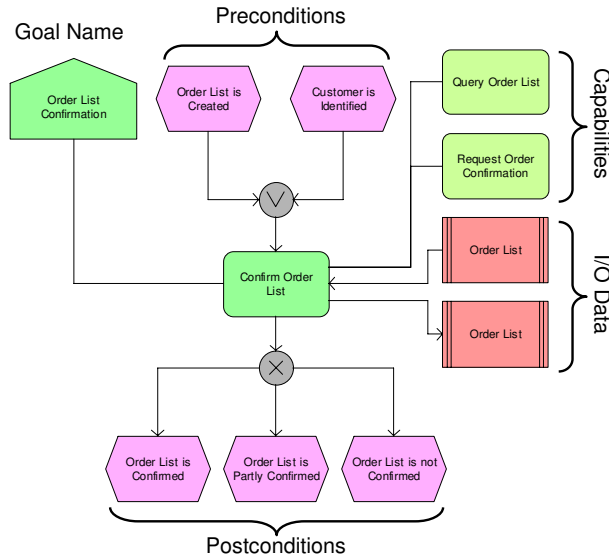


Figure 2: Participant Contribution: Graphical Representation of a WSMO Goal

process modelling. According to participants, those two concepts are not interchangeable, because a business object is always persistent whereas a variable must be stored in a data store explicitly. This is an interesting point, which must be further investigated. *It seems that ontological instances defined by WSMO Goals are not abstract enough to be useful in business process modelling.*

4.7 Motivating Service Binding During Runtime

We asked participants to motivate dynamic service binding during process execution. We received a diverse set of answers with no clear conclusion. However, some answers proved that in contrast to our case study proposition some participants were able to motivate service binding during runtime. We explicitly asked participants for an economic motivation. If they provided such a motivation, they often mentioned failover scenarios. In our opinion, this problem can be already solved today with enterprise service bus (ESB) platforms, but we did not confront participants with our point of view. Participants said that instead of hard coding an endpoint URL into the executable process, only the service name is added to the process. This helps in case the service is moved to another server. Again, this seems to be a case where today's technologies such as service registries can be applied.

Some participants noted dynamic service binding only makes sense if there are several services for each goal. If there is only a 1 : 1 relationship between service and goal, participants were not able to justify dynamic service binding. Participants with research background also pointed out that dynamic service binding in a company might not be as

relevant as e. g. in ubiquitous computing, because a company is able to better control and govern the portfolio of services. Other participants mentioned dynamic service binding is dangerous, because it adds a new error source and increases the complexity of the enterprise computing stack. This shows *there is no consensus whether dynamic service binding is necessary in business process automation.*

4.8 Advantages and Disadvantages of Semantic Approach

We asked participants about advantages and disadvantages of the semantic approach. Surprisingly, most of them mentioned *a better separation of business and IT as the main advantage of the semantic approach, because the business process model is not polluted with technical details* such as web service descriptions. Instead, the business expert only specifies the required capabilities. This helps business experts to concentrate on the business relevant part of process modelling instead of dealing with implementation details. It also allows using not yet existing services. In addition, technical service descriptions (WSDL) do not have to be available in the business process modelling tool, which prevents redundancy. Participants characterised the semantic approach as creating a process template, which can be flexibly enacted.

A significant problem is the conceptual mismatch between business objects and ontological input/output instances. Besides, several participants were not convinced that the investment in semantics will pay off, because ontologies must be defined and maintained. *Some participants were reluctant about ontology modelling, because in their opinion similar efforts such as establishing an enterprise data model failed in the past. All participants agreed that business experts are not able to create ontologies.* Graphical tools are required to overcome hurdles like WSML syntax and logical expressions. Using semantics might require having ontology engineers, which is a specific qualification. In general, *the learning curve is increased, because semantics bring their own set of technologies, methods, and methodologies along. Also, the complexity of the enterprise computing stack is increased,* which augments the probability of introducing errors and integration problems. Many participants pointed to the unbalanced distribution of efforts and benefits for using semantics as another major disadvantage. *Ontologies, goals, and semantic descriptions must be defined by IT after consulting business experts, but those artefacts mainly help business people. This discrepancy must be carefully managed to ensure close cooperation between all involved parties.*

4.9 Feasibility of Semantic Approach for Business Experts

We asked participants whether the semantic approach is possible to follow for business experts. Most of them agreed that it is feasible, but they also mentioned problems. Currently, technology is still too visible as discussed before.

Some participants were surprised that they were asked to annotate functions with semantic

descriptions. They believed that a business process model already contains enough information. Those participants envisioned a more advanced way of using semantics in BPM. For example, one participant desired to have a repository of semantically described process fragments. Instead of defining the control flow of the business process, the participant expected to just define the pre- and postconditions as well as constraints and the control flow would be automatically created by an “intelligent component”. It will be interesting to see if such visionary approaches to sBPM will gain momentum.

5 Summary

We presented our findings of an empirical case study exploring the use of semantics in BPM. The case study was designed and conducted according to a strict methodology ensuring high validity and reliability. The case study was replicated several times using a real-world business process and typical end-user tools. Our results show that semantics promise moving BPM forward, but there are still many open questions. All results found are hypothesis, which must be validated by future research activities. Future research will show if the problems explored can be solved in order to leverage the advantages of sBPM.

References

- [AdMPvdA⁺07] A. K. Alves de Medeiros, C. Pedrinaci, W. M. P. van der Aalst, J. Domingue, M. Song, A. Rozinat, B. Norton, and L. Cabral. An outlook on semantic business process mining and monitoring. In *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops*, volume 4806 of *LNCIS*, pages 1244–1255, Vilamoura, Portugal, November 2007.
- [AFKK07] W. Abramowicz, A. Filipowska, M. Kaczmarek, and T. Kaczmarek. Semantically enhanced Business Process Modelling Notation. In *Workshop on Semantic Business Process and Product Lifecycle Management (SBPM)*, volume 251 of *CEUR Workshop Proceedings*, pages 88–91, Innsbruck, Austria, June 2007.
- [CAAdMZ⁺07] I. Celino, A. K. Alves de Medeiros, G. Zeissler, M. Oppitz, F. Facca, and S. Zöller. Semantic Business Process Analysis. In *Workshop on Semantic Business Process and Product Lifecycle Management (SBPM)*, volume 251 of *CEUR Workshop Proceedings*, pages 44–47, Innsbruck, Austria, June 2007.
- [DSSK07] M. Dimitrov, A. Simov, S. Stein, and M. Konstantinov. A BPMO Based Semantic Business Process Modelling Environment. In M. Hepp, K. Hinkelmann, D. Karagiannis, R. Klein, and N. Stojanovic, editors, *Workshop on Semantic Business Process and Product Lifecycle Management (SBPM)*, volume 251 of *CEUR Workshop Proceedings*, pages 101–104, Innsbruck, Austria, June 2007.
- [EKSMP08] M. El Kharbili, S. Stein, I. Markovic, and E. Pulvermüller. Towards a Framework for Semantic Business Process Compliance Management. In *The Impact of Governance, Risk, and Compliance on Information Systems (GRCIS)*, volume 339 of *CEUR Workshop Proceedings*, pages 1–15, Montpellier, France, June 2008.

- [FLP⁺06] D. Fensel, H. Lausen, A. Polleres, J. de Bruijn, M. Stollberg, D. Roman, and J. Domingue. *Enabling Semantic Web Services: The Web Service Modeling Ontology*. Springer, 2006.
- [FRS07] J. Frankowski, P. Rubach, and E. Szczekocka. Collaborative Ontology Development in Real Telecom Environment. In *1st International Working Conference on Business Process and Services Computing (BPSC)*, volume 116 of *LNI*, pages 40–53, Leipzig, Germany, September 2007.
- [HLD⁺05] M. Hepp, F. Leymann, J. Domingue, A. Wahler, and D. Fensel. Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management. In *IEEE International Conference on e-Business Engineering (ICEBE)*, pages 535–540, Beijing, China, 2005.
- [HR07a] M. Hepp and D. Roman. An Ontology Framework for Semantic Business Process Management. In *8th International Conference Wirtschaftsinformatik*, pages 423–440, Karlsruhe, 2007.
- [HR07b] M. Höst and P. Runeson. Checklists for Software Engineering Case Study Research. In *1st International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 479–481, Madrid, Spain, September 2007. IEEE CS.
- [KPP95] B. Kitchenham, L. Pickard, and S. L. Pfleeger. Case Studies for Method and Tool Evaluation. *IEEE Software*, 12(4):52–62, 1995.
- [NWvL07] J. Nitzsche, D. Wutke, and T. van Lessen. An Ontology for Executable Business Processes. In *Workshop on Semantic Business Process and Product Lifecycle Management (SBPM)*, volume 251 of *CEUR Workshop Proceedings*, pages 52–63, Innsbruck, Austria, June 2007.
- [SKD⁺08] S. Stein, S. Kühne, J. Drawehn, S. Feja, and W. Rotzoll. Evaluation of OrViA Framework for Model-Driven SOA Implementations: An Industrial Case Study. In *6th International Conference on Business Process Management (BPM)*, volume 5240 of *LNCS*, pages 310–325, Milan, Italy, September 2008. Springer.
- [SS08] H. J. Schmelzer and W. Sesselmann. *Geschäftsprozessmanagement in der Praxis*. Carl Hanser Verlag, München, Germany, 6th revised edition, 2008.
- [SSEK08a] C. Stamber, S. Stein, and M. El Kharbili. Prototypical Implementation of a Pragmatic Approach to Semantic Web Service Discovery During Process Execution. In W. Abramowicz and D. Fensel, editors, *11th International Conference on Business Information Systems (BIS)*, volume 7 of *LNBIP*, pages 201–212, Innsbruck, Austria, May 2008. Springer.
- [SSEK08b] S. Stein, C. Stamber, and M. El Kharbili. ARIS for Semantic Business Process Management. In *Workshop on Advances in Semantics for Web Services (semantics4ws)*, Milan, Italy, September 2008.
- [Yin03] R. K. Yin. *Case Study Research: Design and Methods*, volume 5 of *Applied Social Research Methods Series*. Sage Publications, London, UK, 3rd edition, 2003.

Policy-Based Semantic Compliance Checking for Business Process Management

Marwane El Kharbili, Sebastian Stein
IDS Scheer AG, ARIS Research, Altenkesseler Str.
17, D-66115 Saarbrücken, Germany.
{marwane.elkharbili, sebastian.stein}@ids-scheer.com
Elke Pulvermüller
Institute of Computer Science, University of Osnabrück.
Albrechtstr. 28, 49076 Osnabrück, Germany.
elke.pulvermueller@informatik.uni-osnabrueck.de

Abstract: Compliance management, risk analysis, and auditing are disciplines that are critical for large scale distributed enterprise systems. The way these complex systems are developed and deployed makes the management and enforcement of enterprise goals or policies a hard task. This is also true for compliance management of business processes (BPs). Such an observation is emphasized if we give compliance management the scope of the whole enterprise model. In this paper we explain our approach to modeling compliance measures based on policies and present a framework for managing and enforcing compliance policies on enterprise models and BPs. We discuss our ideas in the context of a semantically-enabled environment and discuss why leveraging compliance checking to a semantic level enhances compliance management.

1 Introduction

In past years, an intense public discussion took place dealing with financial scandals happening at major companies and corporations like Enron, WorldCom, Roche, Siemens, and Volkswagen. Based on those events, the importance of compliance management as a critical responsibility at the highest management levels to prevent such scandals has drastically increased. For instance, in 2002 the US government created the “Public Company Accounting Reform and Investor Protection Act” [otUS02], also known as the Sarbanes-Oxley Act, to define mandatory policies for public companies and public accounting companies. Complying with regulations of all sorts is usually needed for purposes ranging from ensuring that specific norms are met (e.g. quality standards such as ISO9000:2005 [fs05]) to proving correct implementation of internal controls imposed by active legislations (e.g. SOX Sec.404 [otUS02]) [KSMP08].

Examples of regulations are the HIPAA¹ (Health sector), FDA regulations² (food/drug sector), BASEL-II³ (Banking sector), ISO27002:2008⁴ (IT security) and KonTraG⁵ (cor-

¹Health Insurance Portability and Accountability Act: <http://www.hipaa.org/>.

²US Food and Drug Administration: <http://www.fda.gov/opacom/laws/>.

³Basel II Revised International Capital Framework.

⁴The ISO 27002:2005 IT security standard: <http://www.iso.org>.

⁵German law for Control and Transparency in the private sector.

porate governance). A given company is likely to be under jurisdiction of several regulations concurrently [KMS07].

The following sections of this paper give a short definition of compliance management and then discuss the problems related in order to grasp the challenges ahead. Section 3 discusses the idea of model-driven compliance checking using policies is and makes a realization proposal. Section 4 introduces a framework for integrated policy-based compliance checking as well as the accompanying ontological framework. Finally, related and future work are outlined before we conclude our contribution.

2 Compliance Management: a definition

Compliance management is a broad term covering all activities and methods to ensure that a company follows all policies required by an external or internal regulation. These regulations are usually described in a natural language document (e.g. as is the case for laws), which can be hardly understood by non-experts of the field the regulation acts on. In the prominent example of the Sarbanes-Oxley Act, if a company follows all guidance defined in such a regulation document, the company is said to be in compliance with the given regulation. Otherwise, the company is said to be violating this regulation.

2.1 Regulatory Compliance

These regulations can be structured and documented in compliance frameworks and complying with the framework is thus regarded as equivalent to complying with the regulation. If no such frameworks exist, then companies have neither guidance nor support for implementing regulations, apart from that of auditors. The examination performed to validate whether a company actually implements a given compliance framework is called audit and the person or organization doing such an audit is called auditor.

Besides legal requirements, in order to use a certain compliance framework, companies often decide to do this for reasons ranging from certification, risk assessment, to the implementation of quality standard implementation, etc. The latter are strategic reasons and do not result from legal pressure exercised by governmental bodies. As an example, practically all companies that reach a certain size decide to endorse quality standards like ISO 9000:2000 [fS05] to publicly demonstrate the company's quality commitment and customer focus.

2.2 Compliance Audits

In order to be successfully audited, a company must in advance ensure that it follows all guidelines defined by the compliance framework. A possible approach for this is to check the degree to which these guidelines are fulfilled by the company's enterprise model. Theoretically, this would have to be done by identifying all relevant aspects of the company's activities on which parts of the compliance Framework apply, and checking this compliance. This task is one of experts who need to either have deep knowledge of the activities, processes, architectures and other enterprise model artifacts of the company (e.g. internal compliance controls), or who are provided tight cooperation with enterprise insiders who

dispose of the necessary knowledge about the activities of the company.

Today, compliance audits are manual and error-prone tasks requiring significant effort. To make the auditing of enterprise models an easier and more efficient task, *automation and full-coverage are key goals*. Moreover, in order to increase quality (in terms of accuracy and credibility of checking reports) and reduce the cost (in terms of human capacities and time) of compliance checking, the idea of using semantic technologies has been proposed [EKSMPO8].

Semantic compliance checking relies on a semantically defined compliance framework and uses semantic technologies such as inference engines to evaluate the compliance of a given semantic enterprise model. Our research focuses on *the use of these technologies for designing a policy-based framework for regulatory enterprise compliance management*.

3 Enterprise Regulatory Compliance: The Problem

Compliance frameworks consist of a set of guidance elements and measures that have to be taken in order to follow the latter. These compliance measures are often represented as guidelines, policies or controls, depending on the level of abstraction from the concrete implementation of compliance measures (See figure 1). For instance, the authors of [F.Y07] use the example of an ISO 17799:2002 (IT security standard) access control requirement being further refined into a set of policies. The adherence of a company to these policies has to be evaluated by humans.

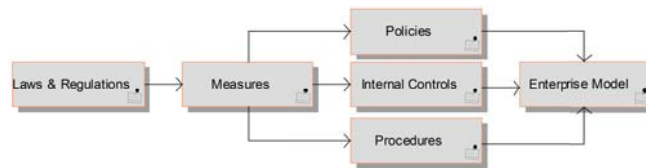


Figure 1: High-Level Compliance Model

3.1 Example: Segregation of Duty (SoD)

The Sarbanes-Oxley Act defines the rule that a financial auditor of a company is not allowed to also be involved in the bookkeeping or accounting of the audited company ([otUS02], Sec. 201 (g) (1)). Such a rule is part of a more complex set of rules specifying which concrete roles/responsibilities can be concurrently carried out by an individual. This set of rule is regrouped under the term SSegregation of Duty (SoD)ppolicy. In concrete cases, SoD policies are two-dimensional matrixes of available roles where SoD violations can be visualized in the cells where two roles cross each other. This is shown in the following simple example in figure 2 by specifying which tasks can be concurrently realized by the same individual.

In this example, the auditor must examine who did the bookkeeping and the financial audit and that both roles are not shared by one person or organization. Given the fact that a compliance framework usually consists of many policies, it is a significant effort for an

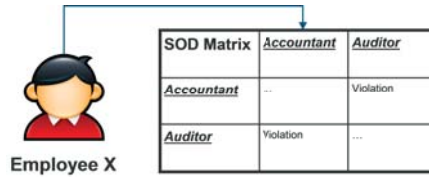


Figure 2: Segregation of Duty (SOD) policy example

auditor to check all of them and for all concerned employees. Therefore, an auditor picks (based on experience or randomly) a set of business artifacts upon which policies have to be checked. This helps to reduce the auditing effort, but this strategy cannot be applied by a company to ensure compliance before the actual audit takes place, neither does it guarantee the benefits of being a highly compliant company with target regulations (i.e. this is particularly relevant in case of quality standards audit).

The consequences of a failed audit can be significant. For example, if a public company finds itself forced to delay its annual balance because of a failed financial audit, it might lose significant market capitalization (e.g. falling stock price resulting from loss of confidence in the market). Consequently, complying with all policies defined by a specific compliance framework becomes a precondition for realistic preparation to an audit. One of the challenges identified here is: how to structure compliance and how to model it?

3.2 Enterprise Compliance Management and Enterprise Models

Compliance is all about control, and it is hard to control what is not thoroughly known. Companies create enterprise models to represent their structure and dynamics. Various guidelines to structure such a model exist like the Zachman [JAZ92], the TOGAF [Gro] or ARIS [Sch00] frameworks. An enterprise model is used to document the as-is reality of a company as well as a planning tool for to-be scenarios. Another dimension in enterprise models is capture full semantics of the latter and thus allowing for machine processability of the information available about these models. There are various research efforts made to formalize the underlying meta-models of enterprise models using Ontologies [HLD⁺05, UKMZ98]. Enterprise models span the whole vertical structure of a company and contain among others BP models and internal governance policies.

On the other hand, regulations, which guide the development of compliance frameworks, are usually available as text documents, often requiring juristic skills to interpret them correctly. Not surprisingly, checking a given enterprise model for compliance is therefore a manual task carried out by certified domain experts. Furthermore, efficient compliance management requires good knowledge of the enterprise model of a company by the auditors, and a good knowledge of the compliance framework by the management. This makes close collaboration of auditors with management boards a necessity. The challenge of automated compliance checking can be seen as *extending an enterprise model* to include aspects defined by the compliance framework and *enabling the needed automation in checking an enterprise model against the policies* defined in a compliance framework. We argue that *measures defined in order to ensure regulatory compliance can be repre-*

sented using policies (See figure 1). Thus, enforcing these policies guarantees a state of compliance. Putting policies to use for this purpose allows profiting from policy management formalisms and frameworks. We also argue that making use of semantic technologies for representing compliance policies is necessary to deal with semantically lifted process models. It also *helps making compliance checking more precise by allowing to model compliance policies at higher abstraction levels in order to cope with the ambiguity inherent to regulations*.

4 Semantic Policy-Based Compliance Checking

4.1 Formal, Declarative, Semantic & Domain-Dependent Policies

Policies have to be structured and expressed using formal means making their automatic processing possible. The authors of [LGRM⁺08] see modeling compliance constraints in declarative fashion, while respecting a trade-off between expressiveness of the formal language used and the cost of inference and analysis. Declarative languages are supported as the preferred approach to modeling compliance as a number of works show [LGRM⁺08, IWH, ZM06, GG06]. This has the advantage of separating between compliance models and targeted enterprise models. In comparison to approaches such as the one presented in [DF06], it also has the advantage of better scaling with regulation change and complexity of the targeted enterprise models.

Policies can also be expressed using formalisms such as rules. Rules are one classical and very intuitive way of expressing/implementing policies. In [OMG], the SBVR⁶ standard is defined for expressing rules and vocabularies on a business level. While providing a natural-language-like syntax that is very easy to use for business users, an underlying formalization of used rule constructs is provided. However, rule interpretation and execution is not enabled because of a missing mapping to an executable rules language. The PRR⁷ [OMG07] standard could fill this gap as it is designed to transport production rule logic and a mapping to SBVR would create the link for rules from the business level to a rule engine (execution) level.

Enterprise models describe architectures, processes and architectures at different degrees of detail and under various perspectives. They usually deal with high heterogeneity on both business and technical levels [JAZ92, Sch00] (strategic, tactical, business, operational and technical levels as distinguished in [vl01]). Ontologies allow *Achieving interoperability between multiple representations of reality[...]and between such representations and reality, namely human users and their perception of reality*. [Hep07]. As compliance is a vertical concern, a compliance framework needs to handle the different perspectives on the various layers. These layers and perspectives can be integrated on a semantic level giving meaning to the relationships between them. The work realized in the SUPER⁸ Research project seek to build a stack of ontologies for BPM doing just this. In [Jab96, Cur92], the functional, behavioral, organizational, and informational perspectives are considered for the BP ontology. In [IM07], a formal model is proposed for describing BPs taking the previous four dimensions into account. A compliance framework should then support

⁶Semantics of Business Vocabulary and Rules

⁷Production Rule representation

⁸Semantics Used for Process management within and between Enterprises. www.ip-super.org.

handling this integrated these ontologies and support automated checking/enforcement of policies on instances of these ontologies.

4.2 An Approach to Compliance Management

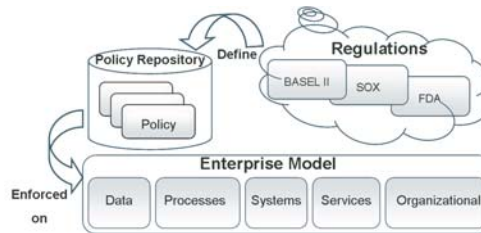


Figure 3: An approach for compliance formalization using policies

Our proposal for handling the aspects we just cited is *semantic compliance checking (SCC)*. SCC extends enterprise models semantically to integrate semantically described policies with semantic enterprise models. The initial Input are, as is the case for auditors, the actual regulations, laws or norms that define the policies to be compliant with. These can be structured and represented using dedicated domain policy ontologies (e.g. IT security, food regulation etc.). These policies are then made available in a policy repository for integration into enterprise models. A compliance engine specifically implemented on the basis of an inference engine for the policy ontology language embodies the necessary compliance checking algorithms. This approach is shown in figure 3. Existing works follow a similar idea, as in [SN07, NS07b], although concentrating on risk management approaches to compliance checking. Other works already started formalizing regulations such as the Sarbanes-Oxley-Act [KMS07] and BASEL II [RF06]. Due to the organization or regulations in separate domains of enactability, it is possible to separate policy ontologies per domain, making an additional level or *super-policies* ontology necessary in order to link domain policy ontologies together. This allows e.g. combining policies and enacting them concurrently on the same enterprise model.

5 An Architecture for Compliance Management

5.1 Enterprise Models: A Layered View

An enterprise's structure can be seen as the set of layers distinguished in 4. On the top layers are business goals which are fulfilled by defining corporate strategies. These strategies are supported by policies and governance guidelines. Policies constrain and control business artifacts: BPs, business rules, business data (vocabularies), etc. On an inferior layer are operational artifacts such as operational rules and executable processes. The lowest layer in this view contains the applications, components, systems and deployment environments that host the concrete carriage of IT activities.

This view of the enterprise can be mapped to a paradigm that we call Decision-Action-Information (DAI) as shown in 4. In this paradigm, the enterprise is seen as composed of



Figure 4: A layered view on the enterprise for compliance

three basic classes of artifacts. The decision class contains all artifacts supporting decision management such as business rules. The action class contains logical and operational artifacts that actually carry out business activities. Finally, the Information class contains all data artifacts such as execution logs or database tables, on the basis of which decision are partly taken and which are needed by action class artifacts in order to realize their functionality. Our work hypothesis is that compliance is defined for action artifacts and needs to be modelled as decision class artifacts.

5.2 A Framework for Compliance Management

In the following, we focus on designing a framework for compliance management shown in figure 5. This framework does not yet define a detailed technical architecture, it rather defines requirements. We have distinguished five axes on which efforts will concentrate: (i) architecture, (ii) compliance management process, (iii) ontologies, (iv) compliance checking algorithms, and (v) policy management lifecycle. The following points have been retained:

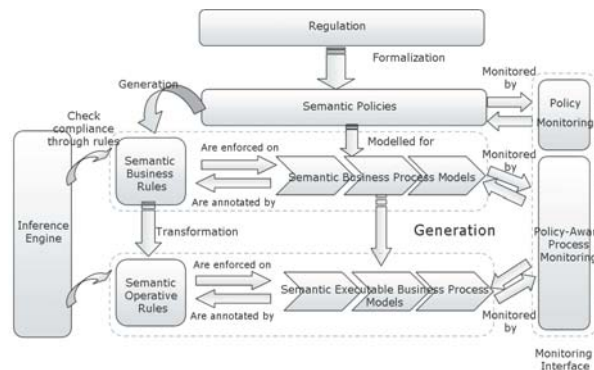


Figure 5: An Architecture for a compliance checking framework

- regulations need to be formalized in order to be machine-processable. We have to

provide mechanisms to structure and then formalize regulations as semantic policies.

- semantic policies have to be modelled into BPs. In the case of semantic business process management (SBPM), this means extending the ontology for modeling BPs with an ontology for modeling policies.
- Rules are an intuitive way of implementing policies. Policies have to be transformed into sets of semantic business rules. It implies defining a business rules ontology and selecting an ontology language supporting expressing rules. No assumptions about the expressiveness and the kind of logic supported by this rule ontology language have been made yet. These business rules can then be integrated into process modeling frameworks and interpreted by an adapted inference engine.
- BPs are represented in languages adapted to BP execution. On this level, it is necessary to further transform business rules into operative rules that can be integrated into semantic executable BP models.
- A compliance checking engine has to be implemented by building on an inference engine. This compliance checking engine implements generic compliance checking algorithms.
- Monitoring components are needed to control the consistency of policies, but also to monitor the checking and enforcement operations on BPs.

Three main layers have been identified and need to be regarded separately. The policy layer is the management layer where policies are expressed, and functionalities such as conflict resolution, speech acts, delegation, policy priorities, meta-policies and the definition of jurisdictions are available. Policy consistency checks also take place on this layer. The second layer contains design-time artifacts such as BP models and business rules. Just as BP models need to be transformed into executable process models that can be run on several execution engines, business rules need to be transformed into operative rules which can be run on the same layer as BP execution engines. The prefix *semantic* means that policies, BP models, executable BP models, business rules and operational rules are all defined using dedicated ontologies.

Additionally to these layers, there are two vertical components: the monitoring components and the inference engine. Monitoring is needed both for the operations taken on policy models and for monitoring design-time and run-time decisions taken by policies which closes the lifecycle for one compliance management iteration.. This requirement has also been identified in [KD06]. The inference engine operates on both business rules and operational rules (which implement the decision-making logic behind policies) in order to check for regulation policy violation. The policy layer disposes of its own engine for inferring on policy management aspects such as inconsistency or conflict detection.

This framework requires the definition of a set of transformations. Figure 6 shows how policy layers map to semantic layers. The semantic policy layer contains ontologies for definition and management of policies as well as domain policy layers. A first transformation is needed in order to generate business rule models out of the policy definitions. A

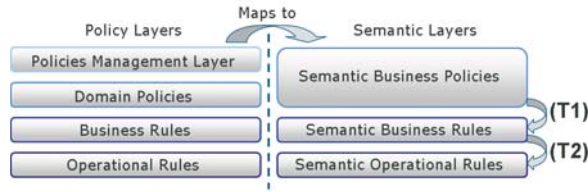


Figure 6: Policy layers and transformations in the framework

second transformation is needed in order to generate operational rule models out of business rules models. Our goal is to define languages (in the form of ontologies) for each of these layers and to complete these with generic transformations between the ontologies. We are currently concentrating on the definition of an ontology for policies and rules. Future work will include the definition of mappings to the SBVR [OMG] and PRR [OMG07] standards.



Figure 7: Managing policies

A lifecycle needs to be defined for the management of semantic policies. Figure 7 makes use of the components identified in the architecture for this lifecycle. After being defined (as a policy ontology instance), a policy must be verified for formal consistency and conflicts with other ontologies. The next steps will then be to generate design-level and execution-level models of these policies and to enforce them on BPs. Furthermore, analyzing the execution of policies and the decisions takes by the latter completes the lifecycle and provides insights into how tight do the designed policies match the initial regulations. The analysis phase outputs also serve as audit artifacts that can show that the right policies have been defined and that these are working correctly.

6 An Ontology for Policies and Rules

The business policy and Rule ontology (BPRO) we introduce has been modelled to fulfill the requirements we already identified. In the following, we will shortly introduce its main concepts and relationships. Because of space restrictions, the ontology components won't be introduced in detail (i.e. information to cardinalities and examples are not included).

Figure 8 shows the core policy ontology. The central concept is the policy concept. A policy is a meta-policy if it is enacted on other policies. A policy belongs to a strategy and is part of the implementation of one or many regulations. A policy fulfills a business goal. A constraint is one kind of policy, next to Decision, functional and core policies. A constraint policy decides on how to constrain a resource in showing some behavior. It delivers one or many of several discrete allowed business artifact behaviors/states and does not provide a binary yes/no answer as a decision policy does. A core policy is a policy which takes no decision that has to be enforced on business artifacts, it can only be invoked by

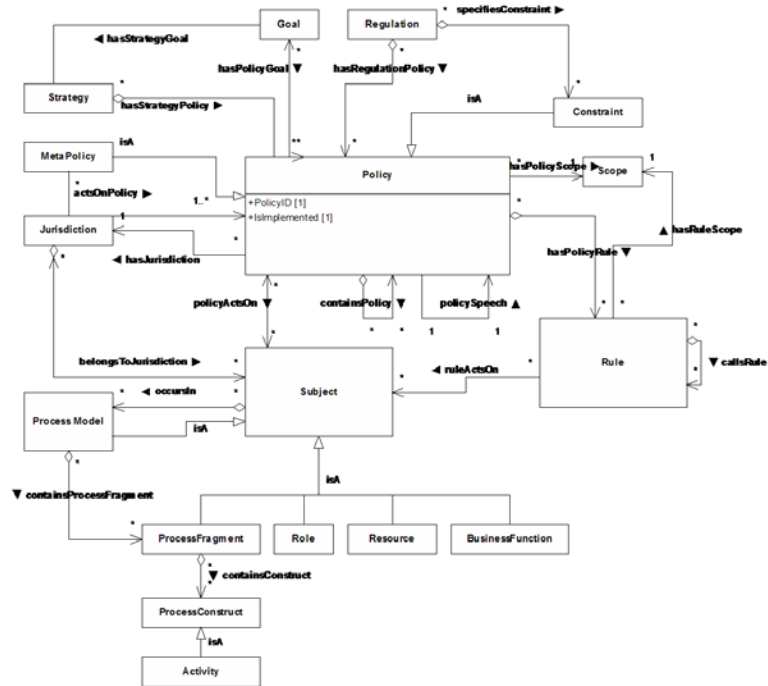


Figure 8: Policy Ontology

other policies and delivers intermediary decisions. A functional policy applies for business functions that are able to execute differently depending on the parameters given to them. A functional policy decides on which concrete action these business function as can take by setting these parameters.

A policy has a subject, which is the entity(ies) on which it can apply. This subject can be a process model for example or any business artifact part of the enterprise model (e.g. role, resource, business function etc.). A process is composed of process fragments and the latter are composed of process constructs such as activities. The concepts related to BP modeling have to be mapped to the used BPM ontologies.

A policy has a jurisdiction and a scope. A jurisdiction is the domain in which a policy has the right to take decisions. Outside its jurisdiction, a policy cannot take any decisions, cannot be solicited, and cannot communicate with other policies about subjects not belonging to its jurisdiction. A Jurisdiction is a set of subjects. These sets of subjects can be defined in a declarative way, such as using assertions on properties of subjects: **all roles of type==[engineer | manager] where role.budget>= 1000 units**. We do not take into account jurisdiction management (which would require a dedicated algebra) in order to define these inter-policy relations unambiguously. Scopes are different from a jurisdiction in that scopes are always strictly included in jurisdictions and define the set of subjects inside a given jurisdiction upon which a policy can take a decision. Scopes introduce additional flexibility in managing policies, by allowing to mmoveä policy’s scope inside a

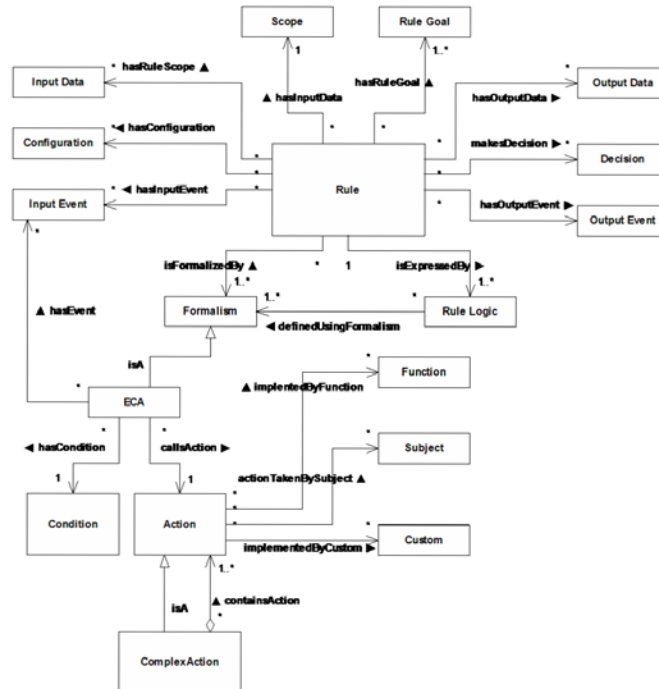


Figure 9: Rule Ontology

given jurisdiction.

A rule belongs to one or many policies. That means rules can be composed in order to implement a certain policy. A rule is also attached to a business goal and has a scope. A rule has input and output data it processes and an input and output event. Input events can trigger the execution of a rule and output events are generated by a rule to trigger other rules or actions. A rule has configuration data which makes a rule able to execute different logic depending on its configuration. A rule makes a decision.

A rule contains rule logic which is expressed in a certain formalism. Formalisms are many and in the figure above, the Event-Condition-Action (ECA) formalism has been used as an example. An ECA rule has a condition and an action which can be a complex expression of actions. It is also triggered by one of the input events of the rule. The action taken by the ECA rule is done on a subject.

7 Related Work

There has been ongoing work on semantic compliance management, as shown in [NS07a], where an approach for semantic compliance management for BPM is presented. However, the approach used concentrates on implementing internal controls. Such an approach is adapted to compliance management but is restrictive because it relies on the necessary definition of risks. Another approach is presented in [SN07] where the authors introduce the

modeling of internal control objectives in BPs as a mean to integrate compliance requirements in BP design. The authors also relate their work to risk analysis and internal control modeling. Policies are meant to be generic and do not depend on a previous definition of risks in processes.

In our approach, policies are meant to be directly extracted from regulations, either in automated fashion, by relying on natural language processing techniques, or semi-automated fashion, by generating policy templates out of regulatory documents for the policy expert to complete. This introduces a layer between the modeling of regulatory compliance requirements and actual regulatory compliance enforcement. Such a layer would allow for example to exchange policies or discover policy conflicts between BPs existing in different departments or organizations. Moreover, policies can themselves be used to implement internal controls. Policies also allow for profiting from inference mechanisms in order to take decisions through the use of specifically designed policy inference engines such as in [Kag04].

In [Hua05], a framework is introduced for semantic security management in BPs. However, the presented approach focuses only on security concerns and does not seek to define its own ontologies. It relies on previous work ([Kag04, ea04]). In [KMS07] and [Kar], another approach for BP compliance management is presented. It defines an extension for a BP meta-model for regulatory compliance. However, the approach does not incorporate ontologies and thus, does not profit from the power of semantic technologies. In [GV06] and [GG06], deontic (obligations and permissions) constraints expressible for BPs are modelled using temporal deontic assignments. The latter can also be used in BP design and in expressing BP contracts.

8 Conclusion and Future Work

In this paper we have thoroughly introduced the business problem of compliance checking and motivated the need for a comprehensive compliance management framework. We proposed and justified the use of policies for this purpose, which decision is also supported by existing works. Enterprise models are defined semantically and enriched with compliance measures modelled as elements of a policy ontology. This requires an integration of enterprise models and compliance management models. Specifically implemented inference engines can be used to reason over the resulting models and decide on or enforce compliance. An architecture has been presented in order to illustrate our approach. The different layers, components and interactions between these components as well as necessary model transformations were introduced. The first steps towards implementing this framework have been taken and an ontology proposed. While we go further in realizing a reference implementation of the proposed architecture, new requirements will appear and can push us to slightly modify the architecture. As a next step, we will design tools to allow editing and building compliance policy ontologies. We also will define and implement the necessary ontology transformations discussed above. As a proof-of-concept, we will seek to define realistic use cases for a specific domain (e.g. quality management) and showcase the use of the compliance framework. Ultimately, the goal of this work is to showcase how using semantics, policy management and rule management can make

compliance checking automatable⁹.

References

- [Cur92] Kellner M.I. Over J. Curtis, B. Process Modeling. Comm. of the ACM, September 1992. 35(9):75.
- [DF06] Wilhelm Rossak Daniel Foetsch, Elke Pulvermueller. Modeling and Verifying Workflow-based Regulations. In *Proceedings of the international workshop on regulations modeling and their validation and verification. REMO2V06.*, pages 825–830. CEUR-WS.org/vol-241, Luxemburg, June 2006.
- [ea04] Tabet S. et al. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 21, May 2004 2004.
- [EKSMP08] Marwane El Kharbili, Sebastian Stein, Ivan Markovic, and Elke Pulvermüller. Towards Policy-Powered Semantic Enterprise Compliance Management – Discussion Paper. In *3rd International Workshop on Semantic Business Process Management (SBPM)*, CEUR Workshop Proceedings, Tenerife, Spain, June 2 2008.
- [fS05] ISO International Organization for Standardization. ISO9000:2005 - Quality management systems, Fundamentals and vocabulary., 20.09.2005 2005.
- [F.Y07] N. Parameswaran & P. Ray F.Yip. Rules and Ontology in Compliance Management. In *Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference*, number 1541-7719, page 435, Washington, DC, USA, 2007. Washington, DC, USA, IEEE Computer Society.
- [GG06] Shazia Sadiq Guido Governatori, Zoran Milosevic. Compliance checking between business processes and business contracts., In *Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06)*, pages pp. 221–232, 2006.
- [Gro] The Open Group. The Open Group Architectural Framework (TOGAF).
- [GV06] Stijn Goedertier and Jan Vanthienen. *Designing Compliant Business Processes with Obligations and Permissions*, volume 4103 of *LNCS*, chapter BPM 2006 Workshops, pages 5–14. Springer Verlag, 2006.
- [Hep07] Martin Hepp. Ontologies: State of the art, business potential, and grand challenges. In Martin Hepp, Pieter De Leenheer, Aldo de Moor, and York Sure, editors, *Ontology Management: Semantic Web, Semantic Web Services, and Business Application*, pages 3–22. Springer, 2007.
- [HLD⁺05] M. Hepp, F. Leymann, J. Domingue, A. Wahler, and D. Fensel. Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management. In Francis C. M. Lau, Hui Lei, Xiaofeng Meng, and Min Wang, editors, *ICEBE*, pages 535–540. IEEE Computer Society, 2005.
- [Hua05] Dong Huang. Semantic policy-based security framework for business processes. Proceedings of the Semantic Web and Policy Workshop - csee.umbc.edu, November 2005. 4th International Semantic Web Conference, 7 November 2005, Galway, Ireland.

⁹Acknowledgements: We thank the EU commission for supporting our research within the SUPER project (www.ip-super.org).

- [IM07] Alessandro Costa Pereira Ivan Markovic. Towards a Formal Framework for Reuse in Business Process Modeling. Workshop on Advances in Semantics for Web services (semantics4ws), in conjunction with BPM '07., September 2007 2007. Brisbane, Australia, September 2007.
- [IWH] Guido Governatori Ingo Weber and Joerg Hoffmann. Compliance Checking for Process Repositories. In Dr Michael zur Muehlen Dr Shazia Sadiq, Dr Marta Indulska, editor, *Proceedings of the Workshop on the Impact of Governance, Risk, and Compliance on Information Systems (GRCIS)*. Montpellier, France, June 2008.
- [Jab96] Bussler C Jablonski, S. *Workflow Management: Modeling Concepts. Architecture, and Implementation*. International Thomson Computer Press., London, UK., 1996.
- [JAZ92] J. F. Sowa J. A. Zachman. Extending and Formalizing the Framework for Information Systems Architecture. *IBM Systems Journal*, Volume 31, No. 3., 1992.
- [Kag04] Lalana Kagal. *A Policy-Based Approach to Governing Autonomous Behavior in Distributed Environments*. Phd thesis, Faculty of the Graduate School of the University of Maryland, 2004.
- [Kar] D. Karagiannis. A Business process Based Modelling Extension for Regulatory Compliance. In Multikonferenz Wirtschaftsinformatik 2008, Munich, 2008.
- [KD06] M. Schwab M.: In Karagiannis D., Nemetz. Dashboards for Monitoring Compliance to Regulations - A SOX-based Scenario. In *Proceedings of IGO'06 - International Conference on Integrating Global Organizations*. Siena, 2006., 2006.
- [KMS07] Dimitris Karagiannis, John Mylopoulos, and Margit Schwab. Business Process-Based Regulation Compliance: The Case of the Sarbanes-Oxley Act. In *Requirements Engineering Conference, 2007. RE '07. 15th IEEE International*, pages 315–321, 2007.
- [KSMP08] M. El Kharbili, S. Stein, I. Markovic, and E. Pulvermüller. Towards a Framework for Semantic Business Process Compliance Management. In S. Sadiq, M. Indulska, and M. zur Muehlen, editors, *GRCIS Workshop - CAISE Conference*, 2008.
- [LGRM⁺08] Linh Thao Ly, Kevin Göser, Stefanie Rinderle-Ma, , and Peter Dadam. Compliance of Semantic Constraints A Requirements Analysis for Process Management Systems. In Sadiq S., Indulska M., and zur Muehlen M., editors, *Proceedings of the GRCIS08: International Workshop on Governance, Risk and Compliance - Applications in Information Systems.*, June 2008.
- [NS07a] Kioumars Namiri and Nenad Stojanovic. A Formal Approach for Internal Controls Compliance in Business Processes. In *8th Workshop on Business Process Modeling, Development, and Support (BPMD07)*, page 9, Trondheim, Norway, 2007. BPMD07.
- [NS07b] Kioumars Namiri and Nenad Stojanovic. *Using Control Patterns in Business Processes Compliance*, volume 4832/2007, pages 178–190. 2007.
- [OMG] OMG. OMG Business Modeling Specifications - Semantics of Business Vocabulary and Rules.
- [OMG07] OMG. Production Rule Representation (PRR) - Beta - OMG adopted specification. <http://www.omg.org/docs/dtc/07-11-04.pdf>, November 2007.
- [otUS02] Congress of the United States. Public Company Accounting Reform and Investor Protection Act (Sarbanes-Oxley Act). Pub. L. No. 107-204, 116 Stat. 745, 2002.

- [RF06] Andre Rifaut and Christophe Feltus. Improving Operational Risk Management Systems by formalizing the BASEL II Regulation with Goal Models and the ISO/IEC 15504 Approach. In *Proceedings of the REMO2V06*, page 831, 2006.
- [Sch00] August-Wilhelm Scheer. *ARIS - Business Process Frameworks*. Springer, 3rd ed. edition, April 2000.
- [SN07] Governatori G. Sadiq, S. and K. Namiri. *Modeling Control Objectives for Business Process Compliance*, pages 149–164. Lecture Notes in Computer Science. Springer, 2007.
- [UKMZ98] Mike Uschold, Martin King, Stuart Moralee, and Yannis Zorgios. The Enterprise Ontology. *The Knowledge Engineering Review*, 13, 1998. <http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html>.
- [vl01] A. van lamsweerde. Goal-Oriented Requirements Engineering: A guided tour. In *Invited minitutorial, proceedings of the RE01*, pages 249–263. International Joint Conference on Requirements Engineering, Toronto IEEE., August 2001.
- [ZM06] M. Orłowska Z. Milosevic, S. Sadiq. Translating business contract into compliant business processes. In *Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06)*, pages pp. 211–220, 2006.