

Transformation und Mapping von Prozessmodellen in verteilten Umgebungen mit der ereignisgesteuerten Prozesskette

Timo Kahl, Florian Kupsch

Institut für Wirtschaftsinformatik (IWi)
im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI)
Stuhlsatzenhausweg 3, Geb. D3 2
D-66123 Saarbrücken
{Kahl|Kupsch}@iwi.uni-sb.de

Abstract: Zwingend notwendig für einen hohen Grad an Interoperabilität zwischen Unternehmen, bei denen Unternehmensmodelle die Grundlage des täglichen Geschäfts darstellen, ist das gegenseitige Verständnis dieser Modelle. Aus diesem Grunde ist zur Optimierung einer Kollaboration ein Modellaustausch zwischen zwei oder mehreren Partnern eine notwendige Voraussetzung. Da in einem kollaborativen Umfeld zumeist mehrere Sprachen und Notationen Anwendung finden, bedingt der Austausch von Modellen zumeist eine Umwandlung von einer Notation oder Sprache in eine andere. Dies geschieht bis dato zumeist manuell. In dieser Ausarbeitung werden deshalb zwei Konzepte diskutiert, das Mapping und die Transformation, die eine teil- und vollautomatisierte Modellumwandlung von einem Format in ein anderes ermöglichen. Beim Mapping geht es um den zuvor beschriebenen Modellaustausch wohingegen die Transformation auf eine Formalisierung der Modelle bis hin zu einer möglichen Ausführung abzielt. Die praktische Umsetzung beider Konzepte wird anhand von zwei Prototypen aufgezeigt.

1. Einleitung

Veränderte Rahmenbedingungen haben Unternehmen vieler Branchen vor eine Vielzahl neuer Herausforderungen gestellt. Diese resultieren vor allem aus der zunehmenden Marktdynamik, dem wachsenden Einfluss der Kunden auf die Gestaltung von Produkten aber auch aus kürzeren Produktlebenszyklen [Sc04]. Eine Reaktion vieler Unternehmen auf diese Entwicklungen ist die Konzentration auf spezifische Kernkompetenzen und eine Auslagerung der nicht zum Kernportfolio gehörenden Unternehmensaktivitäten an Dritte. Dies bedingt jedoch auch eine engere Zusammenarbeit mit anderen Unternehmen und eine Zunahme der Komplexität dieser Zusammenarbeit. Die Gestaltung und das Verhalten kooperativer Organisationsformen wird seit etwa einem halben Jahrzehnt verstärkt mit dem Begriff Collaborative Business (C-Business) bezeichnet [Rö01]. Dabei geht es um eine optimale Gestaltung und Ausführung der unternehmensübergreifenden Zusammenarbeit über die gesamte Wertschöpfungskette hinweg [Ke03]. Eine Voraussetzung für eine

erfolgreiche C-Business-Strategie und deren Umsetzung ist die Interoperabilität einer Organisationsform. Interoperabilität bezeichnet die Fähigkeit eines Unternehmens mit anderen Unternehmen oder Kunden unter der Prämisse eines optimalen Nutzen-Aufwand-Verhältnis zusammenzuarbeiten [INT05]. Zwingend notwendig für einen hohen Grad an Interoperabilität zwischen Unternehmen, bei denen Unternehmensmodelle Kernstücke des täglichen Geschäfts sind, ist das gegenseitige syntaktische und semantische Verständnis dieser Modelle. Dies zu erreichen erscheint sehr schwierig, da kaum ein Unternehmen eine Modellierungsplattform einsetzt, die verschiedene Modellierungssprachen unterstützt. Folgerichtig ist in den letzten Jahren die Bedeutung der sich über den Lebenszyklus eines Unternehmens erstreckenden Modellierung in vergleichbarem Maße gewachsen wie die Ansprüche an die Interoperabilität dieser Modelle. Die Heterogenität der Modellierungssprachen ist hauptsächlich dadurch bedingt, dass verschiedene Sprachen auf unterschiedliche Modellierungsdimensionen fokussieren und unterschiedlichste Möglichkeiten bieten, die verschiedenen Aspekte eines Unternehmens darzustellen und auszudrücken. Im Zuge dieses Ansatzes wird insbesondere die Prozessdimension fokussiert. Dabei hat sich die Ereignisgesteuerte Prozesskette (EPK) zur semi-formalen Modellierung von Geschäftsprozessen als eine an den Bedürfnissen des Geschäftsprozessmanagements ausgerichtete Adaption eines Petrinetzes etabliert [He02]. Der Grundgedanke des hier beschriebenen Konzeptes ist es, ein Format vorzustellen, das es ermöglicht, Abbildungen von unterschiedlichsten Modellierungssprachen in eben dieses gemeinsame Format zu überführen und somit die EPK interoperabel zu gestalten. Neben diesem Austausch von Prozessmodellen soll darauf aufbauend eine Methode vorgestellt werden, die es erlaubt, die EPK über ein Intermediatorformat in die Workflowsprache XPDL zu überführen. Mit Hilfe dieses Konzeptes wird die vollautomatische Abwicklung kollaborativer Geschäftsprozesse (wie bspw. die Bearbeitung eines Bestellauftrages) möglich, was wiederum eine Effizienzsteigerung der Kollaboration und eine Verbesserung der Interoperabilität zur Folge hat.

2. Transformation und Mapping als zentrale Konzepte zur Gestaltung und Ausführung von Unternehmensmodellen in verteilten Umgebungen

In der wissenschaftlichen Diskussion und praktischen Umsetzung gibt es eine Vielzahl von Ansätzen zur Umwandlung von Unternehmensmodellen (eine Klassifikation findet sich bspw. in [CzHe03]). Im Zuge dieser Ausarbeitung sollen zwei Arten von Umwandlungen näher betrachtet werden. Zum einen wird ein Mechanismus vorgestellt, der es erlaubt, semi-formale Modelle zu formalisieren, um so deren Ausführung zu ermöglichen. Hier wurde in der wissenschaftlichen Literatur eine Vielzahl von Ansätzen diskutiert. Diese lassen sich unterteilen in Konzepte, die eine direkte Transformation einer semi-formalen Beschreibungssprache in eine formale postulieren (bspw. [Zi05]) und solche Ansätze, die eine Transformation über ein Intermediatorformat realisieren. Beide Konzepte finden auch im Zuge der Model Driven Architecture Anwendung, bei der zur Softwareentwicklung zwischen den Ebenen Computation Independent Models (CIM), Platform Independent Models (PIM) und Platform specific Model (PSM) unterschieden wird [MDA]. Im Zuge dieses Konzeptes werden die MDA-Ebenen uminterpretiert und an die Bedürfnisse der Unternehmensmodellierung angepasst. In Anlehnung an das Forschungsprojekt ATHENA werden die folgenden drei Ebenen unterschieden [At05a]:

- *Business-Ebene*: Diese Ebene fokussiert auf die Modellgestaltung unter Planungs- und Steuerungsgesichtspunkten. Die Modelle dienen hauptsächlich zu Dokumentationszwecken und sind in der Regel wenig formalisiert. Auf dieser Ebene ist es wichtig, dass der Modellierer nicht durch Formalisierungsvorschriften eingeschränkt wird und eine grafische Repräsentation der Modelle möglich ist. Als Modellierungssprache wird im Zuge des hier vorgestellten Konzepts die erweiterte ereignisgesteuerte Prozesskette (eEPK) verwendet.
- *Technische Ebene*: Die technische Ebene dient als Intermediator zwischen der Business- und Ausführungsebene. In der Regel ist eine direkte Ausführung der Modelle der Business-Ebene nur mit einem verhältnismäßig hohen Aufwand möglich. Aus diesem Grunde werden auf der Technischen Ebene die Sprache der Business-Ebene formalisiert und für die Ausführung notwendige Informationen ergänzt. Als Intermediatorsprache wird hier eine formalisierte Version der eEPK verwendet, die in Kapitel 4 näher vorgestellt wird.
- *Ausführungsebene*: Auf der Ausführungsebene sind die Modelle soweit zu formalisieren, dass sie von Workflow-Engines interpretierbar sind. Im Zuge des hier vorgestellten Ansatzes ist das anvisierte Ausführungsformat die Workflow-Sprache XPDL.

Eine Transformation der Modelle von der Business-Ebene zur Technischen Ebene ist in der Regel nur teilautomatisiert möglich, wohingegen die Überführung von Modellen der Technischen Ebene auf die Ausführungsebene vollautomatisch abläuft [At05a]. In dieser Ausarbeitung ist eine Transformation von Modellen per Definition ebenenübergreifend, muss aber nicht unbedingt sprachübergreifend sein. So kann eine Sprache, die auf der Business-Ebene nicht formalisiert ist mit Hilfe von Modellierungskonventionen in ein formalisiertes Format der Technischen Ebene überführt werden.

Neben der Transformation von Modellen wird insbesondere in der Praxis angestrebt, sprachlich diversifizierte Modelle mit identischem Formalisierungsgrad ineinander zu überführen. Dies wird aufgrund der Tatsache notwendig, dass es insbesondere bei semi-formalen Modellen eine Vielzahl von toolunterstützten Modellierungsmethoden gibt, die auf unterschiedliche Modellierungsdimensionen fokussieren. Beispielhaft werden in dieser Ausarbeitung die Dimensionen Prozess, Produkt, Organisation, Entscheidung und Infrastruktur näher betrachtet. Diese Umwandlung, die innerhalb einer Formalisierungsebene stattfindet, wird im Folgenden als Mapping bezeichnet. Die Differenzierung zwischen Transformation und Mapping ist deshalb notwendig, da im Zuge des Mapping angestrebt wird möglichst alle Dimensionen und Modellinformationen in ein anderes Format zu übertragen. Der Grundgedanke beim Mapping ist es deshalb, ein möglichst generisches Format zur Verfügung zu stellen, das es, kombiniert mit einem methodischen Leitfaden, ermöglicht, Abbildungen von unterschiedlichsten Modellierungssprachen in eben dieses gemeinsame Format zu überführen. Demzufolge zielt das Mappingformat in seiner Funktion als ein Modellaustausch- oder Abbildungsinstrument darauf, die Interoperabilität zwischen kooperierenden Unternehmen, die zugleich unterschiedliche Modellierungssprachen einsetzen, zu verbessern. Bei der Transformation geht es hingegen nicht um einen umfassenden Austausch von Modellinformation, sondern darum semi-formale Modelle in ein formales und ausführbares Modell zu überführen. Dazu ist oftmals eine Anreicherung der Modelle mit

Ausführungsinformationen notwendig. Abbildung 1 visualisiert die zuvor beschriebenen Zusammenhänge.

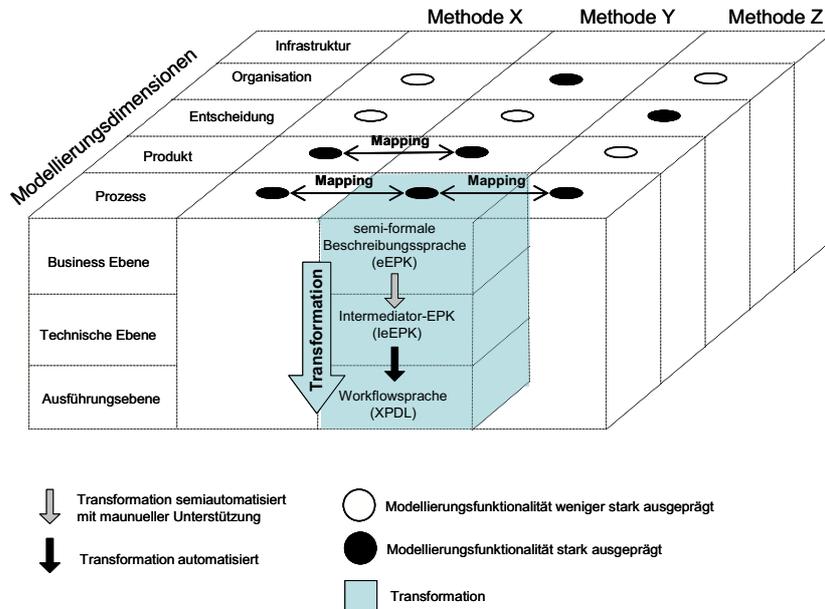


Abbildung 1: Transformation und Mapping

Ziel des hier vorgestellten Ansatzes ist es, basierend auf den Ergebnissen des vom Bundesministerium für Bildung und Forschung (BMBF) geförderten nationalen Forschungsprojektes „P2E2 – Peer-to-Peer Enterprise Environment“ sowie dem europäischen Forschungsvorhaben „ATHENA – Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications“, ein Konzept zu entwickeln, das es ermöglicht die verschiedenen Dimensionen semi-formaler Unternehmensmodelle unterschiedlicher Formate ineinander zu überführen und die Prozessdimension mit Hilfe eines Transformationskonzeptes zur Ausführung zu bringen. Wie Abbildung 1 zeigt, wird nur die Prozessdimension zur Ausführung gebracht, da sie die dynamische Abbildung der anderen Dimensionen ist.

3. POP* als Methode zum Mapping von Unternehmensmodellen

Wie zuvor beschrieben bedingt die Forderung nach einer verbesserten Interoperabilität oftmals den Austausch oder die gemeinsame Nutzung von Modellen. Das Mappingkonzept hat deshalb zum Ziel, ein Instrument zum Austausch von Modellen anzubieten, welches ein generisches Format benutzt, in dem die elementaren Modellierungskonstrukte enthalten sind. Durch ein Mapping von individuellen Modellierungssprachen in dieses Format wird den Unternehmen der Austausch von Modellen eben dieser Modellierungssprache ermöglicht. Das erwähnte Instrument zum Austausch von Modellen ist die POP*-Methode, (nachstehend POP*, was für „Process, Organisation, Product and others“ steht) die im Zuge

des Forschungsprojektes ATHENA entwickelt wird [At05b]. Das hier dargestellte Austauschformat befindet sich aktuell in der Entwicklung und kann sich zukünftig noch ändern.

Das POP*-Metamodell ist, in „Paketen“ organisiert, die fünf Dimensionen abbilden und steht unter dem Einfluss verschiedener existierender Ansätze wie BPDM und UEML 1.0. Es existiert je Dimension ein Paket, zuzüglich eines Paketes, das die allgemeinen Konstrukte enthält, die nicht zu einer spezifischen Dimension gehören. Die Dimensionen lassen sich folgendermaßen beschreiben:

1. Das Paket *Allgemeine Konzepte* beinhaltet Konzepte und Beziehungen, die in jeder Dimension angewandt werden können.
2. Die *Organisationsdimension* fokussiert Organisationsstrukturen, menschliche Wesen und deren Interaktion.
3. Die *Prozessdimension* wiederum enthält Konstrukte mit Bezug auf die Aktivitäten, Aufgaben und Prozesse im oder zwischen Unternehmen.
4. Die *Produktdimension* findet Anwendung, um Produktarchitekturen oder Produktstrukturen zu modellieren.
5. *Entscheidungsstrukturen* bezüglich Entscheidungszentren und Entscheidungsaktivitäten werden mithilfe der Entscheidungsdimension dargestellt.
6. Mit den Konstrukten der *Infrastrukturdimension* wird die Modellierung von Infrastrukturen und den von ihnen erbrachten Leistungen ermöglicht.

POP* zielt in seiner Funktion als ein Modellaustausch- oder Abbildungsinstrument darauf, die Interoperabilität zwischen kooperierenden Unternehmen, die zugleich unterschiedliche Modellierungssprachen einsetzen, zu erleichtern. Im aktuellen Stadium ist POP* so ausgelegt, dass es die Interoperabilität der von den ATHENA-Partnern bereitgestellten Modellierungssprachen ermöglicht. Da von ATHENA ein beträchtlicher Teil des Einsatzgebietes von Modellierungssprachen abgedeckt wird, bleibt zu hoffen, dass der Aufwand, weitere Sprachen zu unterstützen nicht all zu hoch ausfällt. Aktuell wird das POP*-Metamodell von einem UML-Klassendiagramm repräsentiert, das in Abbildung 2 abgebildet ist und im Wesentlichen die generische Struktur des POP*-Metamodells beinhaltet. Dies bedeutet im Einzelnen:

- Alle Konzepte im POP*-Metamodell sind Spezialisierungen der Klasse *Object*.
- Alle Beziehungen im POP*-Metamodell sind Spezialisierungen von *Relationship*.
- Alle Eigenschaften eines Objekts oder einer Beziehung sind Spezialisierungen von *Property*.

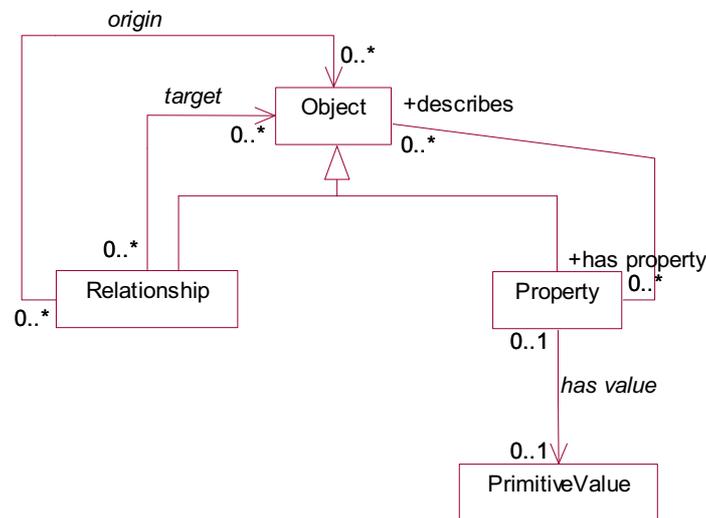


Abbildung 2: Das POP* Kernmodell

Obwohl der Rest des POP*-Metamodells zur Repräsentation auf denselben Formalismus wie das Kernmodell fußt, werden die restlichen Elemente des Metamodells nicht direkt als Spezialisierungen der Kernkonzepte aus Abbildung 2 veranschaulicht. Zu Zwecken der Klarheit und Lesbarkeit werden die folgenden Konventionen in der Notation angewendet:

- Unterklassen von *Objects* (Konzepte des POP* Metamodells) werden als *Klassen* dargestellt;
- Unterklassen von *Relationship* (Beziehungen des POP* Metamodells) werden als *Verbindungen* dargestellt, mit Ausnahme von Beziehungen, die Beziehungen zu anderen Objekten haben (dies bezieht sich augenblicklich nur auf das Konstrukt *Flow*). Im beschreibenden Teil sind alle Beziehungen als "einfache" Klassen mit *origin* (Ausgangspunkt) und *target* (Ziel) als Eigenschaften aufgeführt;
- Unterklassen von *Property* werden in den Darstellungen nicht abgebildet. *Property*-Unterklassen werden im Beschreibungsteil als Eigenschaften der Konzepte die sie beschreiben aufgeführt;
- Wie in den vorangegangenen Aufzählungspunkten dargestellt, findet ein spezifischer typographischer Stil Anwendung, wann immer Konstrukte des POP*-Metamodells im Text oder in Bezeichnungen Verwendung finden. Jedoch wird dieses Schriftbild nicht angewendet, wenn die Rede von *Objekten der „realen Welt“* die Rede ist, die von den Konstrukten bezogen werden. Beispielhaft sei hier das Beispiel des Konstrukts *Process* angeführt, das einen *Prozess* im Unternehmen referenziert.

Das POP*-Metamodell ist, wie in Abbildung 3 zu sehen, in „Paketen“ organisiert. Es existiert je Dimension ein Paket, zuzüglich eines Paketes, das die allgemeinen Konstrukte enthält, die nicht zu einer spezifischen Dimension gehören.

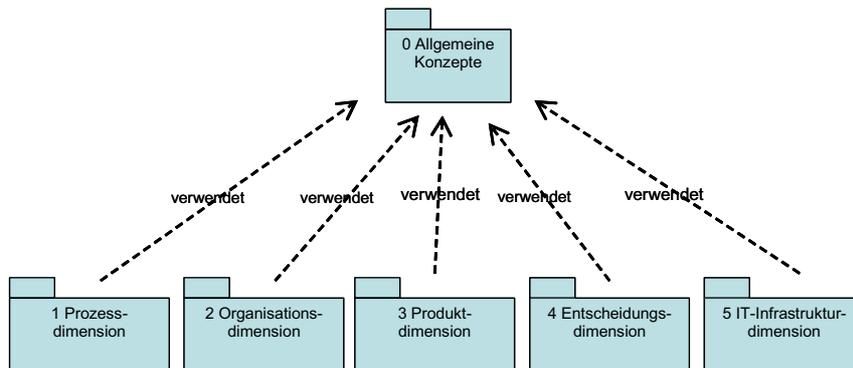


Abbildung 3: Überblick POP* Metamodell

Bislang lag der Schwerpunkt von POP* auf der Prozessdimension, aus diesem Grunde wird in diesem Kapitel exemplarisch für alle anderen Dimensionen das Metamodell der Prozessdimension erläutert.

Die Prozessdimension beinhaltet Konstrukte, die sich auf Aktivitäten, Aufgaben und Prozesse im oder zwischen Unternehmen beziehen. Teilhabe von Objekten des Unternehmens an Prozessen unterschiedlichen Typs wird durch Rollen ausgedrückt. Die Prozesslogik drückt sich in Flüssen und Entscheidungsknoten aus. Eine Übersicht der Modellierungskonstrukte der Prozessdimension ist in Abbildung 4 dargestellt.

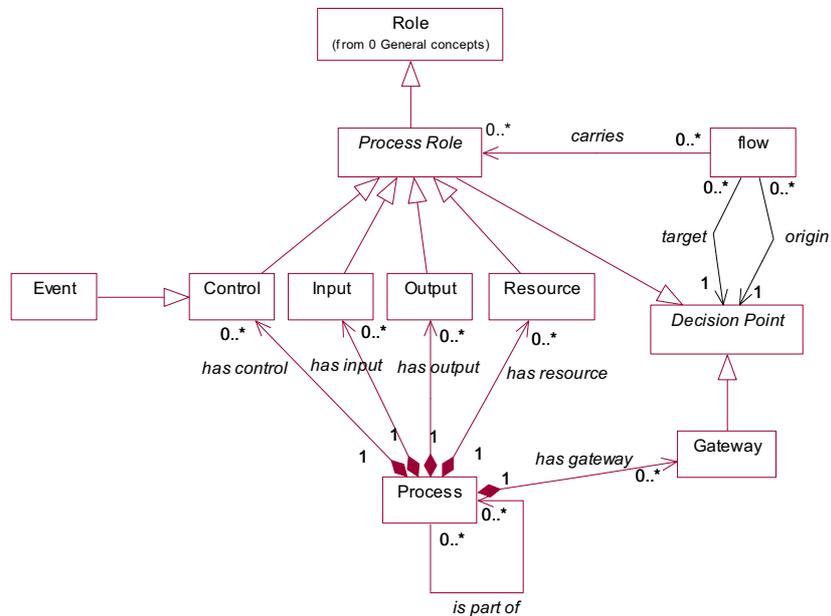


Abbildung 4: Das POP*-Metamodell: Prozessdimension

Process wird dabei verwendet um Prozesse, Aufgaben und Aktivitäten jeglicher Art darzustellen. Ein Prozess kann eine Prozessstruktur beinhalten und Teil einer solchen sein (z.B. Projektplan) und zwar durch eine *is part of*-Beziehung. Darüber hinaus können Prozesse zu Sequenzen mit Hilfe von Flüssen und Entscheidungsknoten kombiniert werden. Decision Point ist ein generisches Konzept, das – wenn es durch Flows verbunden wird – die allgemeine Prozesssequenz bestimmt. Sowohl Decision Point als auch Process Role sind abstrakte Konzepte, so dass in einem realen Prozessmodell jeder Ausgangspunkt und jedes Ziel eines Flusses entweder ein Event, ein Control-Element, ein Input-Element, ein Output-Element, ein Resource-Element oder ein Gateway-Element sein muss. Das Verhalten der Entscheidungsknoten wird im Wesentlichen durch die Eigenschaften *in-flow logic* und *conditional continuation* bestimmt, welche wiederum beschreiben, wie multiple Flüsse zu oder von einem Entscheidungsknoten zu behandeln sind. *In-flow logic* spezifiziert wie die eingehenden Flüsse zu kombinieren sind und kann folgende drei Werte annehmen:

1. 'and': Alle eingehenden Flüsse müssen aktiviert sein, damit der Entscheidungsknoten aktiviert wird.
2. 'xor': Sobald genau einer der eingehenden Flüsse aktiviert wird, wird der Entscheidungsknoten aktiviert
3. 'or': Wenigstens einer der eingehenden Flüsse muss aktiviert werden, damit der Entscheidungsknoten aktiviert wird.

Gateway ist einer von zwei Untertypen von Decision Point, der andere trägt die Bezeichnung Process Role. Obwohl beide Typen Bedingungen für die Fortsetzung eines Prozesses ausdrücken, gibt es einen charakteristischen Unterschied. Gateways sind reine Entscheidungsknoten, wohingegen Process Roles zusätzlich zu ihrem Charakter als Entscheidungsknoten auch mit Unternehmensobjekten in Verbindung stehen können via plays role-Beziehungen.

Process Role ist sowohl ein Untertyp von Role als auch von Decision Point, eine Tatsache, die die dualistische Natur des Konzepts veranschaulicht. Wie Roles können alle Process Role Unternehmensobjekte durch die plays role-Beziehung anbinden. Als Decision Point können sie die Bedingungen für Fortsetzung eines Prozesses repräsentieren in welchem die Process Role enthalten ist. Eine Process Role wird spezialisiert in einem von vier Untertypen: Input, Output, Control, oder Resource.

Ein Input gehört zu einem spezifischen Prozess und kann, als Untertyp von Role, repräsentieren, was in den Prozess eingebracht wird. Als ein Untertyp von Decision Point repräsentiert ein Input eine Bedingung, die seinen „Elternprozess“ startet. Ein Output gehört ebenfalls zu einem spezifischen Prozess und kann, wiederum als Untertyp von Role, darstellen, was aus einem Prozess hervorgeht (z.B. das Ergebnis). Als ein Untertyp von Decision Point, repräsentiert ein Output eine Bedingung für die weitere Ausführung seines „Elternprozesses“.

Auch Resource ist ein Untertyp von Process Role. Eine Ressource gehört zu einem spezifischen Prozess und, in den meisten Fällen, dienen Ressourcen dazu, Rollen und nicht Entscheidungsknoten darzustellen (wobei auch letzteres möglich ist). Als ein Untertyp von Role, werden Ressourcen als Platzhalter für unterschiedlichste Objekte benutzt oder im Prozess erzeugt.

Ein weiteres zentrales Konzept dieses generischen Austauschformats ist das Flow-Konzept. Ein flow ist eine Beziehung zwischen zwei Entscheidungsknoten, Process Roles oder Schnittstellen in einem Prozess. Zusätzlich drückt ein Fluss implizit eine Beziehung zwischen den Prozessen aus, die über Entscheidungsknoten verbunden sind. Im Standardfall ist ein Fluss als Kontrollfluss zu interpretieren. Ein solcher Kontrollfluss wird auf zwei Arten verwendet:

- um zeitliche Abfolgen zwischen Prozessen darzustellen
- um die Auslösung des betroffenen Prozesses kenntlich zu machen

Mit Hilfe dieser generischen Beschreibungselemente ist es möglich mit Pop* eine Vielzahl von Konstrukten aus verschiedenen Sprachen abzudecken. Nur so können die sehr heterogenen Beschreibungsmechanismen unterschiedlicher Sprachen abgedeckt werden. Eines der Hauptprobleme dieser sehr offenen und generischen Beschreibung bleibt der Interpretationsspielraum, der zu Missinterpretationen der unterschiedlichen Konstrukte führen kann. Nachdem in diesem Abschnitt ein generisches Austauschformat zum Mapping von Unternehmensmodellen im Allgemeinen und Prozessmodellen im Speziellen vorgestellt wurde, soll im nächsten Kapitel auf die Transformation von Prozessmodellen eingegangen werden. (das hier vorgestellte Pop*-Austauschformat wurde im Teilprojekt A1 von ATHENA entwickelt und findet sich in der folgenden Spezifikation wieder: [At05b])

4. Konzepte zur Transformation von Prozessmodellen mit Hilfe einer formalisierten eEPK

Nach der Erläuterung des Mapping der Prozessdimension, soll hier aufgezeigt werden, wie eine ebenenübergreifende Transformation zu ermöglichen ist. In Anlehnung an das ARIS-Konzept ist die Prozessdimension die zentrale Dimension, die die weitgehend statischen Dimensionen Organisation, Produkt, IT-Infrastruktur und Entscheidung integriert. Somit wird nur die Prozessdimension als dynamische Abbildung der anderen Dimensionen auf die Technische Ebene übertragen und letztendlich zur Ausführung gebracht.

Wie bereits erläutert hat sich zur semi-formalen Modellierung von Geschäftsprozessen die Ereignisgesteuerte Prozesskette (EPK) als eine an den Bedürfnissen des Geschäftsprozessmanagements ausgerichtete Adaption eines Petrinetzes etabliert [He02]. Zentrales Merkmal der EPK bildet die Veranschaulichung der zu einem Prozess gehörenden Funktionen in deren zeitlichlogischer Abfolge. Damit die Regeln und Bedingungen zur Beschreibung der Kontrollflusssteuerung berücksichtigt werden können, kommen Verknüpfungsoperatoren zur Anwendung. Wesentliche Objekttypen der EPK sind damit folgende Elemente:

- *Funktionen.* Eine Funktion transformiert ein Objekt von einem Startzustand in einen Endzustand. Die Transformation dient der Erzielung einer definierten Leistung, die einen wertschöpfenden Bestandteil innerhalb des Geschäftsprozesses darstellt.
- *Ereignisse.* Ein Ereignis ist eine passive Komponente, die Systemzustände bzw. betriebswirtschaftliche Bedingungen repräsentiert, die Einfluss auf den weiteren Verlauf des Geschäftsprozesses ausüben können.
- *Verknüpfungsoperatoren.* Zur Modellierung des Kontrollflusses werden neben gerichteten Kanten konjunktive (AND), disjunktive (XOR) und adjunktive (OR) Konnektoren zugelassen. Diese begünstigen eine im Vergleich zu Petrinetzen erhöhte Anschaulichkeit der Modelle [Sc98].

Da das Transformationsformat die statischen Sichten integriert, ist es notwendig die EPK um Konstrukte anderer Sichten zu erweitern, so dass die erweiterte ereignisgesteuerte Prozesskette (eEPK) die Basis des hier beschriebenen Intermediatorformats ist. Um diese semi-formale Beschreibungssprache in eine Sprache zu überführen, die als Intermediator zwischen formalisierten Workflow-Modellen und semi-formalen Modellen dient und somit eine Transformation ermöglicht, sind Modellierungskonventionen notwendig. Im Folgenden werden einige allgemeine Konventionen vorgestellt, um im Anschluss daran die Konventionen zur Formalisierung von Modellen zu erläutern.

Die Verwendung von Modellierungskonventionen reduziert die Varietät der Modelle und dient der Vermeidung von Inkonsistenzen sowie einer Verminderung der Anzahl nachträglicher Modellanpassungen [RS02]. Es existieren allgemeine Formalvorschriften für die Ausgestaltung des Kontrollflusses [Ke97] sowie generelle Bestrebungen zur Formalisierung von Syntax und Semantik der EPK [Ru99]. Ebenso wurde die EPK bereits als Spezifikationssprache für Workflows diskutiert [De02] und durch Formulierung von Übersetzungsregeln eine „tragfähige mathematische Basis“ [LSW97] für formale

Geschäftsprozessmodelle geschaffen. Um eine Transformation von EPK-Modellen in ausführbare Workflows zu ermöglichen, sind im Falle einer automatischen Übersetzung besondere Anforderungen an Formalisierungsgrad und Spezifikation des Prozessmodells zu stellen [De02]. Diese Anforderungen sowie Verfahren zu Verifikation von EPK werden seit Jahren in der Forschung thematisiert [Aa99] [Ki04]. In der gängigen Praxis haben sich bei der Modellierung mit EPK neben den ursprünglich von KELLER/NÜTTGENS/SCHEER definierten Konventionen einige weitere Regeln durchgesetzt, die bei einer Modellerstellung zur Anwendung kommen [Ru99]:

- Eine EPK beginnt und endet mit genau einem Ereignis. Das Startereignis triggert den Beginn des Prozesses, das Endereignis definiert einen Zustand, nach dessen Erreichen der Prozess abgeschlossen ist.
- Ereignisse und Funktionen wechseln sich im Ablauf ab. Das Ergebnis der Ausführung einer Funktion wird durch eine Zustandsänderung definiert, welche dem auf die Funktion folgenden Ereignis entspricht.
- Funktionen besitzen genau eine eingehende und eine ausgehende Kante zur Abbildung des Kontrollflusses. Damit ist die Ausführung einer Funktion über eine eindeutige Input-Output Beziehung gekennzeichnet.
- Nach einem Ereignis steht kein OR- oder XOR-Konnektor, da Ereignisse als passive Komponenten keine Entscheidungskompetenz besitzen und damit den weiteren Kontrollfluss der EPK nicht beeinflussen können [KNS92]. Allerdings wird diese Auffassung nicht von allen Autoren vorbehaltlos geteilt [He02].
- Durch Konnektoren verzweigte Pfade werden nur durch gleichartige Konnektoren wieder zusammengeführt. Beispielsweise dürfen die beiden Kontrollflüsse einer XOR-Verzweigung nicht zu einem späteren Zeitpunkt über einen AND-Konnektor vereinigt werden, da diese Bedingung durch die disjunkte Aufspaltung im XOR-Konnektor nicht erfüllt werden kann.
- Werden mehrere Pfade mit einem Konnektor wieder verbunden, darf der Konnektor nur eine auslaufende Kante besitzen, da andernfalls der Kontrollfluss nicht eindeutig beschrieben wäre.
- Direktverbindungen von mehreren Konnektoren (in Form einer Hintereinanderschaltung) sind zulässig, um mehrfache, komplexe Verzweigungen oder Zusammenführungen zu formulieren.

Da semi-formale Modellierungstechniken keine semantische Korrektheit der Prozessmodelle sicherstellen können [EKO96], ist eine Erweiterung der obigen Konventionen erforderlich, um eine formalere Modellerstellung für eine zumindest teilautomatisierte Überführung in ausführbare Workflow-Modelle zu ermöglichen. Da diesem Ansatz die Prämisse zugrunde liegt, die Modelle nicht zu reinen Dokumentationszwecken, sondern als Grundlage zur Ausführung von Prozessinstanzen zu verwenden, zielen die folgenden Konventionen auf die Vereinfachung der Übersetzung in ausführbare Strukturen.

- Innerhalb von Sequenzen darf auf die Formulierung von Ereignissen verzichtet werden. Bei einem linearen Ablauf einer Kette von Funktionen und Ereignissen haben die Ereignisse (sie werden gelegentlich auch als Trivialereignisse bezeichnet) keinen Einfluss auf das Ergebnis der Sequenz. SCHÜTTE unterscheidet in diesem Zusammenhang zwischen Bereitstellungs- und Auslösecharakter von Ereignissen. In einer Sequenz haben Ereignisse damit Auslösecharakter für die folgende Funktion [Sc98]. Sie dürfen daher entfallen, da sie keine zusätzlichen, für die Ausführung relevanten Informationen bereitstellen.
- Bei der Modellierung von Funktionen muss eindeutig zu erkennen sein, ob es sich um manuelle, automatisch-interaktive oder um vollautomatische Funktionen handelt. SINZ konstatiert bei verteilten Anwendungssystemen eine „Spezifikation der Verteilung betrieblicher Anwendungssysteme korrespondierend mit der Zuordnung betrieblicher Aufgaben.“ [FSA96]. Daher sind manuelle und interaktive Funktionen über eine ungerichtete Kante mit genau einer Organisationseinheit zu verbinden, um eine Zuordnung zu einer personellen Ressource herzustellen. Automatische Funktionen müssen über eine ungerichtete Kante mit einem Anwendungssystem oder einem Anwendungssystemtyp verbunden sein, um den unterstützenden maschinellen Aufgabenträger zu identifizieren.

Eine Verwendung des OR-Konnektors (inklusive ODER) ist nicht zulässig. Dies stellt zwar eine wesentliche Restriktion bzgl. des Freiheitsgrades bei der Modellierung dar, die jedoch aufgrund der Eigenschaften der gängigen Workflow-Sprachen für sinnvoll erachtet wird. Beim Setzen eines Konnektors wird implizit festgelegt, ob es sich um eine Zusammenführung oder eine Aufspaltung des Kontrollflusses handelt. Die korrespondierenden Konstrukte der Workflow-Sprachen bestehen in den Operationen SPLIT und JOIN. Analog werden gelegentlich die deutschsprachigen Synonyme „Verzweigung“ und „Synchronisierung“ verwendet [Kr99]. Wie bereits zuvor beschrieben, leistet die EPK in der Modellierung der ARIS Steuerungssicht auf Business-Ebene eine Integration der statischen Sichten [Sc01]. Im Rahmen der Ausführung werden konkrete Workflows gestartet, die aus den Dimensionen Prozess, Organisation und IT-Infrastruktur bestehen. Während eine direkte Übernahme und Anpassung von Prozess- und Organisationsdefinition aus der Organisationsicht möglich ist, ist die IT-Infrastruktur in Form der unterstützenden Anwendungssysteme kein unmittelbarer Bestandteil von ARIS. Zur Beschreibung der IT-Infrastruktur müssen Anwendungssysteme jedoch bereits auf Business-Ebene modelliert in die Prozessdimension integriert werden. Neben Ereignissen und Funktionen werden für die Spezifikation der Aufgabenträger Organisationseinheiten zur Modellierung von Personalressourcen sowie Anwendungssysteme benötigt. Zum Zweck der Aggregation einzelner Prozessteile, beispielsweise um die Detailmodelle nur für interne Organisationseinheiten zugänglich zu halten und nach außen eine höhere Granularitätsstufe zu veröffentlichen, wird der Objekttyp Prozessmodul bereitgestellt. Zur Darstellung des Kontrollflusses finden die Konnektoren „AND“ und „XOR“ sowie gerichtete Kanten Anwendung. In Abbildung 5 ist eine nach obigen Regeln erstellte EPK anhand eines simplifizierten Beispiels der Stornierung einer Bestellung abgebildet. Der Prozess beginnt mit dem Eintreffen einer Auftragsstornierung. Nachdem die Stornierung eingegangen ist, wird die Funktion Stornierung erfassen von einem Mitarbeiter der Organisationseinheit Vertrieb Inland ausgeführt. Der Mitarbeiter benötigt für die Bearbeitung der Funktion das Anwendungssystem KRE-TA.

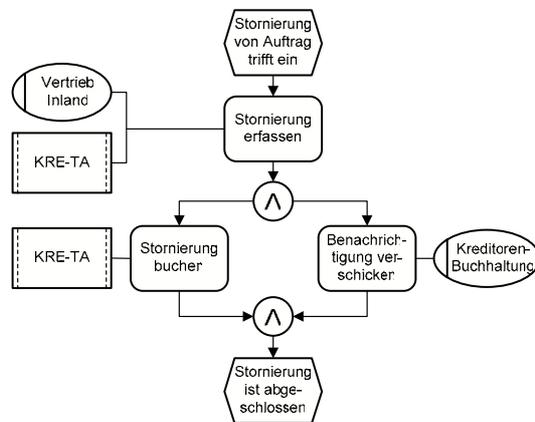


Abbildung 5: Beispiel einer regelkonformen EPK (IeEPK)

Damit handelt es sich um eine automatisch-interaktive Funktion, die durch das Zusammenwirken eines personellen Aufgabenträgers und einer IT-Ressource ausgeführt wird. Nach Erfassung der Stornierung werden die Funktionen Stornierung buchen sowie Benachrichtigung verschicken ausgeführt. Die Buchung geschieht ausschließlich unter Inanspruchnahme des Anwendungssystems »KRE-TA« (vollautomatische Funktion), während das Verschicken der Benachrichtigung ausschließlich von einer Organisationseinheit (und damit manuell) durchgeführt wird. Mit dem Abschluss beider Funktionen ist der Prozess beendet. Mit Hilfe dieser Intermediator-EPK ist es möglich semi-formale Modelle der Business-Ebene über die Technische Ebene auf die Ausführungsebene zu transformieren. Im Folgenden wird diese formalisierte EPK als Intermediator-eEPK (IeEPK) bezeichnet.

5. Prototypische Realisierung der Mapping- und Transformationskonzepte

Nachdem im vorigen Kapitel die konzeptionellen Grundlagen beschrieben wurden, soll hier aufgezeigt werden, wie diese Konzepte im Zuge der Forschungsprojekte P2E2 und ATHENA in Prototypen umgesetzt wurden.

5.1. Prototypische Realisierung des Pop*-Mapping-Konzeptes

Im Zuge des europäischen Forschungsprojektes ATHENA wurde das zuvor beschriebene Pop*-Konzept zum Mapping von Unternehmensmodellen prototypisch realisiert. Um Unternehmensmodelle zwischen Unternehmen zu mappen ist vorgesehen, die Modelle von unterschiedlichen Modellierungsstandards in einen gemeinsamen Standard zu überführen, das POP*-Format. Hierfür wurde eine Transformationsmethode von EPK-Modellen, die mit dem ARIS-Toolset modelliert wurden, in das POP*-Format prototypisch umgesetzt. In

dieser Ausarbeitung sollen die Grundlagen dieses Mapping-Mechanismus kurz beschrieben werden. Sie bilden die Basis einer Java-Applikation, in der diese Konzepte unter der Verwendung des DOM für XML-Transformationen als Basis dienen. Das Programm importiert und exportiert AML (ARIS Markup Language)-Dateien – ein Format, das zur Speicherung von ARIS-Modellen dient.

Hier soll nun lediglich das Mapping grundlegender Elemente der EPK betrachtet werden. Diese Elemente und ihre Beziehungen untereinander werden in der folgenden Abbildung dargestellt.

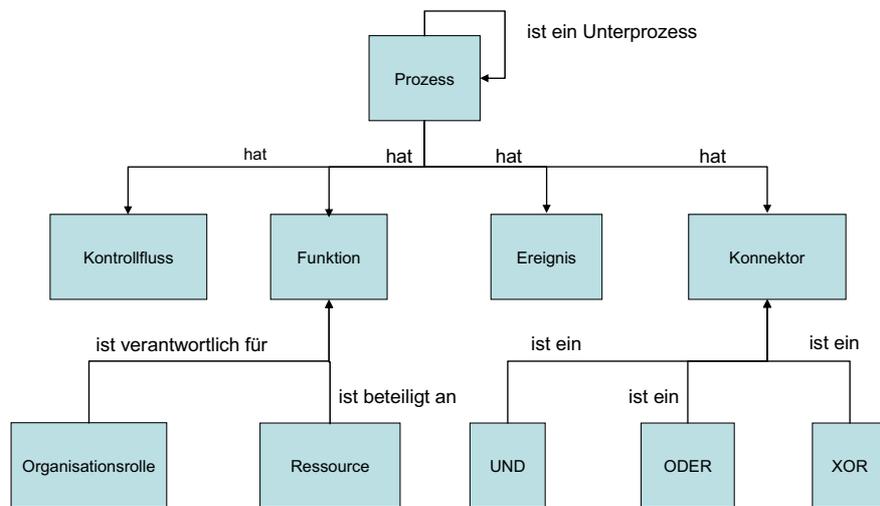


Abbildung 6 Mapping grundlegender eEPK-Elemente

Es können zwei Mapping-Richtungen unterschieden werden. Zum einen sollen mit Hilfe des Pop*-Standards eEPK-Modelle in Modelle anderer Sprachen exportiert werden. Die wesentlich wichtigere Transformationsrichtung für diesen Ansatz ist jedoch der Modellimport. Durch den Import von Modellen, die in anderen Sprachen erstellt wurden, wird es überhaupt erst möglich diese Modelle mit Hilfe der hier vorgestellten Methode auf die Ausführungsebene zu transformieren.

Für den Export ist es notwendig die zuvor beschriebenen Elemente aus der vom ARIS-Toolset generierten AML-Datei zu extrahieren. Die folgende Tabelle bildet die EPK-Quellelemente und die aus der Transformation resultierenden korrespondierenden POP*-Elemente ab.

Tabelle 1: Export von EPK-Modellen

EPK	POP *
Process	Process
Function	Process
AND-Split	Gateway; out-flow logic = AND;
OR-Split	Gateway; out-flow logic = OR;
XOR-Split	Gateway; out-flow logic = XOR;
AND-Join	Gateway; in-flow logic = AND;
OR-Join	Gateway; in-flow logic = OR;
XOR-Join	Gateway; in-flow logic = XOR;
Control Flow (Edge/Link)	Relationship.Flow
Event	State
Organizational Role of a function	ProcessRole/ Role

Um AML-Dateien aus POP*-Daten zu erzeugen, also eine andere Sprache in eine EPK umzuwandeln, werden die folgenden Mappings vorgenommen:

Tabelle 2: Import von EPK-Modellen

POP *	EPK
POP* Process	EPC Process
Decision Point	Connector
Flow	Control Flow
Flow state	Event
Gateway	Connector
Gateway.ConditionalContinuation	Function/Connector
Gateway.inFlowLogic	Connector
Input	Resource attached to function
Output	Only in extended EPC
Output.name	Only in extended EPC
OutputDescription	Only in extended EPC
Process	Process
Process.State	Event
ProcessRole.Name	Organizational Role of a function
Process.hasPart	Subfunction/Hierarchical Function
Resource	Resource
Resource .Activated	-
Resource .ConditionalContinuation	-
Resource .Description	-
Resource .In-flow-logic	-
Resource .Name	Resource (name)
Role	-

Prototypisch umgesetzt wurden die zuvor beschriebenen Implementierungsvorgaben in einem Java-Programm, das es ermöglicht die Modellierungssprache IEM in eine eEPK zu überführen und vice versa. Die Benutzeroberfläche dieses Tools ist in Abbildung 7 zu sehen.

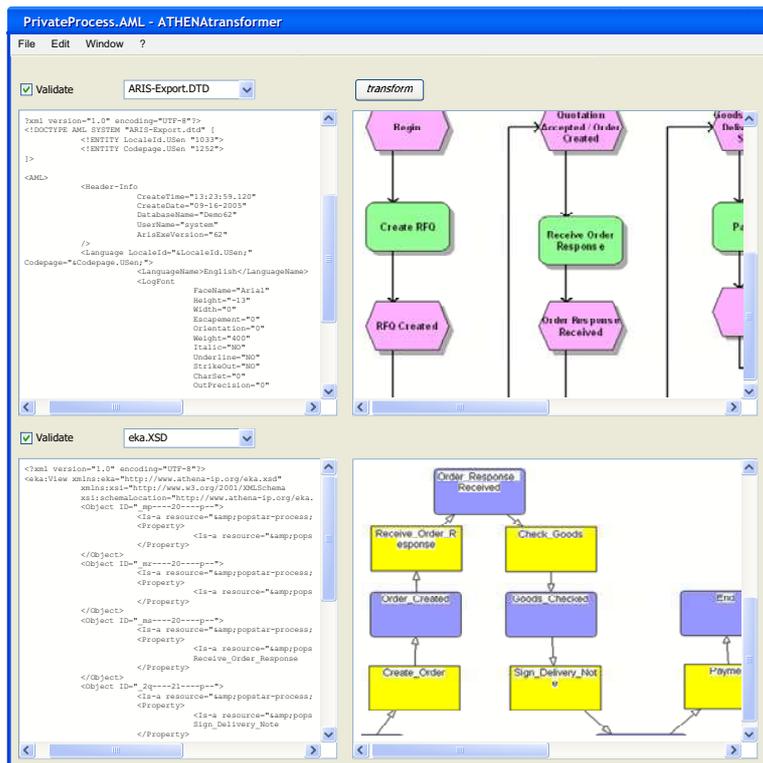


Abbildung 7: Prototyp zum Mapping von Prozessmodellen mit Hilfe von Pop

Die noch nicht vollständig gelöste Aufgabe ist der Abgleich unterschiedlicher POP*- und eEPK-Elemente. Bislang werden nur die grundlegenden prozessbasierten Elemente beider Sprachen transformiert. In den kommenden Entwicklungsphasen müssen weitere Dimensionen (weitere Organisations- und IT-Applikationskonstrukte, etc.) transformiert werden (der hier vorgestellte Prototyp wurde im Projekt ATHENA entwickelt und die Beschreibung und Spezifikation ist in [A05c] zu finden).

5.2. Prototypische Realisierung der Transformation der EPK zu XPDL

Nachdem zuvor ein Prototyp zum Mapping von Unternehmensodellen vorgestellt wurde, soll hier erläutert werden, wie eine ebenübergreifende Transformation prototypisch zu realisieren ist. Der hier beschriebene Prototyp wurde in dem Forschungsprojekt P2E2 entwickelt. Dabei wäre sowohl die Implementierung eines vollständigen Exportformats wie auch die notwendigen Erweiterungen der Workflow-Engines für das Einlesen noch

lückenhaft spezifizierter XPD-Dateien mit einem erheblichen Entwicklungsaufwand verbunden gewesen, der den Rahmen des Projektes überschritten hätte. Aus diesem Grund wurde ein alternativer Lösungsweg zur Überführung der Modelle auf der Businesssebene in eine Spezifikation gewählt. Dieser sieht vor, die mit dem ARIS Toolset erstellte EPK als Grundgerüst für eine zusätzliche Anreicherung mit den für die WFMS essentiellen Informationen zu verwenden. Über ein Annotationswerkzeug werden die rudimentären Modelle in einem Zwischenschritt um zusätzliche, implementierungsspezifische Attribute erweitert und erst anschließend in die Workflow-Engines eingelesen. Das Annotationswerkzeug fungiert damit als Intermediator zwischen dem betriebswirtschaftlichen und dem informationstechnologischen Prozessverständnis und ist somit als Prototyp der technischen Ebene zuzuordnen. Die Felder der Eingabemasken des Annotationswerkzeugs entsprechen den von der XPD-Spezifikation geforderten XML-Attributen. Bereits in der Businessspezifikation enthaltene Attribute werden übernommen, können jedoch fallweise vom Benutzer modifiziert werden. Die Implementierung des Tools erfolgte in der Programmiersprache Java. In Abbildung 8 ist die Annotation der Modelle anhand eines Screenshots des Prototypen illustriert.

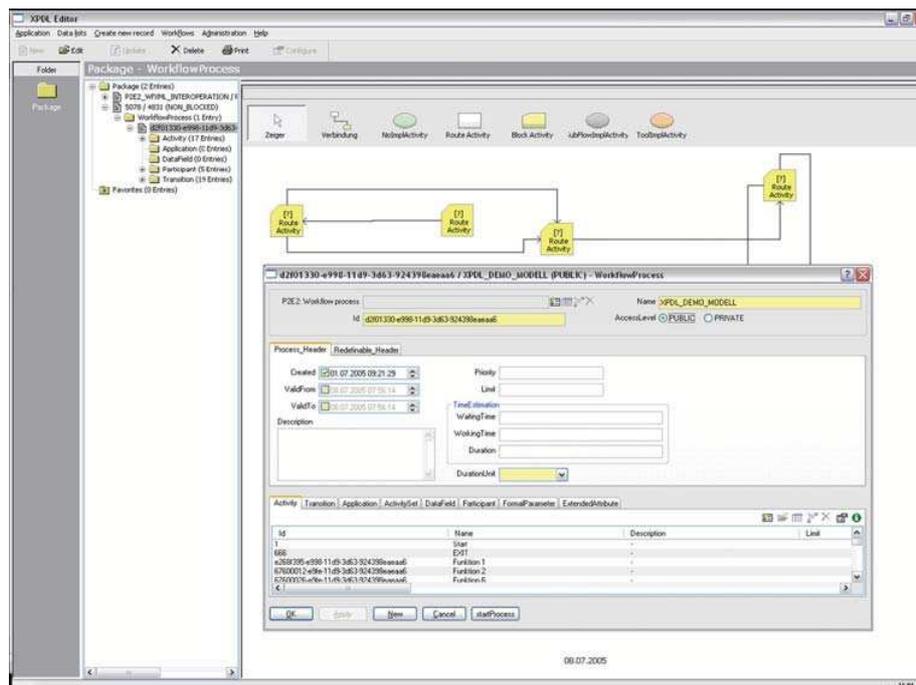


Abbildung 8: Verfeinerung des Workflow-Modells

Das über das Annotationswerkzeug angereicherte Modell kann direkt über die Configuration-API in die Workflow-Engines importiert werden. Zur Konfiguration der Workflow-Engines wurden die Softwareprodukte der Partner um eine Schnittstelle für den XPD-Import erweitert. Damit können die übersetzten EPK Modelle eingelesen und um interne Konfigurationsparameter erweitert werden. Hier werden die bislang fehlenden, für eine Ausführung jedoch zwingend erforderlichen Detailinformationen wie interne

Bezeichnungen, Datenformate, Anmeldeinformationen etc. sowie eventuell erforderliche topologische Änderungen ergänzt. Zur besseren Nachvollziehbarkeit der vorgenommenen Annotationen besitzen sowohl die Produkte von abaXX als auch die von Carnot eine grafische Modellierungsoberfläche. Am Beispiel der Carnot Process Engine wird in Abbildung 9 gezeigt, wie das durch Unterstützung des ARIS Toolset sowie des Annotationswerkzeugs zum XPDL-Standard konforme Prozessmodell in einer proprietären Notation dargestellt und editiert wird.

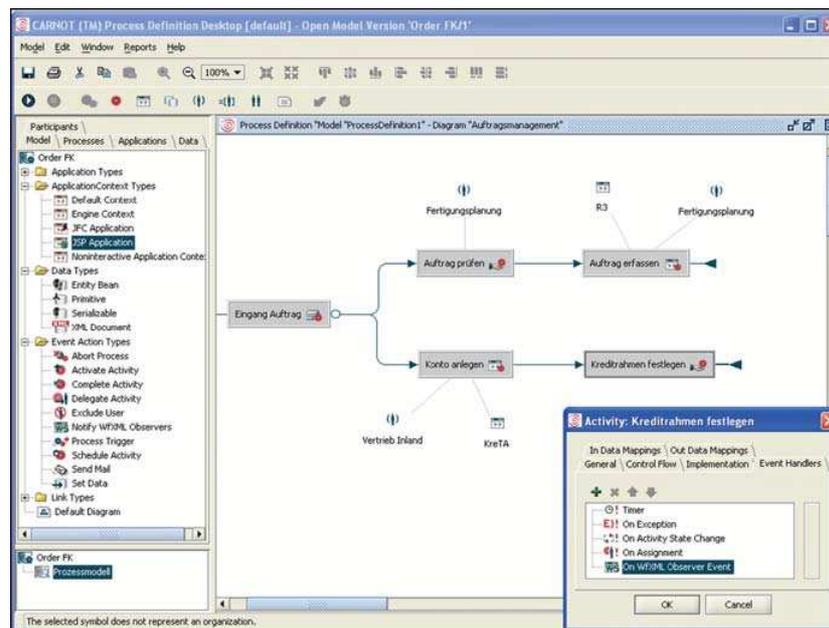


Abbildung 9: Konfiguration der Carnot Process Engine

Der Schwerpunkt des Definitionswerkzeuges der Process Engine liegt jedoch nicht in der grafischen Repräsentation, sondern vor allem in der exakten und vollständigen Konfiguration der Laufzeitumgebung. Einer Aktivität werden zahlreiche Attribute zugeordnet, die deren Ausführungsbedingungen, Parametrisierungen sowie für die Ausführung relevante interne oder externe Ereignisse und Ausnahmen definieren. Das vervollständigte Modell der Konfigurationsdaten wird als sog. Audit Trail in einer beliebigen relationalen Datenbank abgelegt, die über eine JDBC-Schnittstelle verfügt und steht der Process Engine ab diesem Zeitpunkt zur Verfügung.

6. Ausblick

In dieser Ausarbeitung wurden mit dem Mapping und der Transformation zwei zentrale Integrationskonzepte vorgestellt. Es konnte aufgezeigt werden, wie diese Konzepte konzeptionell zu realisieren und prototypisch umzusetzen sind. Des Weiteren wurde erläutert wie das Zusammenspiel beider Konzepte zur Erhöhung der Interoperabilität von

Organisationen beitragen kann. Als zukünftige Herausforderung für das Mapping von Unternehmensmodellen ist insbesondere die Ausweitung des Ansatzes auf weitere Sprachen zu sehen. Erst mit der Evaluation des Formats mithilfe mehrerer Sprachen kann dessen Praxistauglichkeit untermauert und evaluiert werden. Im Zuge der Transformation ist insbesondere die Transformation von der Business-Ebene auf die Technische Ebene weiter zu verfeinern und die Möglichkeiten einer vollautomatischen Transformation zu prüfen. Sowohl für das Mapping als auch für die Transformation verspricht die semantische Annotation von Petrinetzen ein erhebliches bislang kaum ausgeschöpftes Potenzial. Selbst unter der Prämisse eines funktionierenden Austauschs von Unternehmensmodellen über mehrere Sprachen, bleibt immer noch das Problem der Missinterpretation von Modell- bzw. Konstruktbezeichnungen. Und auch bei der Transformation würde eine semantische Annotation automatisierbarer Funktionen eine erhebliche Vereinfachung bedeuten.

Literaturverzeichnis

- [Aa99] Dehnert, J.: Making EPCs fit for Workflow Management. In: Nüttgens, M.; Rump, F. J. (Hrsg.): EPK 2002 – Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens (Trier, November 2002). Trier : GI-Arbeitskreis Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, 2002.
- [An03] Andrews T et al.: Business Process Execution Language for Web Services, Version 1.1. May 2003.
- [At05a] ATHENA Consortium, Specification of a Cross-Organisational Business Process Model – D.A2.2. ATHENA, Integrated Project - Contract n°:IST-507849, 2005, <http://www.athena-ip.org>
- [At05b] ATHENA Consortium, Report on Methodology description and guidelines definition – D.A1.3.1, ATHENA, Integrated Project - Contract n°:IST-507849, 2005, <http://www.athena-ip.org>
- [At05b] ATHENA Consortium, MPCE Specification – D.A1.5.1, ATHENA, Integrated Project - Contract n°:IST-507849, 2005, <http://www.athena-ip.org>
- [CzHe03] Czarnecki, K. ; Helsen, S.: Classification of Model Transformation Approaches. In: OOPSLA'03 Workshop on Generative Techniques in the Context of Model-Driven Architecture, 2003.
- [De02] Dehnert, J.: Making EPCs fit for Workflow Management. In: Nüttgens, M.; Rump, F. J. (Hrsg.): EPK 2002 – Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens (Trier, November 2002). Trier : GI-Arbeitskreis Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, 2002.
- [EKO96] Vgl. Elgass, P.; Krcmar, H.; Oberweis, A.: Von der informalen zur formalen Geschäftsprozessmodellierung. In: Vossen, G.; Becker, J. (Hrsg.): Geschäftsprozessmodellierung und Workflow-Management. Bonn [u. a.] : Thomson Publishing, 1996.
- [FSA96] Ferstl, O. K.; Sinz, E. J.; Amberg, M.: Stichwörter zum Fachgebiet Wirtschaftsinformatik. Bamberger Beiträge zur Wirtschaftsinformatik (Nr. 36). Bamberg, 1996.
- [He02] Heimig, I.: Grammatikbasierte Beschreibung von Geschäftsprozessen. Methodik für das strukturierte Verarbeiten von Modellen. Dissertation. Wiesbaden : Deutscher Universitäts-Verlag, 2002.
- [INT05] o. V.: INTEROP: Presentation of the Project. URL <http://interop-noe.org/INTEROP/presentation>. Abruf am 2005-03-23.
- [Ke97] Keller, G.; Teufel, T.: SAP R/3 prozeßorientiert anwenden: Iteratives Prozeß-Prototyping zur Bildung von Wertschöpfungsketten. 2. Auflage. Bonn [u. a.] : Addison-Wesley, 1997, Edition SAP.
- [Ke03] Kersten, W.; Kern, E.-M.; Held, T.: Auf dem Weg zur E-Collaboration: Entwicklungslinien im Electronic Business. In: Kersten, W.: E-Collaboration: Prozessoptimierung in der Wertschöpfungskette. Deutscher Universitäts-Verlag, Wiesbaden 2003, S. 8-9.

- [Ki04] Kindler, E.: On the Semantics of EPCs: A Framework for Resolving the Vicious Circle. In: Desel, J.; Pernici, B.; Weske, M. (Hrsg.): Business Process Management. Berlin [u. a.] : Springer, 2004.
- [KNS92] Vgl. Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage »Ereignisgesteuerter Prozeßketten (EPK)«. In: Scheer, A.-W. (Hrsg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik ; Nr. 89. Saarbrücken : Institut für Wirtschaftsinformatik, 1992
- [Kr99] Vgl. Kreplin, K.-D.: Konkordanz englischer und deutscher Begriffe des Workflow Management. WfMC, 1999.
- [LSW97] Langner, P.; Schneider, C.; Wehler, J.: Prozeßmodellierung mit ereignisgesteuerten Prozeßketten (EPKs) und Petri-Netzen. In: Wirtschaftsinformatik, 39 1997, Nr. 5).
- [MDA] OMG: Model Driven Architecture. Available at <http://www.omg.org/mda/>
- [Rö01] Röhricht, J.; Schlögel, C.: cBusiness: Erfolgreiche Internetstrategien durch collaborative business am Beispiel mySAP.com. Addison-Wesley, München 2001.
- [RS02] Rosemann, M.; Schwegmann, A.: Vorbereitung der Prozessmodellierung. In: Becker, J.; Kugeler, M.; Rosemann, M. (Hrsg.): Prozessmanagement. Ein Leitfaden zur prozessorientierten Organisationsgestaltung. Berlin [u. a.] : Springer, 2002.
- [Ru99] Rump, F. J.: Geschäftsprozeßmanagement auf der Basis ereignisgesteuerter Prozeßketten : Formalisierung, Analyse und Ausführung von EPKs. Dissertation. Stuttgart; Leipzig : Teubner, 1999.
- [Sc98] Schütte, R.: Grundsätze ordnungsmäßiger Referenzmodellierung. Konstruktion konfigurations- und anpassungsorientierter Modelle. Band 233, Neue betriebswirtschaftliche Forschung. Wiesbaden : Gabler, 1998
- [Sc01] Scheer, A.-W.: ARIS – Modellierungsmethoden, Metamodelle, Anwendungen. 4. Auflage. Berlin [u. a.] : Springer, 2001.
- [Sc04] Scheer, A.-W., Werth, D., Kahl, T., Martin, G.: Lösungen für das Unternehmen von morgen - Next Generation Business. In: IM - Fachzeitschrift für Information Management & Consulting, imc AG, Saarbrücken, 2004.
- [Zi05] Ziemann, J.; Mendling, J.: Transformation of EPCs to BPEL – A pragmatic approach. Akzeptiert für 7th International Conference on the Modern Information Technology in the Innovation Processes of the industrial enterprises, Genua, Italien, September 2005.