

INSTITUT FÜR INFORMATIK
LUDWIG-MAXIMILIANS-UNIVERSITÄT
MÜNCHEN



Diplomarbeit

**Konzeption einer semi-automatischen
Modelltransformation von BPMN-Prozessen
zu SAP Business Workflows**

Lyudmyla Hasay

Aufgabensteller: Prof. Dr. Martin Wirsing
Betreuer: Dr. Martin Rappl (SWM GmbH),
Christian Kroiß (LMU München)

Abgabedatum: 13. Februar 2009

Eidesstattliche Erklärung

Ich erkläre, dass ich die vorliegende Arbeit selbständig verfasst, nicht anderweitig für Prüfungszwecke vorgelegt, alle benutzten Quellen und Hilfsmittel angegeben, sowie wörtliche und sinngemäße Zitate gekennzeichnet habe.

München, den 12.02.2009

Zusammenfassung

Die vorliegende Arbeit entstand in Zusammenarbeit mit den Stadtwerken München (SWM), welches nach einer Effizienzsteigerung im Prozessmanagement strebt. Momentan werden bei SWM die Geschäftsprozesse und Workflows parallel und teilweise unabhängig voneinander von verschiedenen Arbeitsgruppen dokumentiert und modelliert. Es wird nun ein Vorgehen angestrebt, welches die Ableitung technischer Workflows direkt aus Geschäftsprozess-Modellen ermöglicht. Als Modellierungsnotation für die Geschäftsprozesse wurde von SWM die Business Process Modelling Notation (BPMN) vorgeschlagen. Die Workflows werden zum größten Teil durch die SAP Komponente „SAP Business Workflow“ gesteuert.

Da BPMN sehr viele Freiheitsgrade aufweist, können erstellte Modelle nicht immer eindeutig in SAP Workflows überführt werden. Deswegen wurde im Rahmen dieser Diplomarbeit ein hierzu geeignetes Metamodell ausgearbeitet, welches diese Freiheiten einschränkt und die Implementierung in SAP erleichtern soll. Anschließend wurde für einen ausgewählten SAP-Bereich ein anwendungsspezifisches Metamodell erstellt. Zuletzt wurde das BPMN-Metamodell auf das SAP-Metamodell abgebildet.

Inhaltsverzeichnis

Inhaltsverzeichnis	4
1 Einleitung.....	6
1.1 Motivation und Zielsetzung	6
1.2 Aufbau der Arbeit	7
2 Prozessmodellierung.....	8
2.1 Geschäftsprozess vs. Workflow.....	8
2.2 Geschäftsprozess-Modellierung.....	10
2.2.1 Modellierungsansätze und Methoden	10
2.2.2 Modellierung mit BPMN	11
2.3 Workflow-Modellierung mit SAP	18
2.4 Vorschlag für die Herleitung des Workflow-Gerüsts	22
3 Metamodelle für BPMN und SAP Workflow.....	25
3.1 Grundlagen der Metamodellierung.....	25
3.2 BPMN-Metamodell für die Geschäftsprozess-Modellierung bei SWM.....	26
3.2.1 BPMN Core	27
3.2.2 BPMN GraphicalObject.....	31
3.2.3 BPMN Activity	32
3.2.4 BPMN Event.....	37
3.2.5 BPMN Gateway	40
3.2.6 BPMN ConnectingObject.....	43
3.2.7 BPMN Swimlane	46
3.2.8 BPMN Artifact.....	47
3.2.9 BPMN SupportingType	48
3.3 Metamodell für den SAP Business Workflow.....	50
3.3.1 Sap Core.....	51
3.3.2 SAP Workflow-Definition.....	55
3.3.3 SAP Schritt	60
3.3.4 SAP Aktivitaet	63
3.3.5 SAP Steuerungsschritt	67
4 Modell-zu-Modell Transformation vom BPMN-Modell zum SAP Workflow	79

4.1	Grundlagen der Modell-zu-Modell Transformation	79
4.2	Transformationsregeln von BPMN ^(SWM) -Metamodell zum SAP Workflow-Metamodell	79
4.2.1	BPMN Core Transformation	80
4.2.2	BPMN Activity Transformation	80
4.2.3	BPMN Event Transformation	88
4.2.4	BPMN Gateway Transformation	94
4.2.5	BPMN ConnectingObject Transformation	97
4.2.6	BPMN Swimlane Transformation	97
4.2.7	BPMN Artifact Transformation	97
5	Anwendung des Konzepts anhand des ausgewählten Beispiels	98
6	Bewertung und Ausblick	109
Anhang		110
Abbildungsverzeichnis		110
Tabellenverzeichnis		112
Tabellenverzeichnis		112
Abkürzungsverzeichnis und Glossar		113
Literaturverzeichnis		114

1 Einleitung

Die vorliegende Arbeit entstand in Zusammenarbeit mit den Stadtwerken München (SWM) [SWM09]. SWM ist der größte deutsche kommunale Dienstleister. Das Unternehmen bietet seinen Kunden eine Versorgung mit Strom, Gas, Wasser und Telekommunikation und betreibt Bäder sowie den öffentlichen Verkehr.

Zur Wahrung des eigenen Geschäftserfolgs und zur Optimierung des Kundenservices strebt SWM nach einem effizienten Prozessmanagement. Dieses betrifft sowohl Geschäftsprozesse als auch die automatisierte Ableitung von Workflows.

Momentan werden bei SWM die Geschäftsprozesse mit dem Werkzeug ConSense dokumentiert. Workflows werden dazu parallel und unabhängig direkt in SAP entwickelt. Beide Modelle werden in verschiedenen Arbeitsgruppen erstellt, die in den meisten Fällen über unterschiedliche fachliche und technische Hintergründe verfügen. Dadurch wird teilweise unnötige Mehrarbeit geleistet und es treten zahllose Unstimmigkeiten zwischen den beiden Modellen auf.

Aus diesem Grund wird ein Vorgehen angestrebt, welches die Ableitung technischer Workflows direkt aus Geschäftsprozess-Modellen ermöglicht. Den bereits im Unternehmen vorhandenen Modellen mangelt es aber an der notwendigen Semantik und Detaillierung. Deswegen wurde von SWM als alternative Modellierungsnotation die Business Process Modelling Notation (BPMN) vorgeschlagen, welche eine Schnittstelle zwischen den in den Fachabteilungen modellierten Geschäftsprozessen und der in der IT programmierten Workflows schaffen soll.

1.1 Motivation und Zielsetzung

Es gibt bereits mehrere Ansätze zur Transformation von BPMN auf ausführbare Prozesssprachen, wie z.B. BPEL. Bei SWM werden die automatisierten Prozesse zum größten Teil durch die SAP Komponente „SAP Business Workflow“ gesteuert, welche über eine eigene Notation verfügt. Da es noch keinen solchen Ansatz – Transformation der mittels BPMN modellierten Prozesse auf SAP Workflows – gab, wurden zu diesem Zweck von SWM zwei eng zusammenhängende Diplomarbeiten ausgeschrieben – die vorliegende, sowie die Diplomarbeit „Requirements Engineering für Workflowsysteme“. Für die Bearbeitung wurde eine Arbeitsgruppe aus zwei SWM Mitarbeitern und zwei Diplomanden gebildet. In dieser Arbeitsgruppe wurden einige von SWM ausgewählte Geschäftsprozesse auf Basis der vorhandenen fachlichen Beschreibung mittels der Prozessbeschreibungssprache Business Process Modelling Notation (BPMN) modelliert. Im Rahmen der vorliegenden Arbeit wurden dabei die Verfeinerungsmöglichkeiten auf der Geschäftsprozess-Ebene bis hin zur Workflow-Ebene untersucht. Das dabei identifizierte Modell auf Workflow-Ebene diente anschließend als Vorlage zur Ableitung eines SAP Workflows.

Da BPMN sehr viele Freiheitsgrade aufweist, können erstellte Modelle nicht immer eindeutig in SAP Workflows überführt werden. Deswegen wurde ein hierzu geeignetes Metamodell ausgearbeitet, welches diese Freiheiten einschränkt und die Implementierung in SAP erleichtert. Hierzu wurden geeignete Elemente von BPMN identifiziert (Teilsprache), und die Sprachsemantik entsprechend, teils bereits unter Berücksichtigung von SAP-Anforderungen, anpasst.

Anschließend wurde für einen SAP-Bereich, welcher sich für das Modellieren im Hinblick auf die zuvor ausgewählten Geschäftsprozesse eignet, ein anwendungsspezifisches Metamodell erstellt.

Zuletzt stand die Abbildung des BPMN-Metamodells auf das SAP-Metamodell im Vordergrund. Die konkrete Implementierung dieser Abbildung findet im Rahmen der zweiten Diplomarbeit statt.

Das Ziel dieses Vorgehens war nicht die Erzeugung eines lauffähigen SAP Workflows. Es sollte vielmehr eine Vorlage geschaffen werden, die anschließend weiter verfeinert und mit SAP Hintergrundfunktionalität gefüllt werden kann.

1.2 Aufbau der Arbeit

In Kapitel 2 werden die Begriffe Geschäftsprozess und Workflow sowie die Unterschiede zwischen den beiden erklärt. Hier wird ein Vorgehen zur Modellierung der Geschäftsprozesse in SWM mittels BPMN vorgeschlagen, sowie ein Ansatz zur Erzeugung eines Workflow-Gerüsts in BPMN aus einem Geschäftsprozessmodell, welches dann als Vorlage für den lauffähigen SAP Workflow dienen soll.

In Kapitel 3 werden die im Rahmen dieser Arbeit erstellten Metamodelle zu den beiden Modellierungsnotationen vorgestellt.

Das nachfolgende Kapitel 4 beinhaltet die ausgearbeiteten Transformationsregeln zur Überführung des einen Metamodells in das andere.

In Kapitel 5 wird das Konzept an dem ausgewählten Beispiel Lieferantenwechsel „GeLi Gas“ angewendet.

Kapitel 6 liefert eine Bewertung der erreichten Resultate und gibt einen Ausblick auf mögliche Weiterentwicklungen.

2 Prozessmodellierung

„Prozesse sind das Herz jeder funktionierenden Organisation, unabhängig davon, ob sie dokumentiert oder undokumentiert sind, manuell oder automatisiert ablaufen, optimiert oder nicht“ Carl Frappaolo, AIIM Vice President.

Geschäftsprozesse sind von großer Bedeutung für moderne Unternehmen. Ein gutes Prozessmanagement ist ausschlaggebend für den Geschäftserfolg eines Unternehmens. Und als Grundlage für ein wirksames Prozessmanagement treten immer mehr die Geschäftsprozessmodelle in Vordergrund.

2.1 Geschäftsprozess vs. Workflow

Das einheitliche Verständnis der beiden Begriffe Geschäftsprozess und Workflow hat sich in der Literatur noch nicht eindeutig herauskristallisiert. Sie liegen sehr eng zusammen, da sie beide denselben Gegenstand – Arbeitsabläufe in Unternehmen – beschreiben. Aus diesem Grund setzen viele Autoren sie gleich. Andere dagegen differenzieren zwischen einem Geschäftsprozess und einem Workflow auf Grund der unterschiedlichen Zielsetzung und Detaillierungsgrads.

Für diese Arbeit ist es wichtig, diese Begriffe zu unterscheiden und auseinander zu halten.

„Ein Geschäftsprozess ist eine zielgerichtete, zeitlich-logische Abfolge von Aufgaben. ... Er dient der Erstellung von Leistungen entsprechend den vorgegebenen, aus der Unternehmensstrategie abgeleiteten Prozesszielen.“ [Gad05] S. 36.

„Ein Workflow ist dagegen ein formal beschriebener, ganz oder teilweise automatisierter Geschäftsprozess. Er beinhaltet die zeitlichen, fachlichen und ressourcenbezogenen Spezifikationen, die für eine automatische Steuerung des Arbeitsablaufes auf der operativen Ebene erforderlich sind.“ [Gad05] S. 41.

Der Geschäftsprozess beschreibt, „was“ getan werden muss, um die Unternehmensziele zu erreichen; der Workflow dagegen, „wie“ das konkret geschehen soll. Technische Workflows können außerdem durch eine Workflow Engine ausgeführt werden.

Diese Unterscheidung der beiden Begriffe führt zum Ebenenkonzept in der Prozessmodellierung (vgl. [FG08] S. 51 und 130-132).

Die Beschreibung der Geschäftsprozesse findet auf der Prozess-Ebene statt (siehe dazu Abbildung 2.1). Ein maximaler Detaillierungsgrad ist dann erreicht, wenn ein Prozessschritt (hier Aktivität) von einem Mitarbeiter in einem Zug an einem Arbeitsplatz ausgeführt werden kann. Ein Workflow konkretisiert dabei die letzte Ebene eines Geschäftsprozesses dadurch, dass er die Aktivitäten in Arbeitsschritte zerlegt. Diese Arbeitsschritte können manuelle oder automatisierte Tätigkeiten beschreiben.

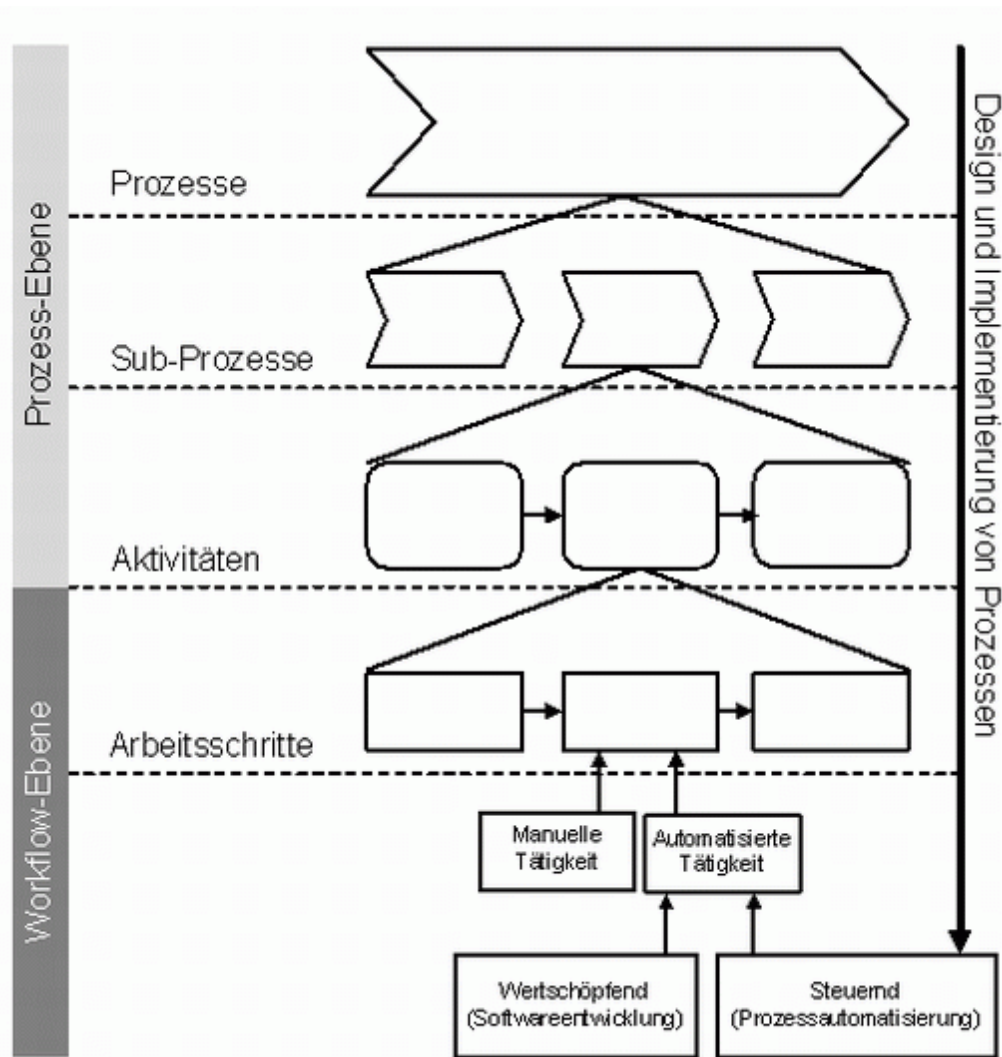


Abbildung 2.1: Top-down-Prozessgestaltung (Quelle [FG08])

Gadatsch unterscheidet 2 Modellansätze im Rahmen der Prozessmodellierung: den einstufigen und den zweistufigen Modellansatz. Bei dem einstufigen Ansatz wird der Workflow direkt modelliert. Bei dem zweistufigen Vorgehen wird zuerst das Geschäftsprozess-Modell erstellt, von welchem dann das Workflow-Modell abgeleitet wird (vgl. [Gad05] S.62). Freund und Götzer verwenden eine viel feinere Klassifizierung der möglichen Wege (insgesamt 5) vom Prozessmodell zum Workflow (vgl. [FG08] S. 128-130).

Bei SWM wird momentan das einstufige Vorgehen (nach Gadatsch) praktiziert. D.h. der Workflow wird direkt in SAP programmiert. Allerdings werden parallel dazu auch die Geschäftsprozess-Modelle zwecks Dokumentation erstellt. Dabei werden beide Modelle in verschiedenen Arbeitsgruppen erstellt, die in den meisten Fällen über unterschiedliche fachliche und technische Hintergründe verfügen. Dieses Vorgehen ist suboptimal, da z.B. ein Teil der bei der Geschäftsprozess-Modellierung gewonnenen Erfahrung verloren gehen kann. Außerdem kann es vorkommen, dass Änderungen oder Op-

timierungen der Geschäftsprozesse nicht bzw. nicht sofort in den Workflow-Modellen berücksichtigt werden. Aus diesem Grund wird angestrebt, den zweistufigen Ansatz (nach. [Gad05]) einzuführen, und die technischen Workflows aus den Geschäftsprozess-Modellen abzuleiten.

Als Werkzeug für die Geschäftsprozess-Modellierung wird bei SWM derzeit ConSense verwendet [ConS]. Da es allerdings nur für die Dokumentation verwendet wird, verfügen die damit modellierten Geschäftsprozesse nicht über die notwendige Semantik, um daraus ohne weiteres einen lauffähigen Workflow ableiten zu können.

Außerdem ist „eine vollständig automatische Transformation des fachlichen Modells in ein technisches Modell und zurück ... nicht erreichbar, da in beiden Modellen die vollständige Information enthalten sein müsste und die gleiche Modellierungssemantik verwendet werden müsste.“ [FG08].

Aus diesem Grund wurde nach alternativen Modellierungsansätzen für die Geschäftsprozessmodellierung bzw. nach einer Schnittstelle zwischen den in Fachabteilungen modellierten Geschäftsprozessen und den in den IT-Abteilungen programmierten Workflows gesucht.

2.2 Geschäftsprozess-Modellierung

2.2.1 Modellierungsansätze und Methoden

Zur Modellierung von Geschäftsprozessen und Workflows haben sich mittlerweile viele Methoden herausgebildet. In [FG08] werden 3 grundsätzliche Möglichkeiten für die Auswahl einer Methode unterschieden:

- Einfache Metaphern
- Formale Modelle
- Programmnahe Modelle

Unter einfachen Metaphern werden hier Modelle verstanden, die sich durch einfache und sehr verständliche Darstellung auszeichnen. Sie bestehen nämlich nur aus Aktivitäten, die in einer logischen Reihenfolge angeordnet sind und sich hervorragend für die Dokumentation der Geschäftsprozesse eignen. So einfach diese Modelle sind, so schnell stoßen sie auf Ihre Grenzen bei der tieferen Prozess-Analyse. Somit sind sie für unsere Ziele nicht geeignet.

Die bekanntesten Repräsentanten der formalen Modelle sind Petrinetze und Ereignisgesteuerte Prozessketten (EPK). Diese Modelle verfügen über wenige Elemente und sind daher auf den ersten Blick leicht und schnell verständlich und eignen sich gut für die Modellierung von Workflows. V.a. die EPK werden sehr erfolgreich in der Praxis eingesetzt. Bei größeren Prozessen können die formalen Modelle allerdings genau wegen der beschränkten Anzahl an Elementen schnell unübersichtlich werden.

Prozessnahe Modelle orientieren sich stark an Abläufen und eignen sich daher bestens für den Einsatz in der Prozess-Modellierung. Die bekanntesten Vertreter dieser Sparte sind UML-Aktivitätsdiagramme sowie die Business Process Modeling Notation (BPMN). UML-Aktivitätsdiagramme beschreiben sehr präzise das dynamische Verhalten eines Systems, können aber für einen nicht IT-Anwender sehr komplex und unverständlich erscheinen.

In dieser Arbeit wurde BPMN zur Geschäftsprozess-Modellierung eingesetzt.

2.2.2 Modellierung mit BPMN

Grundlagen

Die Business Process Modeling Notation (BPMN) wurde 2002 bei IBM entwickelt und von der Business Process Management Initiative (BPMI) veröffentlicht. Im Jahr 2005 wurde die BPMN von der Object Management Group (OMG) übernommen und adaptiert. Die aktuelle BPMN Version zur Zeit der Entstehung dieser Arbeit war die 1.1. Anfang dieses Jahres hat OMG die Version 1.2 veröffentlicht, die sich von BPMN 1.1 inhaltlich nicht unterscheidet, sondern nur einige Bugs beseitigt. Größere Veränderungen sind mit der Version 2.0 zu erwarten, die sich bereits im finalen Ausarbeitungsstadium befindet.

BPMN wurde mit dem Ziel entwickelt, sowohl der IT als auch den Fachabteilungen gerecht zu werden. BPMN ist eine Notation, die sowohl für Nicht-IT-Benutzer intuitiv genug ist, als auch in der Lage ist, die komplexe Prozess-Semantik zu repräsentieren. Darüber hinaus kann ein BPMN Diagramm in eine ausführbare Sprache wie Business Process Execution Language (BPEL) [OAS07] übersetzt werden.

Im Folgenden wird ein kurzer Überblick über die Grundelemente von BPMN gegeben.

In BPMN gibt es 4 Gruppen von Elementen:

- Flow Objects
- Connecting Objects
- Swimlanes
- Artifacts

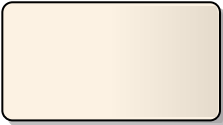
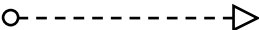
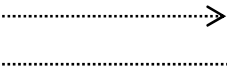
Flow Objects sind die wichtigsten graphischen Elemente, um das Verhalten des Geschäftsprozesses zu definieren. Es gibt drei Arten von Flow Objects: Events, Activities und Gateways.

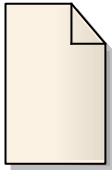
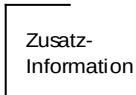
Connecting Objects verbinden die Flow Objects miteinander oder mit anderen graphischen Elementen. Es gibt drei Arten von Connecting Objects: Sequence Flow, Message Flow und Association.

Zum Gruppieren und Kategorisieren der Modellierungselemente gibt es Swimlanes: Pools und Lanes.

Artifacts bringen zusätzliche Informationen über den Geschäftsprozess in die Modellierung ein. BPMN besitzt drei Standard-Arten von Artifacts: Data Object, Text Annotation und Group und bietet die Möglichkeit, eigene Artifacts zu definieren.

Tabelle 2.1: BPMN Grundkonstrukte

Konstrukt	Graphische Notation	Sachverhalt
Event		Ein Event signalisiert, dass ein Ereignis während der Prozessausführung aufgetreten ist. Es gibt 3 Event-Typen: Start, Intermediate und End. Start Event bezeichnet den Anfang des Prozesses, End Event repräsentiert das Prozessergebnis und Intermediate Event beeinflusst den Prozess-Ablauf: erzeugt Ereignisse oder reagiert auf diese oder aber verzögert die Prozessausführung.
Activity		Eine Activity bezeichnet die eigentliche Aufgabe, die während des Prozesses ausgeführt wird. Es gibt 2 Activity-Arten: eine atomare Activity Task und eine zusammengesetzte – Sub-Process.
Gateway		Ein Gateway wird zur Kontrolle vom Sequence Flow verwendet.
Sequence Flow		Ein Sequence Flow verbindet die einzelnen Objekte innerhalb eines Pools und legt die Reihenfolge der Aufgaben im Prozess fest.
Message Flow		Ein Message Flow zeigt den Nachrichtenaustausch zwischen zwei Teilnehmern.
Association		Eine Association verbindet Flow Objects mit den Informationen in Form von Artifacts.
Pool		Pools repräsentieren einen Teilnehmer in einem Prozess. Das können Unternehmen, Geschäftsrollen, Einzelpersonen oder Personengruppen sein.

		Ein Pool kann einen Prozess enthalten. In diesem Fall tritt er auch als Container für andere graphische Objekte auf.
Lane		Eine Lane ist eine Unterteilung eines Pools in Sub-Partitionen (z.B. einer Organisation in Abteilungen oder einer Gruppe in Einzelpersonen). Lane organisiert und kategorisiert Aufgaben eines Prozesses. Es existiert mindestens ein Lane in dem Pool, auch wenn nicht sichtbar.
Data Object		Data Objects werden als Artifacts betrachtet, da sie keinen direkten Einfluss auf Sequence Flow oder Message Flow des Prozesses haben, aber wichtige Informationen darüber besitzen, was die Aufgaben für ihre Ausführung brauchen oder was sie produzieren.
Group		Gruppiert andere Elemente ohne den Flow zu beeinflussen. Kann zur Dokumentations- und Analysezwecken verwendet werden. Group wurde in dieser Arbeit ausgelassen (Gründe dafür sind weiter unten aufgeführt).
Text Annotation		In der Text Annotation kann der Modellierer Zusatzinformationen für die Leser des Diagrams eintragen.

In Abbildung 2.2 ist ein kleines Beispiel für einen in BPMN modellierten Prozess zu sehen:

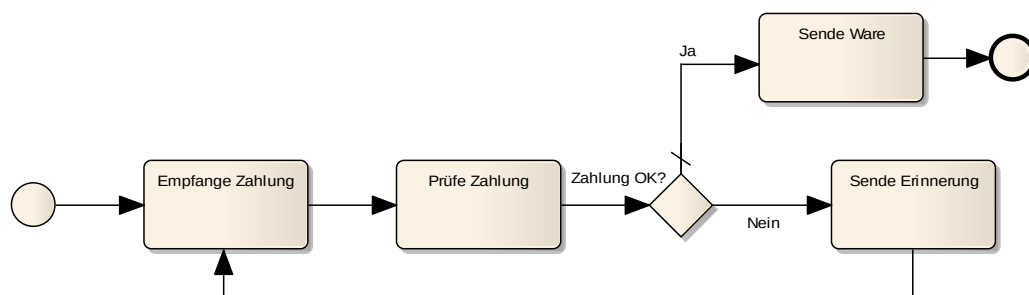


Abbildung 2.2: BPMN Prozess

Die BPMN Spezifikation umfasst mehr als 300 Seiten. Deswegen wird für die nähere Beschreibung auf die BPMN Spezifikation [BPMN10] sowie [BPMN11] verwiesen.

Modellierung der Geschäftsprozesse bei SWM mit BPMN

Das gesamte Modellierungsvorgehen mit BPMN wird hier nicht näher erläutert. Es wird nur auf die bedeutendsten Besonderheiten der Prozessmodellierung bei SWM eingegangen.

Als Modellierungswerkzeug kommt Enterprise Architect von Sparx Systems [EntA71] zum Einsatz, das in der aktuellen Version 7.1 bisher nur BPMN 1.0, nicht aber die neueren Versionen BPMN 1.1 bzw. BPMN 1.2 unterstützt.

Die in BPMN modellierten Geschäftsprozesse werden bei SWM in keine ausführbare Sprache übersetzt, sondern sollen als Vorlage für die in SAP Business Workflow laufenden Workflows dienen.

Aus diesem Grund wurden der gesamte Wortschatz sowie die Semantik von BPMN untersucht. Da BPMN sehr viele Freiheitsgrade aufweist, können erstellte Modelle nicht immer eindeutig in SAP implementiert werden. Deswegen musste der Wortschatz und die Semantik einheitlich begrenzt und unter Berücksichtigung von SAP-Anforderungen angepasst werden.

Im Folgenden werden jeweils zu einem Element (einer Elementengruppe) nur die Sachverhalte erläutert, die in der verwendeten Modellierung einen Unterschied zur BPMN Spezifikation aufweisen:

Events

Folgende Event-Trigger werden bei der Modellierung nicht verwendet:

- Multiple – wird aus Eindeutigkeitsgründen nicht verwendet
- Message Intermediate Event wird nicht im Sequence Flow verwendet – die Send und Receive Tasks haben dieselbe Bedeutung. In [WM08] S. 69 wird empfohlen, sich für eines davon zu entscheiden; daher werden hier konsistent nur Send und Receive Tasks verwendet.
- Cancel und Compensation Events kommen nicht vor, da der Transaction Sub-Process und Compensation Activity herausgenommen werden.

Activities

- Sowohl Reference Task als auch Reference Sub-Process werden einfachheitshalber herausgenommen.

- Transaction Sub-Process und Compensation Activity werden vorerst herausgenommen, da sie nicht auf korrespondierende Elemente eines SAP Workflows abgebildet werden können.

Gateways

- Wegen der zukünftigen Abbildung eines BPMN Modells auf einen SAP Workflow werden die Gateways in auseinander- oder zusammenführende aufgeteilt. Beides gleichzeitig wird nicht erlaubt.
- Aus demselben Grund beschränkt man sich vorerst auf XOR, AND und ein zusammenführendes Complex Gateway. Ein OR Gateway müsste für die Abbildung auf den SAP Workflow in eine Reihe von XOR Gateways zerlegt werden.

Connecting Objects

- Conditional Flow wird nur nach einem XOR Gateway erlaubt, nicht nach einer Activity.
- Sequence Flow Looping wird vorerst herausgenommen.

Artifacts

- Group wird herausgenommen. Auf Grund der EA-spezifischen Modellierung kann sie mit der Activity verwechselt werden.

Wie bereits im Abschnitt 2.1 beschrieben, werden die Geschäftsprozesse des Unternehmens in Ebenen aufgeteilt.

Auf der ersten Ebene werden die zentralen Geschäftsprozesse der SWM repräsentiert. Diese Ebene wird nicht mit BPMN-Mitteln modelliert. Der Enterprise Architect bietet aber eine graphische Darstellung für den BPMN Business Process, welcher dafür benutzt werden kann.

Die Ebene der Sub-Prozesse ist für die abstrakte Darstellung des Prozessflusses zuständig. Die primäre Aufgabe dieser Ebene ist es, das Verständnis über die zusammenhängende Ablaufstruktur im Prozess zu vermitteln. Auf dieser Ebene kommt BPMN zum Einsatz. Die Darstellung muss allerdings nicht 100% BPMN-konform sein. Bei der Modellierung auf dieser Ebene wird folgendermaßen vorgegangen (siehe hierzu auch Abbildung 2.3):

- Für einen ausgewählten Prozess wird ein Diagramm angelegt.
- Für jeden Beteiligten in diesem Prozess wird jeweils ein Pool im Diagramm angelegt, der diesen Beteiligten, z.B. Organisation, oder seine aktuelle Rolle darstellt. (Falls es sich aber um einen unternehmensinternen Prozess handelt, d.h. der mit keinem anderen kommuniziert, muss kein Pool als graphisches Element angelegt werden. In BPMN wird angenommen, dass ein Pool mit unsichtbaren Grenzen vorhanden ist).

- Prozess des aktuellen Beteiligten wird sichtbar in einem aufgeklappten Pool modelliert. Die anderen Teilnehmer werden immer als zugeklappte Pools dargestellt, da ihre internen Abläufe nicht bekannt sind. Pro Diagramm soll es jeweils nur einen aufgeklappten Pool geben. Kollaborative Prozesse werden zur Zeit nicht verwendet.
- Der Geschäftsprozess wird analysiert und in seine ablauf-relevanten Bestandteile zerlegt. Diese Teile werden als Activities überwiegend vom Typ Sub-Process angelegt.
- Die Kommunikation zwischen den Teilnehmern wird mittels Message Flow modelliert, der eine Activity im aufgeklappten Pool mit dem Rand des zugeklappten Pools verbindet. Dabei spielt es keine Rolle, ob auf der darunter liegenden Ebene mehrere Message Flows in dieselbe Richtung laufen. Auf dieser Ebene werden sie nur einmal dargestellt.
- Ein Prozess muss immer ein Start und ein End Event haben. Ein Prozess soll möglichst nur ein Start Event haben, darf aber mehrere End Events haben, falls er unterschiedlich ausgehen kann. Diesen End Events sollen entsprechende Namen zugewiesen werden (vgl. [Sil07] Part 3).
- Falls ein Sub-Prozess mit unterschiedlichen Ergebnissen ablaufen kann, was den Ablauf des übergeordneten Prozesses beeinträchtigt, soll nach diesem Sub-Prozess ein Gateway eingebaut werden, der dies entsprechend darstellt (vgl. [Sil07] Part 5).

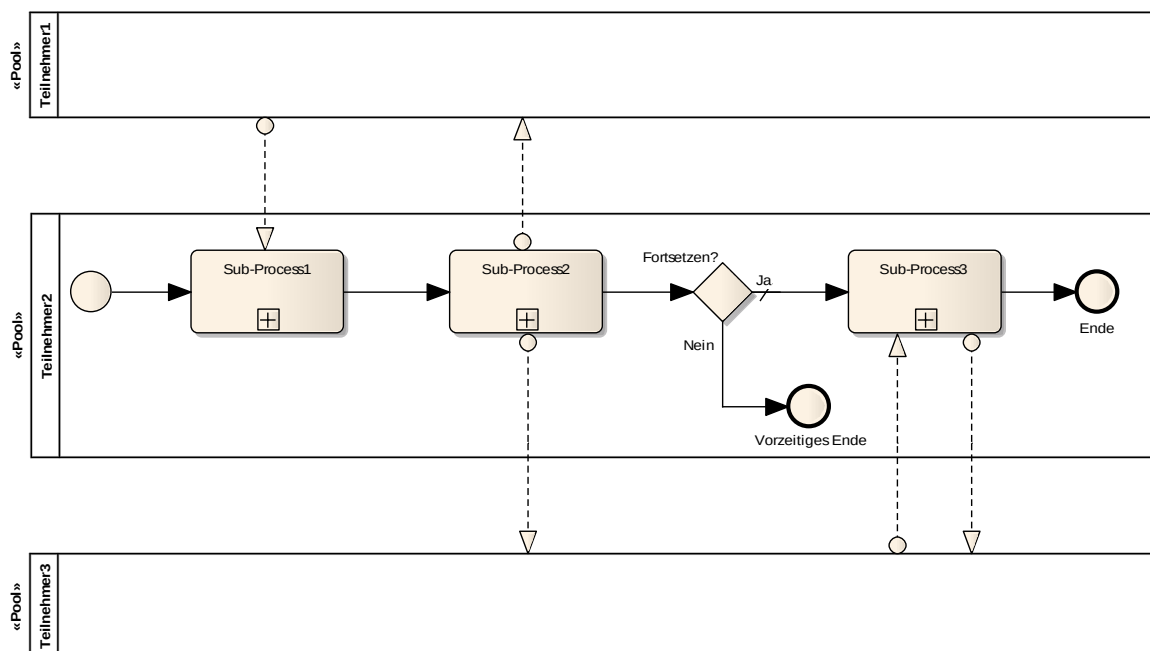


Abbildung 2.3: Geschäftsprozessmodell Ebene 2

Auf der zweiten Modellierungsebene dürfen folgende BPMN-Elemente vorkommen: Pool, Lane, Start und End Event, Sub-Process, selten Task, XOR und AND Gateway, Sequence und Message Flow, Text Annotation.

Auf der Ebene der Aktivitäten werden die abstrakt beschriebenen Sub-Prozesse der oberen Ebene detailliert modelliert. Der Schwerpunkt der Modellierung auf dieser Ebene ist die Repräsentation der tatsächlichen operativen Tätigkeit einzelner Mitarbeiter. Sie kann als Arbeitsanweisung zur Durchführung konkreter Aufgaben betrachtet werden.

Folgendes ist bei der Modellierung dieser Ebene zu beachten:

- Auf dem Diagramm zum entsprechenden Sub-Prozess werden nur die Pools verwendet, mit welchen der Nachrichtenaustausch im betroffenen Sub-Prozess stattfindet.
- Als Start und End Events dürfen nur leere Events (mit None Trigger) vorkommen. Wenn allerdings dieser Sub-Process auch eigenständig, als Prozess der oberen Ebene verwendet wird und durch ein Ereignis ausgelöst werden kann, dürfen 2 Start Events im Modell vorkommen. Dies soll aber nach Möglichkeit vermieden werden.
- Der Nachrichtenaustausch wird wie oben dargestellt. Allerdings werden auf dieser Ebene alle vorkommenden Message Flows modelliert.
- Auf dieser Ebene dürfen alle BPMN-Elemente vorkommen; Tasks überwiegen jedoch.

In Abbildung 2.4 ist ein Beispiel für einen Sub-Prozess auf der 3. Ebene zu sehen.

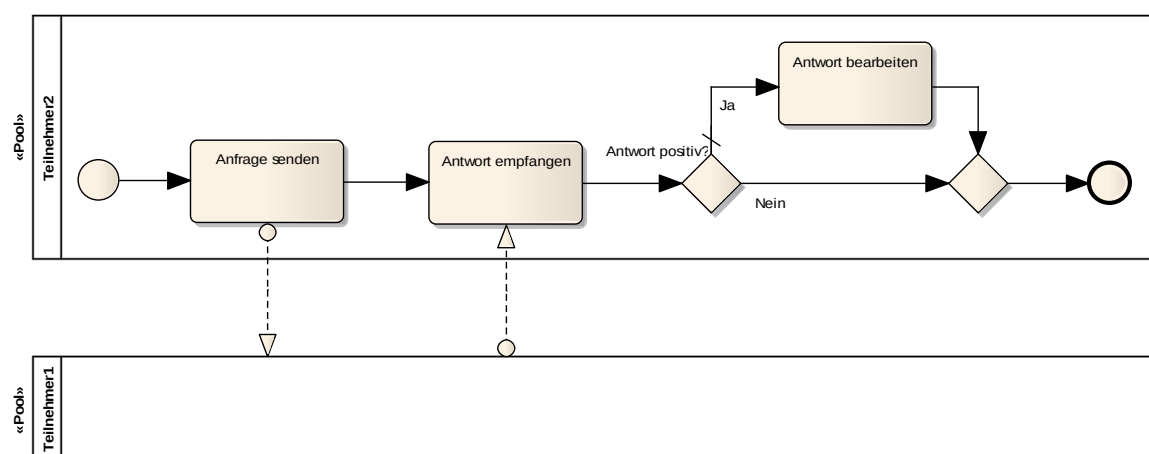


Abbildung 2.4: Geschäftsprozessmodell Ebene 3

Weiter wird aus den Geschäftsprozess-Modellen der 2. und 3. Ebenen ein so genanntes Workflow-Gerüst in BPMN abgeleitet. Dieses Workflow-Gerüst ist aus bereits oben erwähntem Grund kein lauffähiger Workflow. Abgesehen davon soll das BPMN Modell

nicht lauffähig sein – als Workflow Engine kommt SAP Business Workflow bei SWM zum Einsatz.

Aus diesem Workflow-Gerüst wird durch Modell-zu-Modell Transformation der zu diesem Zweck ausgearbeiteten Metamodelle eine Vorlage für den SAP Workflow erstellt (siehe Kapitel 3 und 4). Die technische Realisierung dieser Transformation wird in der zweiten Diplomarbeit dieser Arbeitsgruppe ausgearbeitet.

Bevor aus dem modellierten Geschäftsprozess eine Vorlage für den lauffähigen Workflow abgeleitet wird, wird im Unterkapitel 2.3 ein kurzer Überblick über das in SWM verwendete SAP System gegeben.

2.3 Workflow-Modellierung mit SAP

Wie bereits oben erwähnt wurde, kommt SAP Business Workflow bei SWM als Workflow Engine zum Einsatz.

SAP Business Workflow ist ein Werkzeug von SAP ERP für die Automatisierung der Geschäftsprozesse im SAP System. Er stellt dem Endbenutzer standardisierte vordefinierte Workflow-Muster zur Verfügung, bietet aber auch die Möglichkeit, eigene Workflow-Muster anzulegen oder bereits vorhandene zu modifizieren.

Die Architektur vom SAP Workflow wird in 4 Ebenen aufgeteilt:

- Die unterste Ebene repräsentiert Advanced Business Application Programming-Code (ABAP), der zur Laufzeit gestartet wird.
- Die 2. Ebene bilden die so genannten Business-Objekttypen, die in einem Tool namens Business Object Repository verwaltet werden. Ein Business-Objekttyp ist mit einer Klasse in der Objekt-Orientierten Modellierung vergleichbar. Er beschreibt also eine Menge an konkreten Objekten mit gleichen Merkmalen, Struktur und Verhalten. Alle konkreten Objekte, die im SAP System existieren, sind Instanzen der Business-Objekttypen.

Für den Zugriff auf die Business-Objekttypen verfügen diese über Methoden, Attribute und Ereignisse. Eine Methode repräsentiert wie z.B. auch bei einer Java-Klasse die Operation, die auf einem Objekt ausgeführt werden kann. Sie bezieht sich normalerweise auf eine ABAP-Funktion der untersten Ebene. Ein Ereignis repräsentiert die Zustandsänderung eines Objektes, und die Attribute beschreiben die Eigenschaften und Merkmale eines Objektes.

- Die 3. Ebene ist die eigentliche Ablauf-Ebene, d.h. hier wird eine Folge an Schritten definiert, die zur Ausführung einer Aufgabe dienen. Es wird zwischen den Einzelschrittaufgaben, den so genannten Standardaufgaben, und den Mehrschrittaufgaben, den Workflow-Mustern, unterschieden.

Standardaufgaben beziehen sich auf eine Methode des Objekttyps. Sie können eigenständig oder als Bestandteil des Workflow-Muster ausgeführt werden. Workflow-

Muster bestehen aus mehreren Standardaufgaben sowie Schritten zur Steuerung der Ausführung. Darüber hinaus verfügt ein Workflow-Muster über mindestens eine Workflow-Definition, d.h. eine graphische Beschreibung der Workflow-Schritte und deren Reihenfolge. Zum Datenaustausch wird eine spezielle Struktur verwendet - Container. Ein Container lässt sich mit einer Variablen in einem Java-Programm vergleichen. Der Workflow-Container entspricht dabei einer Klassenvariablen, ein Aufgaben-Container einer Methodenvariablen. Auf diese Ebene wird weiter unten näher eingegangen, da sie als Grundlage für diese Arbeit dient.

- Organisationsebene – die oberste Ebene, die den zuständigen Bearbeiter für einzelne Schritte festlegt.

Es gibt 2 Vorgehen bei der Modellierung von Workflow-Mustern in SAP Business Workflow: Bottom-Up und Top-Down.

Vorgehen beim Bottom-Up Ansatz:

- Business Objekt Repository nach den passenden Objekttypen mit den passenden Methoden, Attributen und Ereignissen prüfen.
- Erforderliche Aufgaben, welche die Methoden realisieren, anlegen.
- Workflow-Muster anlegen und die Aufgaben als Schritte einbinden.
- Zusätzliche Steuerungsschritte (Schleifen, Bedingungen, etc.) definieren.

Grobes Vorgehen bei dem Top-Down Ansatz:

- Workflow-Schritte definieren.
- Methoden der Objekttypen diesen Schritten zuweisen.
- Zusätzliche Steuerungsschritte (Schleifen, Bedingungen, etc.) definieren.
- Mögliche Zusatzfunktionen (Startereignisse, Ausnahmen, Terminüberwachungen, etc.) festlegen.

Diese Arbeit orientiert sich am Top-Down Ansatz, da man davon ausgeht, dass dem Geschäftsprozess-Modellierer nicht bekannt ist, welche Objekttypen und welche Methoden in SAP zur Verfügung gestellt sind.

Wie oben bereits erwähnt wurde, verfügen Workflow-Muster über eine graphische Notation – die so genannte Workflow-Definition. Diese Workflow-Definition wird in einem speziellen Werkzeug – dem Workflow-Builder angelegt.

Die Workflow-Definition wird aus einzelnen Schritten aufgebaut, die sich in 2 Typen unterscheiden lassen:

- Schritte, die sich auf eine Aktivität beziehen: Aktivität, Benutzerentscheidung, Dokument aus Vorlage, Web-Aktivität, Mail versenden, Formularaufgabe.

- Schritte, die der Ablauf-Steuerung dienen: Bedingung, Mehrfachbedingung, Schleife, Paralleler Abschnitt, Container-Operation, Ablaufsteuerung, Ereigniserzeuger, Warten auf Ereignis, Ad-Hoc Anker, Block, Lokaler Workflow. In Abbildung 2.5 und Abbildung 2.6 sind die graphischen Notationen der Schritte in der aufgeführten Reihenfolge dargestellt.



Abbildung 2.5: Schritte, die sich auf eine Aktivität beziehen



Abbildung 2.6: Schritte, die der Ablauf-Steuerung dienen

Außerdem hat die Workflow-Definition standardmäßig einen Start- und einen End-Schritt, die automatisch angelegt werden und nicht bearbeitet werden können. Falls der Workflow über in einem Objekttyp definierte Ereignisse gestartet wird, werden diese auch neben dem Start-Schritt automatisch angelegt und graphisch dargestellt (siehe dazu Abbildung 2.7).

Im Unterschied zu vielen anderen graphischen Modellen ist die Workflow-Definition nicht graphen-, sondern blockorientiert. D.h. ihre elementaren Bauelemente sind nicht Knoten und Kanten, sondern Blöcke, die immer eine konstante Anordnung der Elemente aufweisen. Die Blöcke können entweder hintereinander angeordnet oder ineinander verschachtelt werden, aber niemals die Grenzen voneinander durchqueren.

Ein Block hat immer einen Anfang und ein Ende. Zu einem Block gehört immer ein Schritt. Mit dem Anlegen eines Schrittes wird immer ein Block erzeugt und beim Löschen eines Schrittes wird auch immer ein kompletter Block gelöscht. Alle parallelen oder alternativen Pfade, die ein Schritt erzeugt, werden in demselben Block wieder zusammengeführt.

Ein Beispiel für die Workflow-Definition mit den Schritten und zugehörigen Blöcken ist in Abbildung 2.7 zu sehen. Die Darstellung der Blöcke als farblich hervorgehobene Rechtecke lässt sich abschalten, was aber an der Struktur nichts ändert.

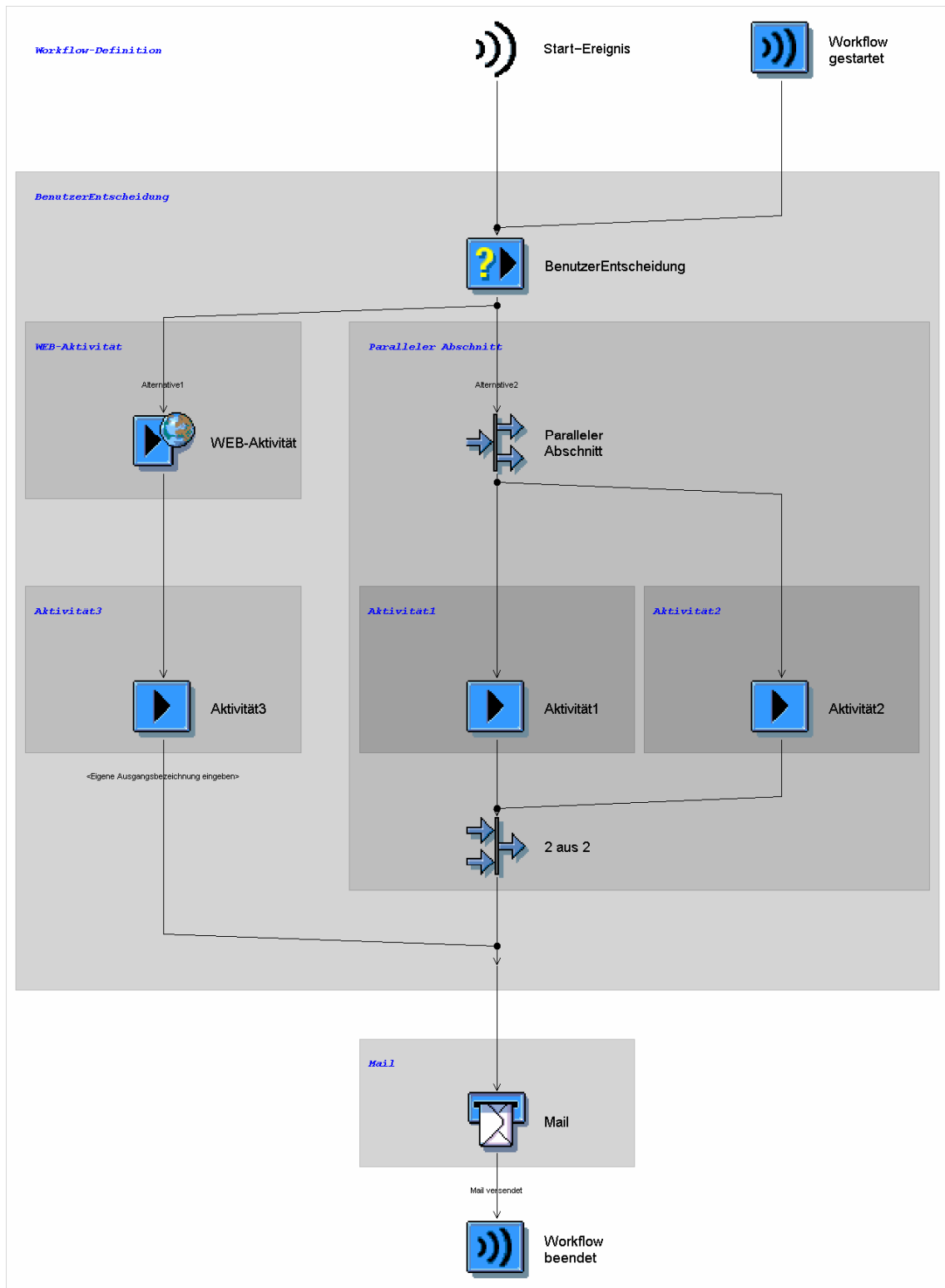


Abbildung 2.7: Beispiel einer Workflow-Definition

2.4 Vorschlag für die Herleitung des Workflow-Gerüsts

Wie Freund in seinem Blog auf [Fre08] zum Ausdruck bringt, ist Prozessverfeinerung allgemein kein triviales Problem. Auch die Herleitung eines Workflows aus dem Geschäftsprozess-Modell gestaltet sich nicht unproblematisch. Allweyer schlägt in [All08a] z.B. vor, die ausführbaren Workflows durch Modell-zu-Modell Transformationen aus den Geschäftsprozess-Modellen zu generieren. Freund und Götzer stellen in [FG08] mehrere mögliche Wege vom Geschäftsprozess zum Workflow dar. Als besonders viel versprechend bezeichnen sie einen Ansatz, in dem die erste Workflow-Fassung aus dem Geschäftsprozess-Modell generiert wird. Dieses, wie sie es nennen, „Proto-Modell“ wird dann manuell an die technischen Erfordernisse angepasst und kann anschließend für die Generierung benutzt werden.

Das Vorgehen in dieser Arbeit lehnt sich teilweise an den von Freund und Götzer vorgeschlagenen Ansatz an – aus dem mittels BPMN modellierten Geschäftsprozess wird eine erste Fassung des Workflows, ebenfalls in BPMN, abgeleitet. Diese wird weiterhin Workflow-Gerüst genannt.

Da allerdings die Workflow-Generierung nicht das Hauptziel dieser Arbeit ist, wird sie hier nicht ausführlich ausgearbeitet, sondern nur kurz vorgestellt. Außerdem erfordert dies viel tiefere Kenntnisse über die Unternehmensprozesse, sowie über das SAP System. Deswegen wurde der Geschäftsprozess zusammen mit den SWM Mitarbeitern analysiert und der Teil identifiziert, welcher mit Hilfe des SAP Systems automatisiert werden kann.

Für das Workflow-Gerüst wird in einem neuen Diagramm das Pool namens „SAP“ angelegt, der das System symbolisieren soll (vgl. auch [All08b]). Bei dem Geschäftsprozess selber wird die unterste ausführliche Prozess-Ebene betrachtet. Zu beachten ist hier, dass es bei Enterprise Architect nicht möglich ist, die Sub-Prozesse aufzuklappen. Deswegen müssen die entsprechenden Diagramme einzeln betrachtet werden.

Bezüglich der BPMN-Elemente aus dem Geschäftsprozessmodell wird wie unten angegeben vorgegangen:

Aktivities

- Bei den Tasks wären das die Tasks vom Typ User, Service, Send, Receive, Script (falls vorhanden). Die reinen Manual Tasks können nicht automatisiert werden. Allerdings wurden die Tasks von diesem Typ in dieser Arbeit teilweise „zweckentfremdet“, um die Benutzerentscheidung zu modellieren, die in SAP auch erfasst werden kann. Die erste Task, in der der Benutzer mit dem System interagiert, d.h. den Workflow startet, wird im Pool „SAP“ auf ein Start Event abgebildet. Für alle anderen werden ihre analogen Repräsentanten ebenfalls als Tasks im Pool „SAP“ angelegt.

- Bei den Sub-Processes wird abhängig von deren Typ vorgegangen: Independent Sub-Processes werden auch im Pool „SAP“ als solche angelegt, Embedded Sub-Processes, die als Looping gekennzeichnet sind oder einen gemeinsamen angehefteten Intermediate Event haben, werden ebenfalls als solche im Pool „SAP“ angelegt; sonst können sie auch in den Hauptablauf integriert werden.

Events

- Als notwendig identifizierte Start Events werden in den Pool „SAP“ übernommen. Ausnahme bilden die Embedded Sub-Processes, die aufgeklappt in den übergeordneten Prozess integriert werden.
- Intermediate Events, die für den automatisierten Ablauf wichtig sind, werden ebenfalls übernommen.
- Da der SAP Workflow nur einen Anfang und ein Ende hat, steht das im Widerspruch zu der Vorgabe, dass in dem Geschäftsprozess abhängig von dem Ergebnis mehrere End Events modelliert werden sollen. Dies kann bei der Ableitung des Workflow-Gerüsts wie folgt gelöst werden. Die End Events eines Sub-Prozesses können auf Service Tasks abgebildet werden, welche diese Ergebnisse im System bekannt machen sollen (falls solche Tasks noch nicht im Geschäftsprozess modelliert wurden; siehe dazu auch Abbildung 5.3 und Abbildung 5.9). Die alternativen Pfade werden mit einem Gateway zusammengeführt und die End Events zu einem End Event zusammengefasst, falls der Sub-Prozess als solcher in das Workflow-Gerüst übernommen wird, oder aufgelöst, falls der Sub-Prozess in den übergeordneten Prozess integriert wird. Auf der Ebene des übergeordneten Prozesses werden mehrere End Events ebenfalls zusammengeführt. Allerdings ist deren wiederholtes Abbilden auf die Service Tasks meistens nicht mehr nötig.

Gateways

- Gateways, die für den automatisierten Ablauf wichtig sind, werden ebenfalls übernommen. Bei den alternativen bzw. parallelen Pfaden, welche durch die Gateways erzeugt werden, muss außerdem die Blockstrukturierung des SAP Workflows berücksichtigt werden. Der Prozess wie in der Abbildung 2.8 kann wegen dem Sequence Flow vom zweiten Gateway zu der TaskB nicht zu SAP transformiert werden.

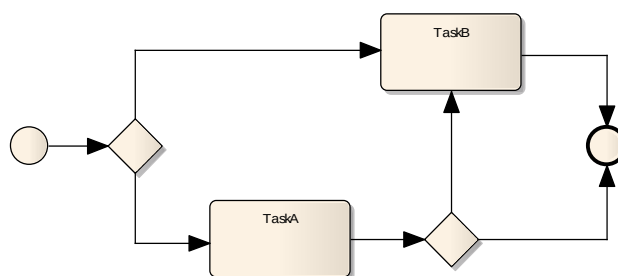


Abbildung 2.8: BPMN Prozess nicht SAP-konform strukturiert

Dieser Prozess muss wie in der Abbildung 2.9. umgeformt werden. In [FG08] wird diese Lösung als unsauber, aber pragmatisch möglich bezeichnet.

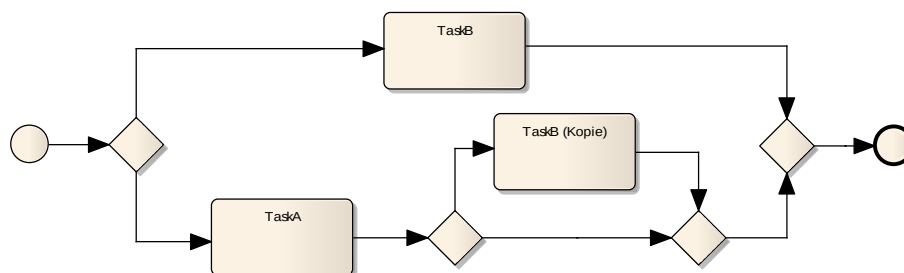


Abbildung 2.9: BPMN Prozess SAP-konform strukturiert

Connecting Objects

- Alle Flow Objects im Workflow-Gerüst werden gemäß der Geschäftsprozesslogik mit dem Sequence Flow verbunden.
- Der Message Flow kommt auf dieser Ebene nicht vor.

Artifacts

- Können ebenfalls übernommen werden, dienen allerdings eher dem besseren Verständnis des Modells. Wie man später sehen wird, werden sie nicht zum SAP Workflow transformiert.

Dieses Workflow-Gerüst kann schließlich vor der Transformation zu SAP auch in BPMN weiter verfeinert werden. Über den Sinn, Grad der Verfeinerung sowie das Kosten-Nutzen Verhältnis soll dann in jedem speziellen Fall entschieden werden; da unabhängig davon, wie detailliert und informationsreich dieses Workflow-Gerüst ist, muss der SAP Workflow mit der internen Funktionalität angereichert werden.

3 Metamodelle für BPMN und SAP Workflow

3.1 Grundlagen der Metamodellierung

Für die Metamodellierung wurde von der Object Management Group (OMG) ein Standard – Meta Object Facility (MOF) – eingeführt. Dieser Standard beschreibt eine spezielle Metadaten-Architektur, die aus vier Ebenen besteht:

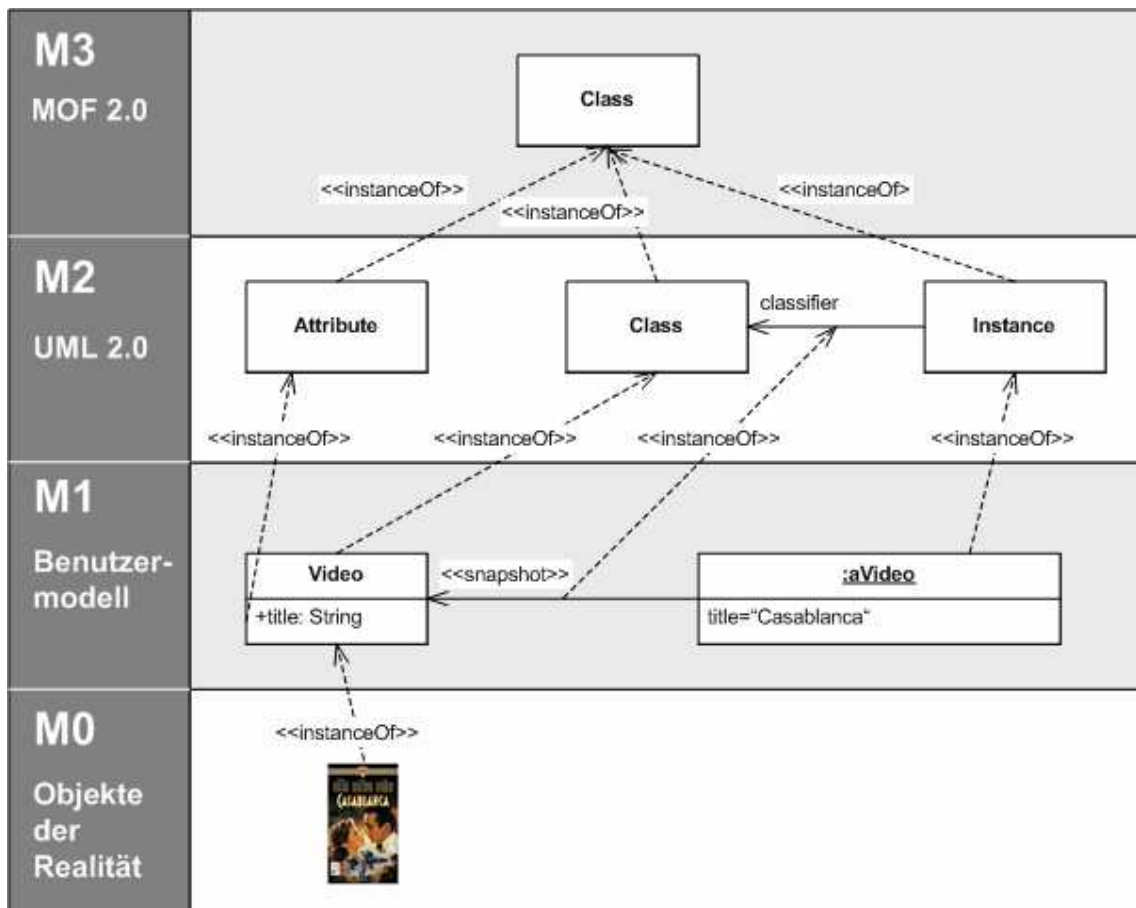


Abbildung 3.1: Hierarchie der Metamodellierung (Quelle [Wik09a])

- M3-Ebene ist die MOF-Ebene. Auf dieser Ebene befinden sich die Meta-Metamodelle, die Aufbau von Metamodellen der M2-Ebene spezifizieren.
- M2-Schicht ist die Metamodell-Ebene. Hier befinden sich Metamodelle, die beschreiben, wie die eigentlichen Modelle strukturiert werden müssen. Die im Rahmen dieser Arbeit ausgearbeiteten Metamodelle sind ebenfalls Instanzen von MOF.
- M1-Ebene. Diese Ebene repräsentiert die konkreten Modelle. In dieser Arbeit sind das die BPMN Modelle der Geschäftsprozesse und die SAP Workflow Modelle.
- M0-Ebene repräsentiert die konkreten Daten.

MOF definiert eine gemeinsame Grundlage für unterschiedliche Metamodelle. Wenn zwei Metamodelle MOF-konform sind, können die von ihnen definierten Modelle ineinander transformiert werden.

In diesem Kapitel werden 2 Metamodelle vorgestellt, die im Rahmen dieser Arbeit erstellt wurden: das Metamodell für den BPMN-Teil, der für die Geschäftsprozess-Modellierung bei SWM verwendet wird (weiter genannt BPMN^(SWM)-Metamodell), und das Metamodell für SAP Workflow, die dann detailliert beschrieben werden. Beide Metamodelle wurden MOF-konform erstellt.

3.2 BPMN-Metamodell für die Geschäftsprozess-Modellierung bei SWM

Das Fehlen eines standardisierten Metamodells wird oft als Nachteil von BPMN gesehen. Es gibt zwar das BPMN Extension im Rahmen des Business Process Definition Metamodels (BPDM) [OMG06], die Modellierungstools greifen allerdings auf ihre eigenen Lösungen zu. Aus diesem Grund wurde im Rahmen dieser Arbeit ein eigenes BPMN^(SWM)-Metamodell ausgearbeitet. Dabei wurden mehrere Faktoren berücksichtigt (so wie z.B. die im Abschnitt 2.2.2 beschriebenen Modell-Einschränkungen, Realisierung der BPMN-Modellierung im Enterprise Architect, die nachfolgende Transformation des Modells auf das SAP Workflow-Modell, etc.) und dementsprechend Entscheidungen getroffen.

- Das unten beschriebene BPMN^(SWM)-Metamodell basiert auf der BPMN 1.0-Spezifikation [BPMN10], da das verwendete Modellierungstool Enterprise Architect in der Version 7.1 die BPMN 1.1 bzw. 1.2 nicht unterstützt. Es wurden allerdings einige Aspekte der BPMN 1.1 Spezifikation [BPMN11] berücksichtigt, um den späteren Umstieg auf die neue Version zu erleichtern (z.B. keine Assoziation vom FlowObject zum Pool oder Auslagerung der Attribute Id, Categories und Documentation von GraphicalObject in BPMNObject).
- Es wurden ausschließlich englische Begriffe der BPMN-Spezifikation verwendet, da es noch keine standardisierte Übersetzung der BPMN-Begriffe ins Deutsche gibt, und auch weil im Unternehmen eine englischsprachige Version vom Enterprise Architect verwendet wird. Einen wertvollen Vorschlag für die Übersetzung kann man aber im Blog vom Prof. Dr. Thomas Allweyer unter [BPMNa] finden.
- Bezüglich der BPMN-Elemente und deren Attribute wurde folgendes berücksichtigt:
 - o Attribute, die Informationen für das Mapping auf die ausführbaren Sprachen (z.B. BPEL) bereitstellen, wurden ausgelassen, da die im Enterprise Architect modellierten Prozesse auf keine ausführbare Sprache gemappt werden und nicht direkt durch eine Prozess-Engine ausgeführt werden (z.B. folgende Attribute des

BPMN-Elementes Process: ProcessType, Status, SuppressJoinFailure, u.a. oder Activity-Attribut LoopCounter).

- o Da AdHoc-Prozesse generell nicht ausführbar sind, wurden die dazugehörigen Attribute ebenfalls ausgelassen.
- o BPMN-Supporting Types, die Enterprise Architect nicht unterstützt oder als Strings speichert, wurden dementsprechend angepasst.
- o Wegen der zukünftigen Transformation des BPMN-Modells ins SAP Workflow-Modell wurde z.B. stricte Trennung der Gateways in Splitting und Merging vorgenommen. Auch Elemente und Attribute, die nicht ins SAP importiert werden oder bei der Modellierung nicht gebraucht werden, wurden ebenfalls ausgelassen (z.B. Activity-Attribute MI_FlowCondition, ComplexMI_FlowCondition).

Struktur der Beschreibung des Metamodells wurde an [OMG07] angelehnt. Zwecks Übersichtlichkeit wird auf die einzelnen Punkte

- Generalisierungen
- Attribute
- Assoziationen
- Constraints
- Notation

sowie auf die ausführliche Semantik nur dann näher eingegangen, falls eine Klasse diese Merkmale besitzt und / oder wenn es Unterschiede zu der BPMN Spezifikation gibt.

3.2.1 BPMN Core

Das Diagramm auf der Abbildung 3.2 definiert abstrakte Metaklassen, die die 2 Haupt-Gruppen von BPMN Elementen sowie die Klasse BusinessProcess behandeln.

BPMNObject

BPMNObject ist der Hauptbestandteil des Modells. BPMNObject ist eine abstrakte Metaklasse ohne Superklasse. Sie wird als allgemeine Superklasse für (fast) alle Klassen in diesem Metamodell verwendet und definiert ein Set an allgemeinen Attributen, die alle BPMN Elemente referenzieren müssen.

Attribute

- Id: String
Eindeutige Kennung eines Elementes, die es von den anderen innerhalb eines Diagramms unterscheidet.

- Name : String
Name des Elementes. Da Enterprise Architect den Namen für alle Elemente vorsieht, wurde dieses Attribut in die Superklasse verlagert.
- Categories: String [0..*]
Ein optionales Attribut, das für das Reporting oder Analyse des Modells verwendet werden kann.
- Documentation: String [0..*]
Optionale textuelle Dokumentation vom BPMNObject.

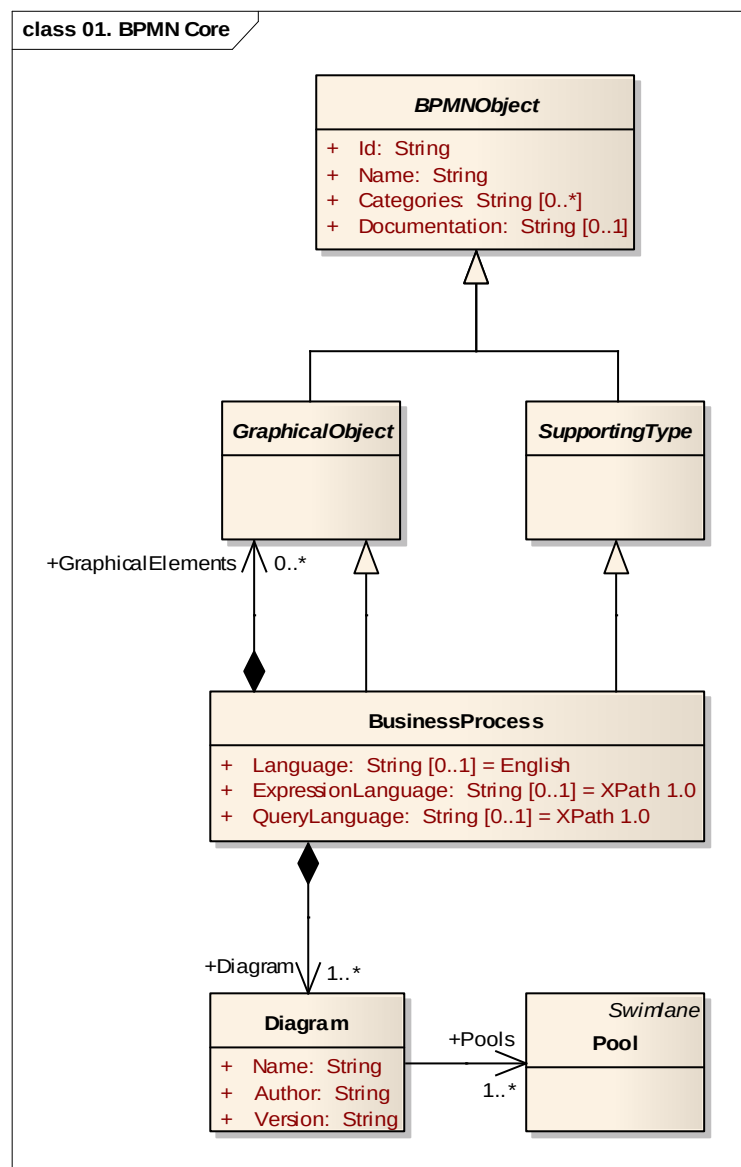


Abbildung 3.2: Diagramm BPMN Core

GraphicalObject

Die abstrakte Metaklasse GraphicalObject repräsentiert ein BPMN Element mit graphischer Darstellung.

Generalisierungen

- BPMNObject auf der Seite 27.

SupportingType

SupportingType definiert Typen, die der Unterstützung der Attribute anderer Elemente dienen.

Generalisierungen

- BPMNObject auf der Seite 27.

Notation

SupportingTypes haben in BPMN keine graphische Notation. EA unterstützt einige von SupportingTypes als UML-Klassen.

BusinessProcess

BusinessProcess repräsentiert den in BPMN modellierten Geschäftsprozess oder das Workflow-Gerüst. Anders als in der BPMN Spezifikation wurden in Enterprise Architect die Elemente Business Process Diagram und Process zu einem zusammengesetzten graphischen Element namens BusinessProcess mit dem dazugehörigen Diagramm zusammengefasst. BusinessProcess hat den größten Teil der Attribute vom Business Process Diagram übernommen. Außerdem können für einen BusinessProcess mehrere Diagramme erzeugt werden.

Generalisierungen

- GraphicalObject auf der Seite 29.
- SupportingType auf der Seite 29.

Attribute

- ExpressionLanguage: String [0..*] = XPath 1.0
Sprache, welche die Syntax der im Modell verwendeten Ausdrücke (Expressions) definiert. Enterprise Architect verwendet als default XPath 1.0.
Obwohl die Ausdrücke (Expressions) in dieser Arbeit als Strings betrachtet werden, wurde das Attribut für die spätere Weiterentwicklung – wenn Expressions bereits in BPMN definiert sind – belassen.

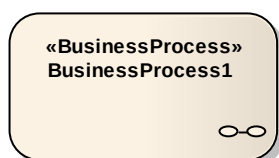
- Language: String [0..*] = English
Definiert die Sprache, in der Texte des Modells geschrieben wurden. Enterprise Architect verwendet als default English.
- QueryLanguage = XPath 1.0
Sprache, welche die Syntax der im Modell verwendeten Queries definiert. Enterprise Architect verwendet als default XPath 1.0.
Obwohl die Queries in dieser Arbeit als Strings betrachtet werden, wurde das Attribut für die spätere Weiterentwicklung – wenn Expressions bereits in BPMN definiert sind – belassen.

Assoziationen

- GraphicalElements: GraphicalObject [0..*] auf der Seite 29
Spezifiziert alle graphischen Elemente, die im BusinessProcess enthalten sind.
- Diagram: Diagram [1..*] auf der Seite 30
Das zum BusinessProcess zugehörige Diagramm.

Notation

BusinessProcess besitzt in Enterprise Architect eine eigene graphische Notation – ein Rechteck mit stark abgerundeten Ecken und einem Zeichen unten rechts, ähnlich dem komplexen Zustand im dynamischen UML Modell.



Diagram

Diagram repräsentiert die graphische Darstellung von BusinessProcess. Anders als das Business Process Diagram in der BPMN-Spezifikation beinhaltet das Diagramm in Enterprise Architect nicht mehrere Prozesse, sondern ist selber ein Bestandteil vom Element BusinessProcess. Der größte Teil der Attribute wurden auch vom BusinessProcess übernommen (Language, ExpressionLanguage, QueryLanguage).

Attribute

- Name: String
Name des Diagramms.
- Author: String
Autor des Diagramms.
- Version: String
Version des Diagramms.

Assoziationen

- Pool: Pool [1..*] auf der Seite 47
Die Pools, die das Diagramm enthält.

3.2.2 BPMN GraphicalObject

Das Diagramm auf der Abbildung 3.3 definiert abstrakte Klassen, welche die 4 Gruppen von BPMN Elementen mit graphischer Darstellung abdecken.

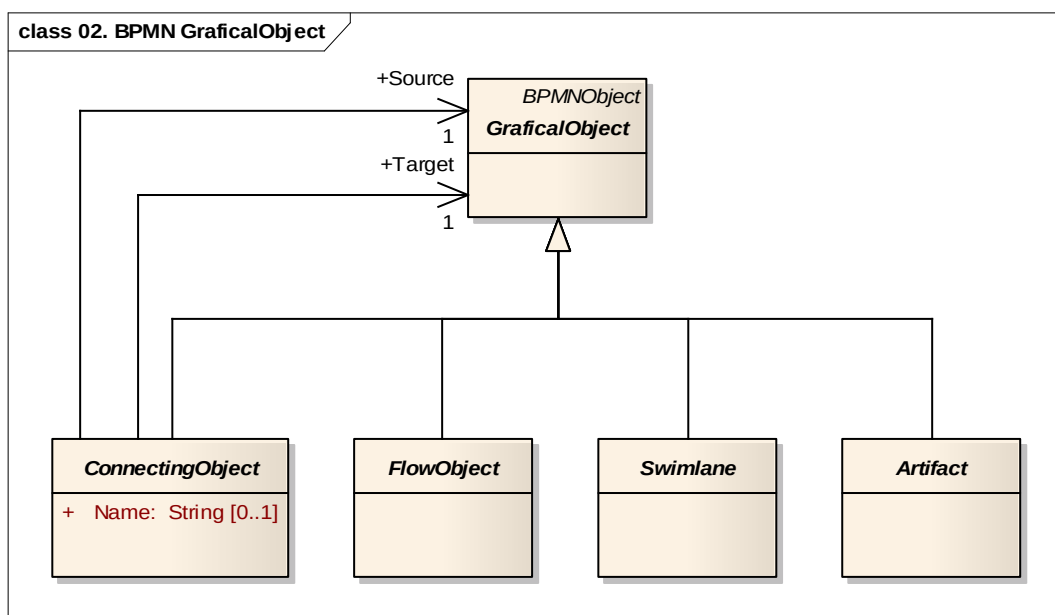


Abbildung 3.3: Diagramm BPMN GraphicalObject

FlowObject

FlowObject repräsentiert die wichtigsten graphischen Elemente, die das Verhalten eines Prozesses beschreiben.

Generalisierungen

- GraphicalObject auf der Seite 29.

ConnectingObject

ConnectingObject definiert graphische Elemente, welche zwei andere graphische Elemente miteinander verbinden und die Reihenfolge der anderen graphischen Elemente im Prozess vorschreiben.

Generalisierungen

- GraphicalObject auf der Seite 29.

Attribute

- Name: String [0..1]
Redefiniert das Attribut Name vom GraphicalObject, so dass der für ConnectingObjects optional ist.

Assoziationen

- Source: GraphicalObject auf der Seite 29
Identifiziert das graphische Element, von welchen das ConnectingObject ausgeht.
- Target: GraphicalObject
Identifiziert das graphische Element, in welches das ConnectingObject hineingeht.

Constraints

- Als Source und Target von einem ConnectingObject können im EA im Unterschied zur BPMN-Spezifikation nicht die anderen ConnectingObjects auftreten.

Swimlane

Swimlane repräsentieren Container, welche graphische Elemente partitionieren und in einem Diagramm organisieren.

Generalisierungen

- GraphicalObject auf der Seite 29.

Notation

Graphische Notation der Swimlanes in Enterprise Architect entspricht nicht der BPMN-Spezifikation; sie stellen ein Rechteck mit einer offenen Seite dar.

Artifact

Abstrakte Klasse Artifact definiert die Elemente, die die zusätzlichen Informationen zum Prozess anbieten.

Generalisierungen

- GraphicalObject auf der Seite 29.

3.2.3 BPMN Activity

Das Diagramm auf der Abbildung 3.4 definiert die Klassen Activity, Task und Sub-Process.

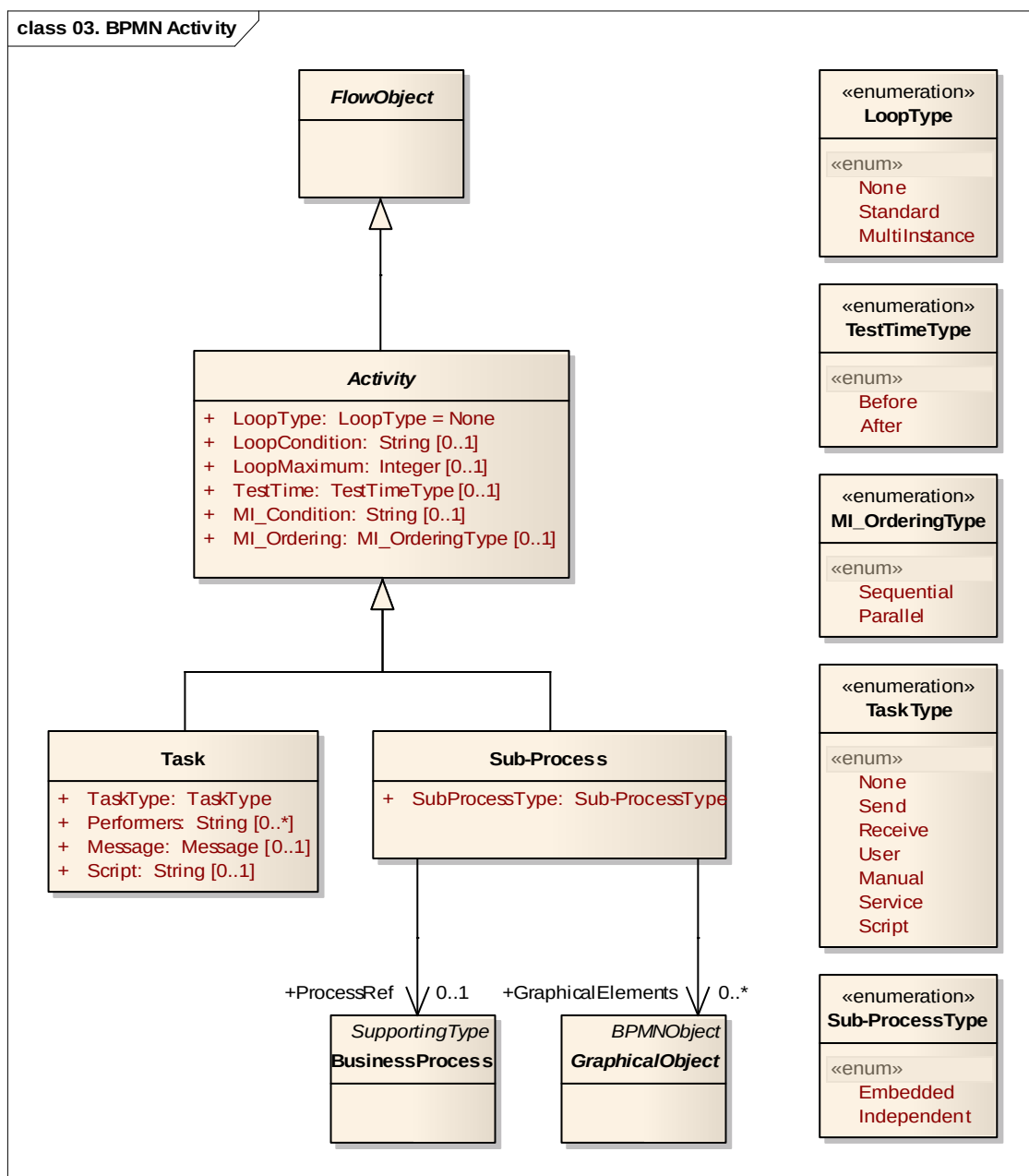


Abbildung 3.4: Diagramm BPMN Activity

Activity

Abstrakte Klasse Activity definiert ein BPMN-Element, welches die eigentlichen Aufgaben symbolisiert, die während des Prozesses ausgeführt werden.

Generalisierungen

- FlowObject auf der Seite 31.

Attribute

- LoopType: LoopType
Gibt an, ob die Activity mehrfach ausgeführt wird oder nicht.

- LoopCondition: String [0..1]
Ausdruck, der bei jedem Durchlauf ausgewertet wird und bestimmt, ob ein Durchlauf der Activity stattfindet.
- LoopMaximum: Integer [0..*]
Optionales Attribut, das die maximale Anzahl an Durchläufen angibt.
- TestTime: TestTimeType [0..*]
Gibt an, ob die LoopCondition vor dem Durchlauf einer Activity oder erst danach ausgewertet wird.
- MI_Condition: String [0..*]
Ausdruck, der zu einem numerischen Wert ausgewertet wird und bestimmt, wie viele Instanzen dieser Activity erzeugt werden.
- MI_Ordering: MI_OrderingType [0..*]
Gibt an, ob die erzeugten Instanzen der Activity sequentiell oder parallel durchlaufen werden.

Constraints

- Attribute LoopCondition, LoopMaximum und TestTime nur bei LoopType = Standard.
- Attribute MI_Condition, MI_Ordering nur bei LoopType = MultiInstance.

LoopType

LoopType ist ein Enumeration Typ, welcher textuelle Werte zur Beschreibung der Art des Looping / Schleifendurchlaufs für eine Activity definiert:

- None – kein Looping.
- Standard – Standardlooping derselben Activity.
- MultiInstance – Erzeugung mehrerer Instanzen dieser Activity.

TestTimeType

TestTimeType ist ein Enumeration Typ, der textuelle Werte zur Angabe definiert, wann die Bedingung zum erneuten Schleifendurchlauf ausgewertet wird.

- Before – vor dem Ausführen der Activity.
- After – nach dem Ausführen der Activity.

MI_OrderingType

MI_OrderingType ist ein Enumeration Type, der textuelle Werte definiert, die zur Beschreibung der Art, wie mehrere Instanzen einer Activity ausgeführt werden, dient:

- Sequential – die einzelnen Instanzen werden hintereinander ausgeführt.
- Parallel – die einzelnen Instanzen werden gleichzeitig ausgeführt.

Task

Die Klasse Task beschreibt das BPMN-Element, welches eine atomare Aufgabe, die nicht weiter verfeinert wird, repräsentiert.

Generalisierungen

- Activity auf der Seite 33

Attribute

- TaskType: TaskType
Typ einer Task.
- Performers: String [0..*]
Optionale Bearbeiter der Aufgabe.
- Message: Message [0..1]
Optionale Nachricht, die eine Task empfängt oder sendet.
- Script: String [0..1]
Optionales Script, welches ausgeführt werden kann (nur für die zukünftige Entwicklung; momentan werden keine Scripts definiert).

Constraints

- Attribut Performers nur bei TaskType = User oder Manual.
- Task vom Typ Receive darf nur LoopType = None haben.
- Attribut Performers darf nur bei den Tasks vom Type Manual und User vorkommen.
- Attribut Message kommt nur bei den Tasks vom Type Send und Receive vor.
- Attribut Script kommt nur bei der Task vom Typ Script vor.

Sub-Process

Die Klasse Sub-Process beschreibt ein BPMN-Element, welches eine zusammengesetzte Aufgabe bedeutet.

Generalisierungen

- Activity auf der Seite 33.

Attribute

- Sub-ProcessType: Sub-ProcessType.

Assoziationen

- ProcessRef: BusinessProcess auf der Seite 29
Identifiziert denjenigen BusinessProcess, welcher den Sub-Process vom Sub-ProcessType Independent referenziert.
- GraphicalElements: GraphicalObject auf der Seite 29
Identifiziert alle Graphischen Elemente, die das Sub-Process vom Sub-ProcessType Embedded beinhaltet.

Constraints

- Assoziation ProcessRef nur bei Sub-ProcessType = Independent.
- Assoziation GraphicalElements nur bei Sub-ProcessType = Embedded.
- Als GraphicalElements können nicht Swimlanes auftreten.

Notation

In Enterprise Architect kann der Sub-Process nur in zugeklappter Form dargestellt werden.

TaskType

TaskType ist ein Enumeration Typ, der textuelle Werte zur Beschreibung des Typs einer Task definiert:

- None – Typ ist nicht näher spezifiziert.
- Send – Task zum Senden einer Nachricht an den anderen Prozessteilnehmer.
- Receive – Task zum Warten auf Eingang einer Nachricht.
- User – Bedeutet eine Aufgabe, die der Benutzer mit Hilfe des Systems erledigt.
- Manual – Aufgabe, die der Benutzer ohne Hilfe des Systems erledigt. Wird in dieser Arbeit auch teilweise „zweckentfremdet“, um eine BenutzerEntscheidung zu definieren, die dann in das System eingetragen wird.
- Service – Ausführung einer automatisierten Applikation.
- Script – Berechnung, die nach der Modelltransformation von SAP ausgeführt werden kann.

Sub-ProcessType

Sub-ProcessType ist ein Enumeration Typ, welcher textuelle Werte zur Beschreibung der Art eines Sub-Process definiert:

- Embedded – der Sub-Process ist von seinem Eltern-Prozess abhängig und kann nur da vorkommen.
- Independent – der Sub-Process ruft einen anderen Prozess auf, der unabhängig von dem aufgerufenen Prozess modelliert ist.

3.2.4 BPMN Event

Das Diagramm in Abbildung 3.5 definiert die Klassen Event, StartEvent, EndEvent, IntermediateEvent und Trigger.

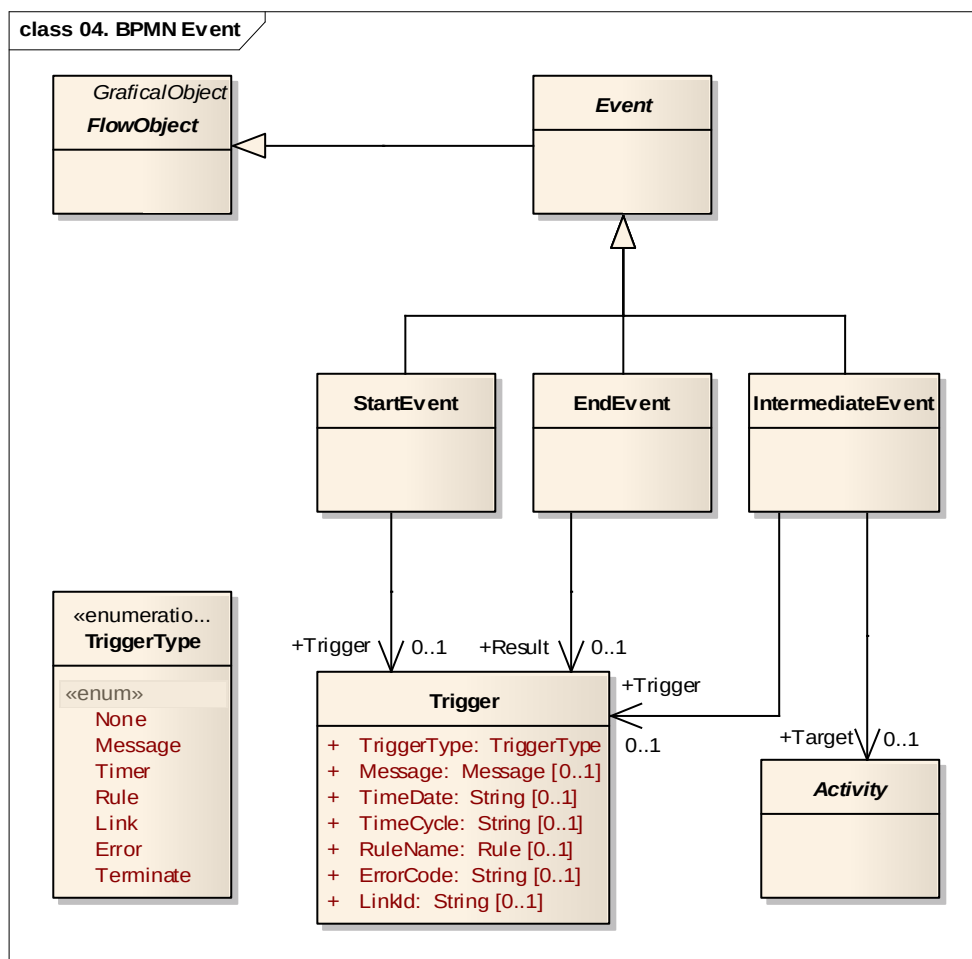


Abbildung 3.5: Diagramm BPMN Event

Event

Die abstrakte Klasse Event definiert ein BPMN-Element, welches bedeutet, dass während der Prozess-Ausführung ein Ereignis stattfindet.

Generalisierungen

- FlowObject auf der Seite 31.

StartEvent

Klasse StartEvent definiert ein BPMN-Element, das den Anfang des Prozesses bezeichnet.

Generalisierungen

- FlowObject auf der Seite 31.

Assoziationen

- Trigger: Trigger [0..1]
Optionale Ursache des Ereignisses.

Constraints

- StartEvent kann nur mit Triggern vom TriggerType None, Message, Timer, Rule und Link assoziiert werden.
- StartEvent kann nicht Target vom SequenceFlow sein.

EndEvent

Die Klasse EndEvent definiert ein BPMN-Element, welches das Ende eines Prozesses bestimmt.

Generalisierungen

- FlowObject auf der Seite 31

Assoziationen

- Result: Trigger [0..1].
Optionales Ergebnis am Ende des Prozesses.

Constraints

- EndEvent kann nur mit den Triggern vom TriggerType None, Message, Error, Link und Terminate assoziiert werden.
- EndEvent kann nicht Source vom SequenceFlow sein.

IntermediateEvent

IntermediateEvent beschreibt ein Ereignis, das während des Prozess-Ablaufes stattfindet.

Generalisierungen

- FlowObject auf der Seite 31.

Assoziationen

- Trigger: Trigger [0..1]
Optionale Ursache des Ereignisses.
- Target: Activity [0..1]
Activity, an deren Rand das IntermediateEvent angekoppelt ist.

Constraints

- IntermediateEvent kann nur mit Triggern vom TriggerType Message, Timer, Error, Rule und Link assoziiert werden.
- IntermediateEvent mit einem Trigger vom TriggerType Message oder Error muss ein Target haben.
- IntermediateEvent mit einem Trigger vom TriggerType Link darf kein Target haben.

Trigger

Die Klasse Trigger definiert eine Ursache des Events oder sein Ergebnis.

Attribute

- TriggerType: TriggerType
Art des Triggers.
- Message: Message [0..1]
Empfangene bzw. gesendete Nachricht.
- TimeDate: String [0..1]
Zeit, wann das Ereigniss ausgelöst wird (einmalig).
- TimeCycle: String [0..1]
Zeit, wann das Ereigniss ausgelöst wird (zyklisch).
- RuleName: Rule [0..1]
Regel, die als wahr oder falsch ausgewertet werden kann.
- ErrorCode: String [0..1]
Fehler, der ausgelöst oder gefangen wird.
- LinkId: String [0..1]

Id zum Verbinden mit dem 2. Event im Paar. Zwei Events mit dem Link Trigger erzeugen in einem BPMN Modell den so genannten virtuellen SequenceFlow. Dieser wird verwendet, wo der normale SequenceFlow nicht verlaufen darf (z.B. zwischen den FlowObjects unterschiedlicher Prozess-Ebenen) oder um den langen SequenceFlow in einer Prozess-Ebene zu vermeiden.

Constraints

- Attribut Message nur beim Trigger mit dem TriggerType = Message.
- Attribute TimeDate und TimeCycle nur beim Trigger mit dem TriggerType = Timer.
- Trigger mit dem TriggerType = Timer darf entweder TimeDate oder TimeCycle haben.
- Attribut RuleName nur beim Trigger mit dem TriggerType = Rule.
- Attribut ErrorCode nur beim Trigger mit dem TriggerType = Error.
- Attribut LinkId nur beim Trigger mit dem TriggerType = Link.

TriggerType

TriggerType ist ein Enumeration Typ, der textuelle Werte zur Beschreibung der Art des Ereignisses oder des Ergebnisses definiert:

- None – kein Ereignis bzw. Ergebnis spezifiziert.
- Message – Ereignis bzw. Ergebnis ist eine Nachricht.
- Timer – zeitgesteuertes Ereignis.
- Error – Ereignis ist ein Fehler.
- Rule – eine Regel wird wahr.
- Link – tritt nur im Paar auf und verbindet virtuell die Beiden Events.
- Terminate – beim Erreichen wird als Ergebnis der Prozess terminiert.

3.2.5 BPMN Gateway

Das Diagramm auf der Abbildung 3.6 definiert die Klassen Gateway, SplittingGateway und MergingGateway.

Gateway

Die abstrakte Klasse Gateway definiert ein BPMN-Element, welches alternative oder parallele Pfade realisiert und die Wegewahl-Logik repräsentiert.

Generalisierungen

- FlowObject auf der Seite 31

Attribute

- GatewayType: GatewayType
Art, wie die Wegewahl im Prozess-Ablauf stattfinden wird.

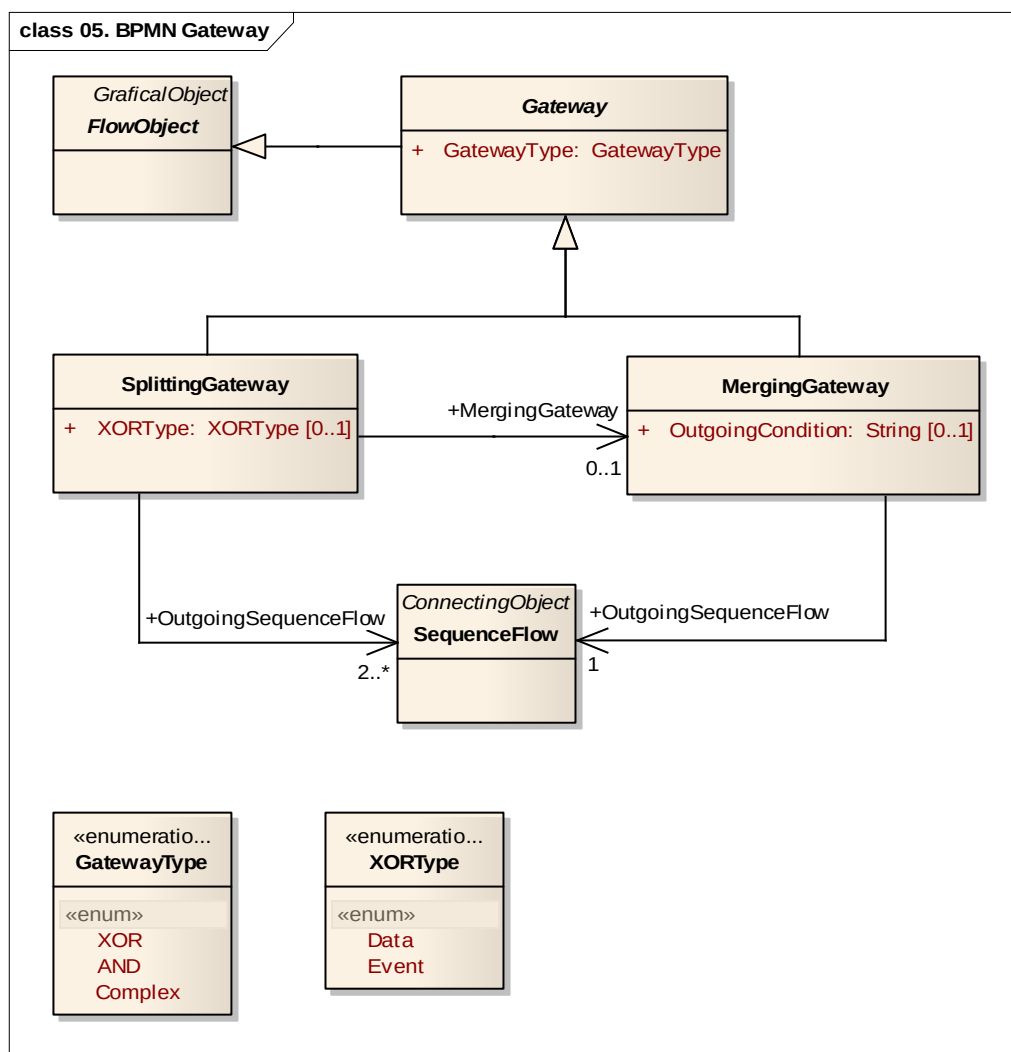


Abbildung 3.6: Diagramm BPMN Gateway

SplittingGateway

SplittingGateway ist ein Gateway, welcher einen eingehenden und mehrere ausgehende SequenceFlows hat und somit mehrere alternative oder parallele Pfade im Prozess erzeugt.

Generalisierungen

- Gateway auf der Seite 40.

Attribute

- XORType: XORType [0..1]
Grundlage für die Wegewahl-Entscheidung.

Assoziationen

- OutgoingSequenceFlow: SequenceFlow [2..*]
Ausgehender SequenceFlow.
- MergingGateway: MergingGateway [0..1]
Ein optionales Attribut. Wird für die Transformation benötigt. Soll auf der 3. Modellierungsebene durch entsprechende eindeutige Namensgebung veranschaulicht werden.

Constraints

- SplittingGateway darf nur einen eingehenden SequenceFlow und muss mindestens 2 ausgehende SequenceFlows haben.
- Als GatewayType dürfen nur XOR oder AND vorkommen.
- Attribut XORType nur bei dem SplittingGateway vom GatewayType XOR.
- SplittingGateway mit GatewayType = XOR und XORType = Data darf als OutgoingSequenceFlow nur SequenceFlow mit dem ConditionType = Expression oder Default haben.
- SplittingGateway mit GatewayType = AND darf als OutgoingSequenceFlow nur SequenceFlow mit dem ConditionType = None haben.

MergingGateway

MergingGateway ist ein Gateway, welcher mehrere eingehende und einen ausgehenden SequenceFlow hat und somit mehrere alternative oder parallele Pfade im Prozess wieder zusammenführt.

Generalisierungen

- Gateway auf der Seite 40.

Attribute

- OutgoingCondition: String [0..1]
Bedingung, die definiert, wie und welche eingehende Pfade zusammengeführt und weitergeleitet werden.

Assoziationen

- OutgoingSequenceFlow: SequenceFlow
Ausgehender SequenceFlow.

Constraints

- MergingGateway darf nur einen ausgehenden SequenceFlow und muss mindestens 2 eingehende SequenceFlows haben.
- Attribut OutgoingCondition gibt es nur beim MergingGateway vom GatewayType Complex.
- MergingGateway darf als OutgoingSequenceFlow nur SequenceFlow mit dem ConditionType = None haben.

GatewayType

GatewayType ist ein Enumeration Typ, der textuelle Werte zur Beschreibung der Art der Pfad-Verzweigung definiert:

- XOR – Fall SplittingGateway: Prozess wird nur an einem Pfad aus mehreren ausgehenden fortgesetzt.
Fall MergingGateway: nur ein Pfad aus mehreren ausgehenden kommt durch.
- AND – Fall SplittingGateway: Prozessauführung wird an allen parallelen ausgehenden Pfaden fortgesetzt.
Fall MergingGateway: es wird mit der Prozessauführung gewartet, bis alle parallel eingehenden Pfade eine Antwort gemeldet haben.
- Complex – mehrere Pfade werden auf Grund einer definierten Bedingung zusammengeführt.

XORType

XORType ist ein Enumeration Typ, der textuelle Werte definiert, die die Art, wie die Wegwahl-Entscheidung getroffen wird, beschreiben:

- Data – die Entscheidung wird auf Grund der ConditionExpression vom OutgoingSequenceFlow getroffen.
- Event – die Entscheidung wird durch ein Ereignis getroffen, welches als erstes eintritt.

3.2.6 BPMN ConnectingObject

Das Diagramm auf der Abbildung 3.7 definiert die Klassen SequenceFlow, MessageFlow und Association.

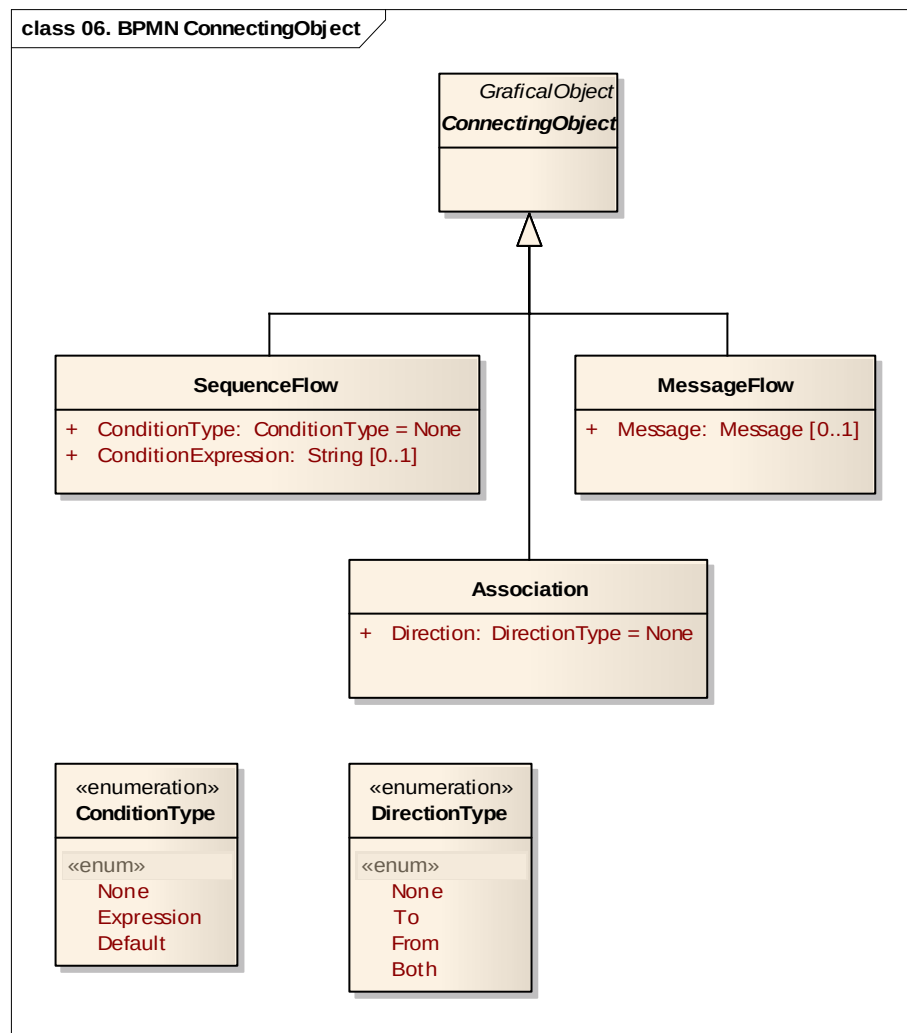


Abbildung 3.7: Diagramm BPMN ConnectingObject

SequenceFlow

Die Klasse SequenceFlow definiert ein BPMN-Element, welches die FlowObjects innerhalb eines Pools verbindet und die Reihenfolge der Aufgaben im Prozess festlegt.

Generalisierungen

- ConnectingObject auf der Seite 31.

Attribute

- ConditionType: ConditionType
Art des SequenceFlows.
- ConditionExpression: String [0..1]
Bedingung, die als wahr oder falsch ausgewertet werden kann.

Constraints

- Attribut ConditionExpression kommt nur beim SequenceFlow vom ConditionType Expression.
- SequenceFlow darf die Pool-Grenzen nicht überschreiten.

MessageFlow

Die Klasse MessageFlow definiert ein BPMN-Element, welches den Nachrichtenaustausch zwischen zwei Prozess-Teilnehmern zeigt.

Generalisierungen

- ConnectingObject auf der Seite 31.

Attribute

- Message: Message [0..1]
Optionale Nachricht, die an den anderen Prozess-Teilnehmer geschickt wird.

Constraints

- MessageFlow darf nur die FlofObjects in einem Pool mit dem Rand eines anderen Pools verbinden.

Association

Die Klasse Association definiert ein BPMN-Element, welches die Artifacts mit den anderen GraphicalObjects verbindet.

Generalisierungen

- ConnectingObject auf der Seite 31.

Attribute

- Direction: DirectionType
Richtung der Association.

Constraints

- Entweder Source oder Target muß Artifact sein.

ConditionType

ConditionType ist ein Enumeration Typ, der textuelle Werte definiert, die beschreiben, ob während des Prozess-Ablaufs eine Auswertung bei Auswahl des SequenceFlows stattfindet:

- None – es findet keine Auswertung statt.
- Expression – es wird eine mit dem SequenceFlow assoziierte Bedingung (ConditionExpression) ausgewertet. Der SequenceFlow wird nur dann gewählt, wenn diese ConditionExpression als wahr ausgewertet wird.
- Default – wird ausgewählt, wenn keine ConditionExpression von SequenceFlows, die aus demselben Gateway ausgehen, als wahr ausgewertet wird.

DirectionType

DirectionType ein Enumeration Typ, der textuelle Werte definiert, die die Richtung einer Association beschreiben:

- None – nicht gerichtete Association.
- To – zum FlowObject gerichtete Association.
- From – zum Artifact gerichtete Association.
- Both – beidseitig gerichtete Association.

3.2.7 BPMN Swimlane

Das Diagramm auf der Abbildung 3.8 definiert Klasse Pool und Lane.

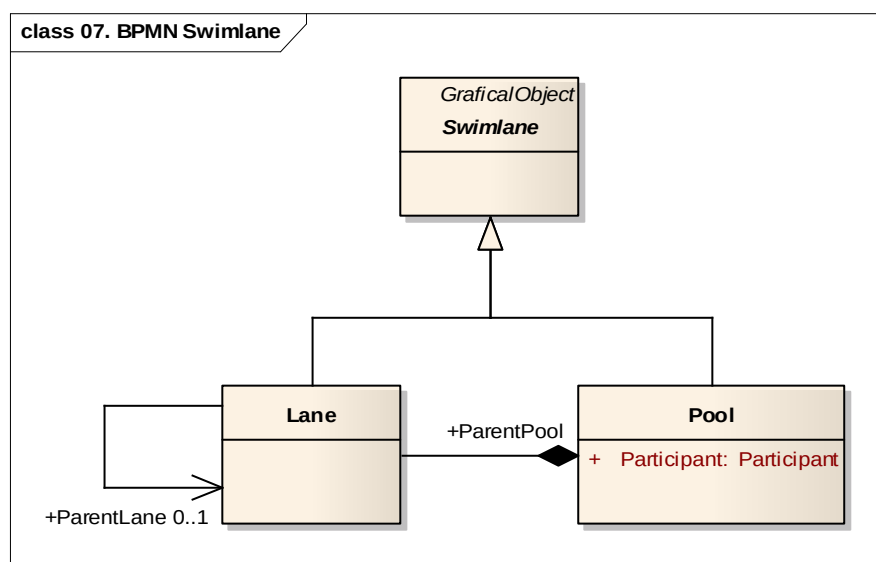


Abbildung 3.8: Diagramm BPMN Swimlane

Pool

Die Klasse Pool definiert ein BPMN-Element, welches einen Teilnehmer im Prozess bedeutet, wie z.B. eine abstrakte Business-Rolle, eine konkrete Organisation auf den Prozess-Ebenen oder das SAP System auf der Workflow-Ebene. Ein Pool kann einen Prozess enthalten. In diesem Fall tritt er auch als Container für diesen Geschäftsprozess bzw. das Workflow-Gerüst auf.

Generalisierungen

- Swimlane auf der Seite 32.

Attribute

- Participant: Participant
Teilnehmer des in dem Pool modellierten Prozesses.

Lane

Die Klasse Lane definiert ein BPMN-Element, welches einen Pool in Sub-Partitionen unterteilt und die Aufgaben im Prozess kategorisiert.

Generalisierungen

- Swimlane auf der Seite 32.

Assoziationen

- ParentPool: Pool
Der übergeordnete Pool.
- ParentLane: Lane [0..1]
Die übergeordnete Lane.

3.2.8 BPMN Artifact

Das Diagramm auf der Abbildung 3.9 definiert die Klassen DataObject und TextAnnotation.

DataObject

Die Klasse DataObject definiert ein BPMN-Element, welches keinen unmittelbaren Einfluss auf SequenceFlow oder MessageFlow des Prozesses hat, aber wichtige Informationen darüber besitzt, was eine Aufgabe für ihre Ausführung benötigt oder was sie nach ihrer Ausführung produziert.

Generalisierungen

- Artifact auf der Seite 32.

Attribute

- State: String [0..1]
Optionaler Zustand vom DataObject.

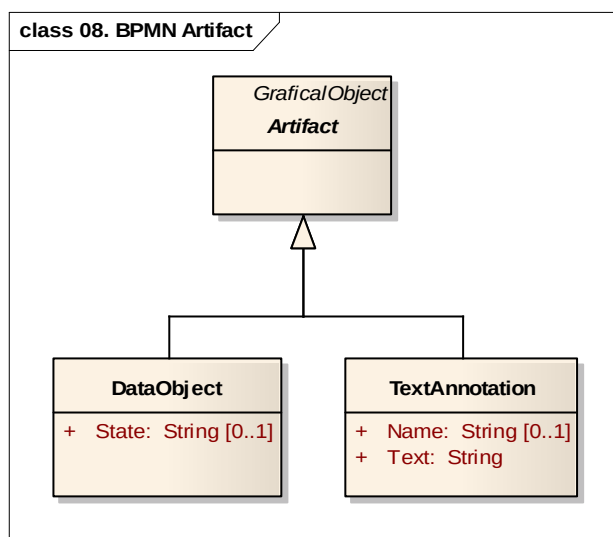


Abbildung 3.9: Diagramm BPMN Artifact

TextAnnotation

Die Klasse TextAnnotation definiert ein BPMN-Element, in dem der Modellierer Zusatzinformationen für die Leser des Diagrams eintragen kann.

Generalisierungen

- Artifact auf der Seite 32.

Attribute

- Name: String [0..1]
Redefiniert das Attribut Name vom GraphicalObject, so dass der für die TextAnnotation optional ist.
- Text: String
Eigentlicher Text.

3.2.9 BPMN SupportingType

Das Diagramm auf der Abbildung 3.10 definiert die Klassen Participant, Rule und Message.

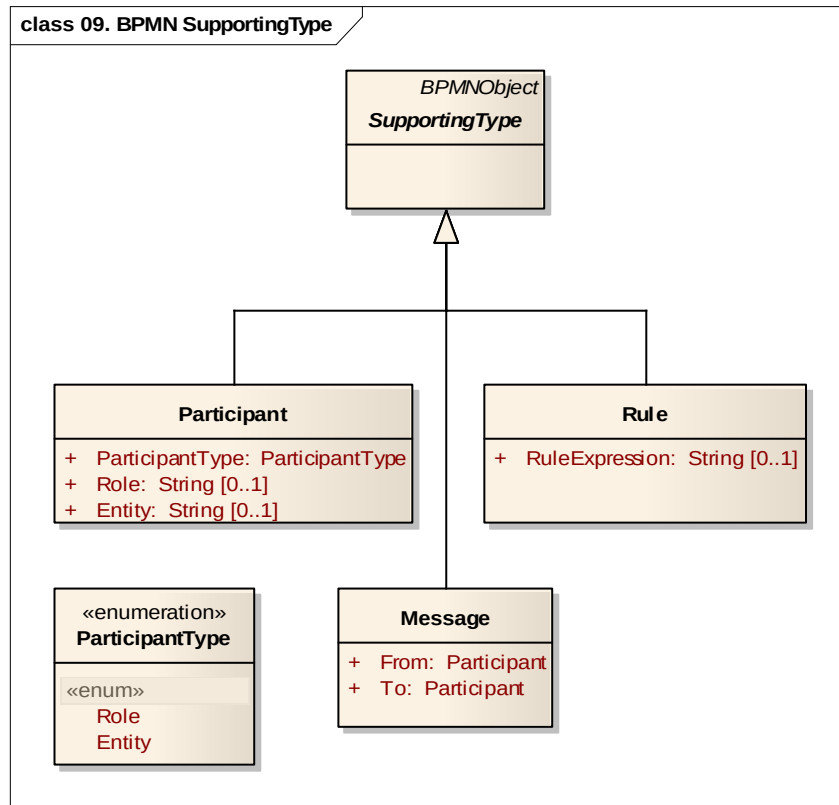


Abbildung 3.10: Diagramm BPMN SupportingType

Participant

Ein Participant repräsentiert einen Teilnehmer in einem Prozess. Das kann ein Unternehmen, eine Geschäftsrolle, eine Einzelperson oder eine Personengruppe sein.

Generalisierungen

- SupportingType auf der Seite 29.

Attribute

- ParticipantType: ParticipantType.
- Role: String [0..1]
Angabe der Teilnehmer-Rolle.
- Entity: String [0..1]
Angabe der Entität des Teilnehmers.

Constraints

- Participant hat entweder das Attribut Role, wenn ParticipantType = Role ist oder das Attribut Entity wenn ParticipantType = Entity ist.

Rule

Rule repräsentiert eine Regel, die erfüllt werden kann.

Generalisierungen

- SupportingType auf der Seite 29.

Attribute

- RuleExpression: String [0..1]
Eine vordefinierte Regel.

Message

Message definiert eine Nachricht, die zwischen Prozess-Teilnehmern gesendet wird.

Generalisierungen

- SupportingType auf der Seite 29.

Attribute

- From: Participant
Sender der Nachricht.
- To: Participant
Empfänger der Nachricht.

ParticipantType

ParticipantType ist ein Enumeration Typ, der textuelle Werte für den Typ eines Prozess-Teilnehmers definiert:

- Role – Teilnehmer ist eine Rolle.
- Entity – Teilnehmer ist eine Entität oder Person.

3.3 Metamodell für den SAP Business Workflow

Genau wie bei BPMN gibt es kein offizielles Metamodell für den SAP Business Workflow (im Weiteren als SAP Workflow benannt) bzw. es wurde nicht veröffentlicht. Aus diesem Grund wurde im Rahmen dieser Arbeit auf Basis der SAP Dokumentation, der Online-Hilfe zu SAP, persönlicher Arbeit mit dem System sowie Zusammenarbeit mit den SWM-Mitarbeitern das Metamodell für den SAP Workflow erstellt. Dabei wurde zum einen versucht, dem Leser ein Verständnis über das System zu vermitteln, aber gleichzeitig auch, sich auf das Nötige zu beschränken. Dieses Metamodell umfasst also nicht die volle Funktionalität des Systems.

Im Gegensatz zum BPMN^(SWM)-Metamodell wurden hier deutsche Begriffe verwendet, da bei SWM Deutsch als SAP-Anwendungssprache verwendet wird.

3.3.1 Sap Core

Abbildung 3.11 beschreibt Klassen, welche die interne SAP-Logik definieren. Diese Logik kann man von außen nicht beeinflussen oder manipulieren. Deswegen wird in der Beschreibung der Klassen auf nicht relevante Elemente und Attribute verzichtet.

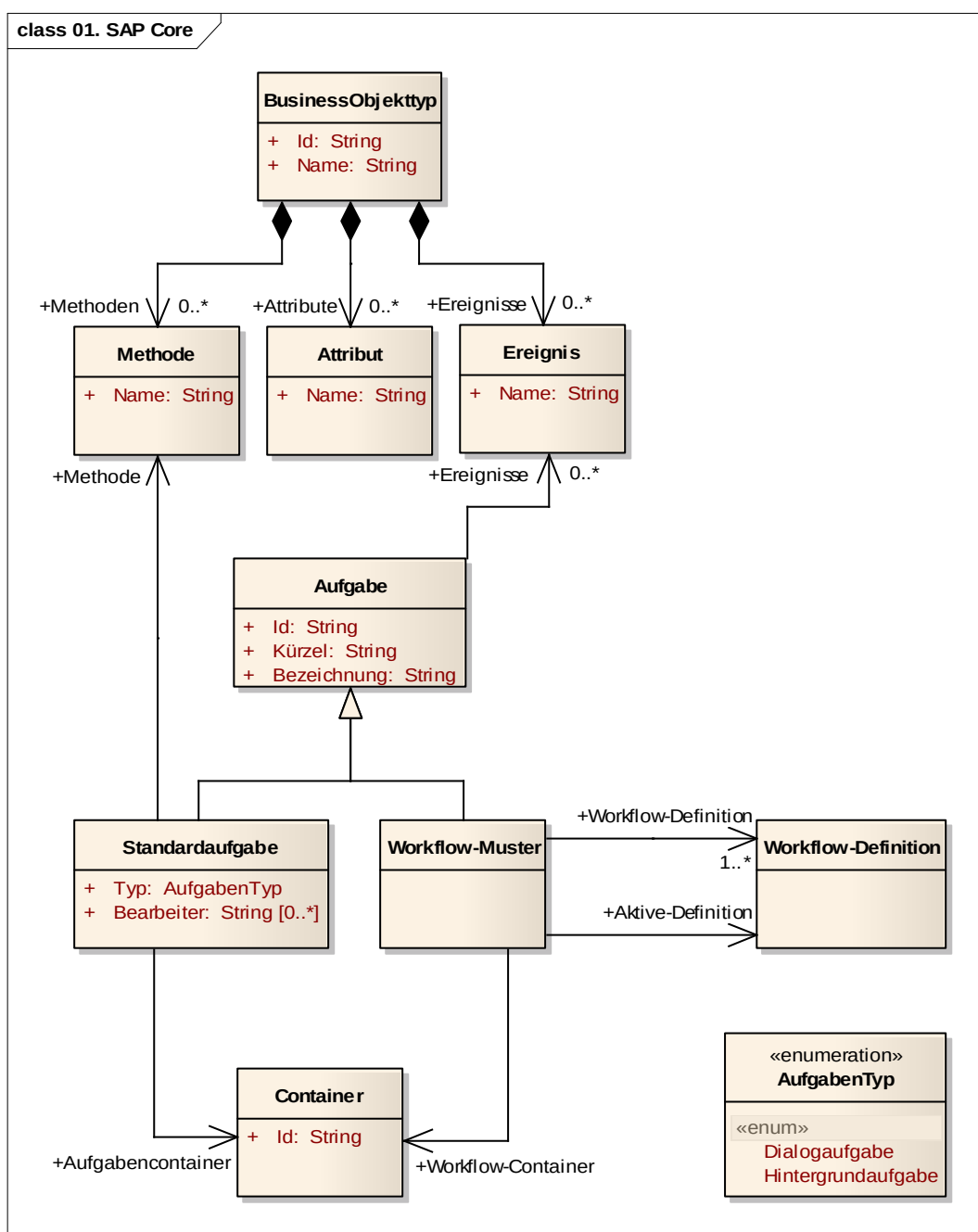


Abbildung 3.11: Diagramm SAP Core

BusinessObjekttyp

BusinessObjekttyp definiert Objektbeschreibungen, die im System verwendet werden. Zur Laufzeit entstehen im System die konkreten Objekte eines zuvor definierten Objekttyps. Man kann es mit der Klasse und deren Instanzen in einem UML-Diagramm vergleichen.

Attribute

- Id: String
Eindeutige Kennung eines BusinessObjekttyps.
- Name: String
Name des BusinessObjekttyps.

Assoziationen

- Methoden:
Operation, die auf einem Objekt ausgeführt werden kann.
- Attribute:
Eigenschaft eines Objektes
- Ereignisse:
Zustandsänderungen eines Objektes.

Methode

Operation, die auf einem Objekt ausgeführt werden kann. Methoden beziehen sich in der Regel auf vorhandene ABAP-Funktionen wie z.B. Transaktionen, Funktions- oder Dialogbausteine. Sie werden meistens über ihren Namen und ihre Parameter aufgerufen. Nach außen ist nicht sichtbar, wie Methoden tatsächlich implementiert sind [SAPBF].

Attribute

- Name: String
Name der Methode.

Attribut

Eigenschaft oder Merkmal eines Objekts. Mit Attributen können bei der Workflow-Definition Bedingungen formuliert werden. Zur Laufzeit werden die Attributwerte gelesen oder berechnet und können auch zur Steuerung des Workflows benutzt werden [SAPBF].

Attribute

- Name: String
Name des Attributes.

Ereignis

Auftreten einer Zustandsänderung eines Objektes, das systemweit publiziert wird. Als Reaktion auf ein Ereignis können Workflows und Aufgaben gestartet, beendet oder fortgesetzt werden [SAPBF].

Attribute

- Name: String
Name des Ereignisses.

Aufgabe

Aufgabe beschreibt eine elementare betriebswirtschaftliche Tätigkeit. Zur Laufzeit wird eine Instanz der Aufgabe (Workitem) erzeugt.

Attribute

- Id: String
Eindeutige Kennung der Aufgabe; vom System vergeben.
- Kürzel: String
Eindeutige Kennung der Aufgabe; vom Anleger vergeben; maximal 12 Zeichen lang.
- Bezeichnung: String
Name der Aufgabe.

Assoziationen

- Ereignis: Ereignis
Ereignis, das die Aufgabe starten oder beenden kann.

Standardaufgabe

Standardaufgabe ist eine Einzelschrittaufgabe.

Generalisierungen

- Aufgabe auf der Seite 53.

Attribute

- Typ: AufgabenTyp
Typ der Standardaufgabe.
- Bearbeiter: String [0..*]
Optionalen Bearbeiter der Aufgabe.

Assoziationen

- Methode: Methode
Operation, auf die sich die Standardaufgabe bezieht.
- Aufgabencontainer: Container
Container einer Aufgabe zur Aufbewahrung von Daten aus dem Umfeld der Aufgabe [SAPBF]. Aufgabencontainer ist mit einer MethodenvARIABLE im Java-Programm vergleichbar.

Constraints

Attribut Bearbeiter hat nur Standardaufgabe vom Aufgabentyp Dialogaufgabe.

Workflow-Muster

Workflow-Muster ist eine Aufgabe, die aus mehreren Schritten besteht.

Generalisierungen

- Aufgabe auf der Seite 53.

Assoziationen

- Workflow-Definition: Workflow-Definition [1..*]
Technische Beschreibung eines Workflow.
- Aktive-Definition: Workflow-Definition
Aktuell verwendbare Workflow-Definition.
- Workflow-Container: Container
Container einer Aufgabe zur Aufbewahrung von workflow-spezifischen Systemelementen. Workflow-Container ist mit einer Klassenvariable im Java-Programm vergleichbar.

Container

Grundlegende Datenstruktur der unterschiedlichen Definitions- und Ausführungskomponenten des SAP Business Workflows. Container enthalten Containerelemente zur Aufnahme von

- Werten (Konstanten)
- Strukturen
- Objektreferenzen

Die Containerelemente können zu Kontroll- und Steuerungszwecken bei der Ausführung des Workflows und von Workitems genutzt werden [SAPBF].

Attribute

- Id: String
Eindeutige Kennung des Containers.

AufgabenTyp

AufgabenTyp ist ein Enumeration Typ, der textuelle Werte für den Typ einer Standardaufgabe definiert:

- Dialogaufgabe – Aufgabe, die zur Ausführung Benutzereingaben benötigt.
- Hintergrundaufgabe – Aufgabe, die automatisch ausgeführt wird.

3.3.2 SAP Workflow-Definition

Das Diagramm auf der Abbildung 3.12 beschreibt Klassen, welche die grundlegenden Bestandteile einer Workflow-Definition festlegen.

Workflow-Definition

Workflow-Definition repräsentiert die technische Beschreibung eines Workflows und besitzt eine graphische Notation.

Attribute

- Version: String
Eindeutige Id, die Workflow-Definitionen eines Workflow-Musters voneinander unterscheidet.

Assoziationen

- Block: Block
Das elementare Aufbauelement einer Workflow-Definition.
- LokalesEreignis: LokalesEreignis
Lokales Ereignis einer konkreten Workflow-Definition.

LokalesEreignis

LokalesEreignis ist ein vom Workflow-Muster unabhängiges Ereignis, das nur in der konkreten Workflow-Definition definiert ist.

Attribute

- Name: String

Name des lokalen Ergebnisses.

- Typ: EreignisTyp
Typ vom Lokalen Ereignis.

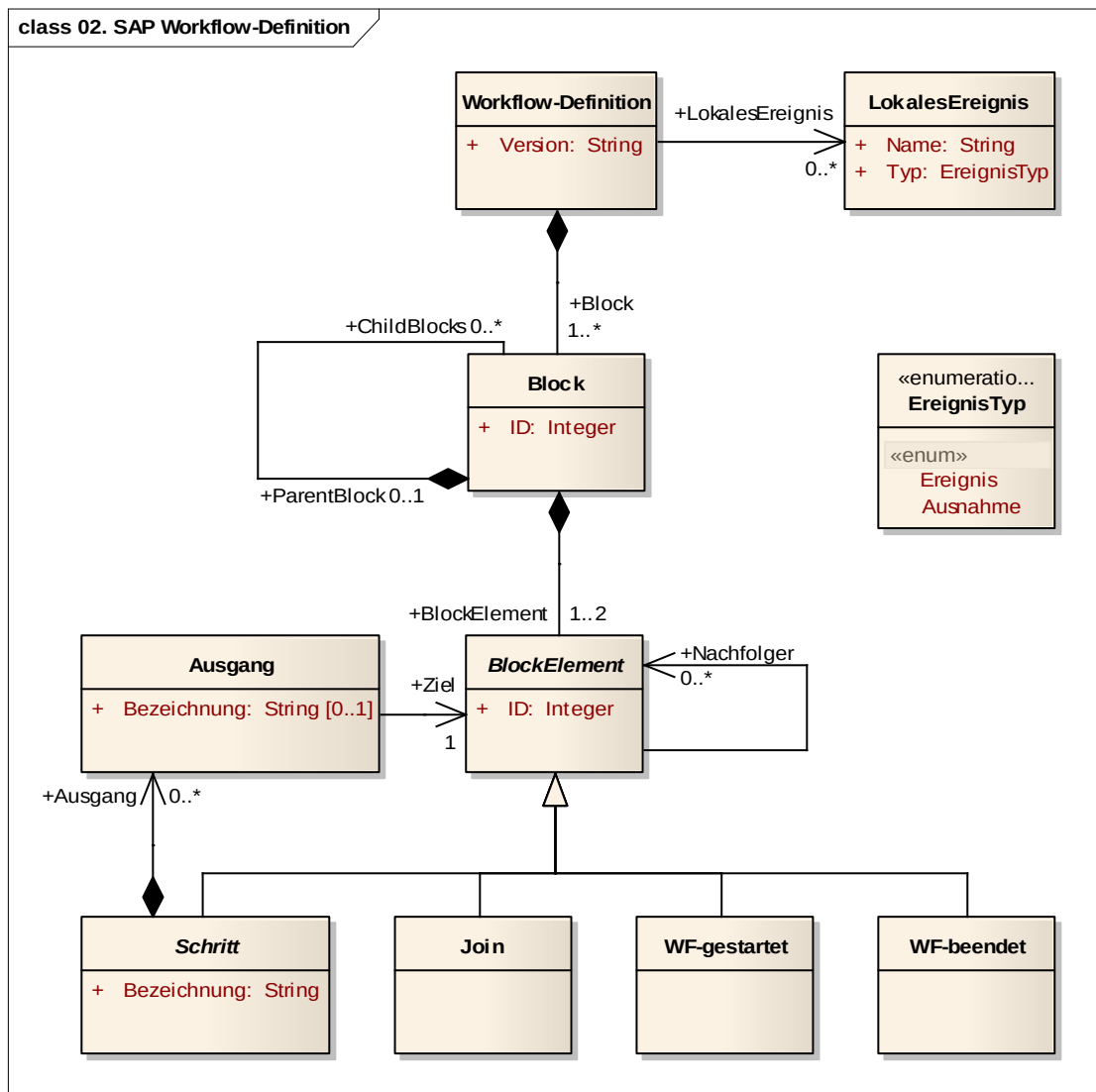


Abbildung 3.12: Diagramm SAP Workflow-Definition

EreignisTyp

EreignisTyp ist ein Enumeration Typ, der textuelle Werte für den Typ des lokalen Ereignisses definiert:

- Ereignis – Nachricht über eine Änderung im Workflow-Ablauf.
- Ausnahme – Nachricht über einen Fehler im Workflow-Ablauf.

Block

Block ist das elementare Strukturelement einer Workflow-Definition mit konstanter Anordnung seiner Bestandteile. Blöcke können sequentiell hintereinander oder ineinander verschachtelt sein, aber niemals die Grenzen voneinander überschneiden. Block hat immer einen Anfang und ein Ende (siehe auch Abschnitt 2.3 Workflow-Modellierung mit SAP).

Attribute

- ID: Integer
Eine eindeutige numerische Bezeichnung eines Blocks, die ihn von anderen Blocks innerhalb einer Workflow-Definition unterscheidet.

Assoziationen

- ParentBlock: Block [0..1]
Ein optionaler übergeordneter Block.
- ChildBlocks: Block [0..*]
In dem Block verschachtelte Unterblöcke.
- BlockElement: BlockElement [1..2]
Ein Bestandteil des Blocks.

Constraints

- Nur der äußerste Block einer Workflow-Definition hat keinen ParentBlock und besteht aus den BlockElementen WF-gestartet und WF-beendet.

BlockElement

Attribute

- ID: Integer
Eine eindeutige numerische Bezeichnung

Assoziationen

- Nachfolger: BlockElement [0..*]
Nachfolgende Elemente in der Workflow-Definition.

Constraints

- Nur das BlockElement WF-beendet hat keinen Nachfolger.

Schritt

Schritt ist der wichtigste Baustein einer Workflow-Definition mit einer graphischen Notation. Kann innerhalb von seinem Block mehrere alternative oder parallele Pfade erzeugen.

Generalisierungen

- BlockElement auf der Seite 57.

Attribute

- Bezeichnung: String
Name des Schrittes.

Assoziationen

- Ausgang: Ausgang [0..*]
Ergebnis eines Schrittes.

Constraints

- Falls der Schritt Ausgänge besitzt, muß die Anzahl der Ausgänge gleich der Anzahl der Nachfolger sein.

Bemerkung

Falls der Schritt einen Ausgang hat, wird manchmal in der Beschreibung der Transformationsregeln zwecks Übersichtlichkeit entweder der Nachfolger vom Schritt oder das Ziel vom Ausgang ausgelassen.

Ausgang

Ausgang beschreibt mögliches Ergebnis eines Schrittes.

Attribute

- Bezeichnung: String
Optionaler Name des Ausgangs.

Assoziationen

- Ziel: BlockElement
Nachfolger des Schrittes nach dem vom Ausgang definierten Ergebnis.

Join

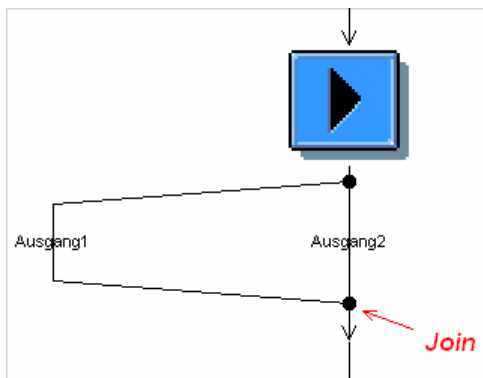
Definiert meistens das Ende des Blocks und führt alle vom Schritt erzeugten Pfade wieder zusammen. In diesem Metamodell wird der Join nur dann verwendet, falls die Blockarchitektur dies benötigt, oder falls der Schritt mehr als einen Ausgang hat.

Generalisierungen

- BlockElement auf der Seite 57.

Notation

Join wird nur dann grafisch dargestellt, wenn der Schritt mehrere Nachfolger hat, d.h. mehrere Pfade erzeugt hat, die zusammengeführt werden müssen, und zwar als kleiner schwarzer Punkt. Es gibt allerdings Schritte, die auch für Join besondere grafische Darstellung besitzen; dies wird bei der Beschreibung der jeweiligen Schritte erwähnt.



WF-gestartet

Definiert den Start einer Workflow-Definition und wird bei ihrem Anlegen standardmäßig erzeugt.

Generalisierungen

- BlockElement auf der Seite 57.

Constraints

- Kein BlockElement darf ein WF-Gestartet als Nachfolger haben.

Notation



WF-beendet

Definiert das Ende einer Workflow-Definition. Wird ebenfalls beim Anlegen einer neuen Workflow-Definition miterzeugt.

Generalisierungen

- BlockElement auf der Seite 57.

Constraints

- Hat keinen Nachfolger.

Notation

Die gleiche wie bei WF-gestartet:



3.3.3 SAP Schritt

Das Diagramm auf der Abbildung 3.13 definiert die Klassen Aktivität, Steuerungsschritt, UnbestimmterSchritt und Terminüberwachung.

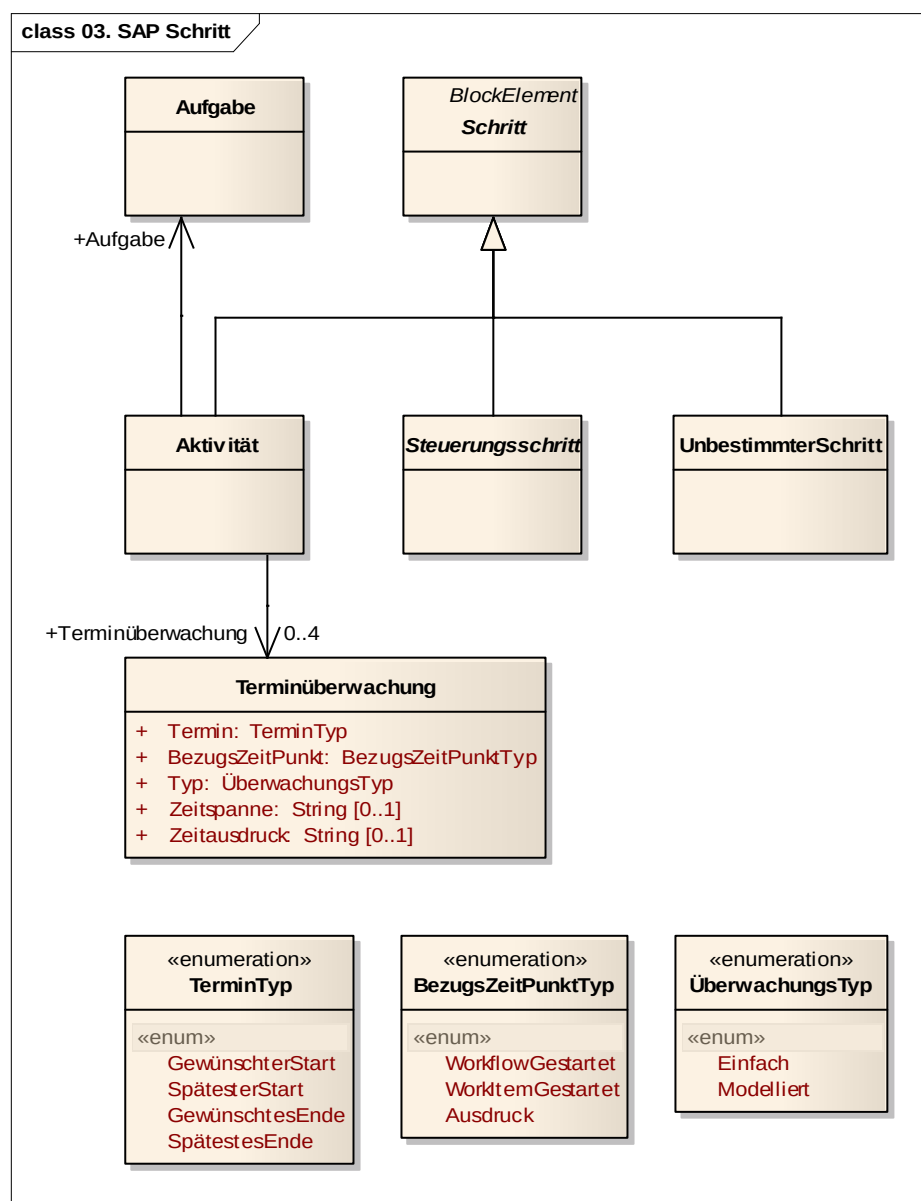


Abbildung 3.13: Diagramm SAP Schritt

Aktivität

Aktivität ist ein Schritttyp in der Workflow-Definition, der eine Aufgabe referenziert, die zur Laufzeit ausgeführt wird. Als Aufgabe kann sowohl eine Standardaufgabe als auch ein Workflow-Muster vorkommen.

Generalisierungen

- Schritt auf der Seite 58.

Assoziationen

- Aufgabe: Aufgabe
Aufgabe, die zur Laufzeit ausgeführt wird.
- Terminüberwachung: Terminüberwachung [0..4]
Überwachung der Start- und Endtermine einer Aufgabe zur Laufzeit. Die Start- und Endtermine können jeweils als gewünscht oder endgültig angegeben werden.

Notation

- Aktivität, die eine Standardaufgabe referenziert.



- Aktivität, die ein Workflow-Muster referenziert.



Steuerungsschritt

Die abstrakte Klasse Steuerungsschritt definiert Schritttypen, die den Workflow-Ablauf steuern.

Generalisierungen

- Schritt auf der Seite 58.

UnbestimmterSchritt

Schritt ohne Funktionalität. Dient als Platzhalter in der Workflow-Definition und kann durch jeden anderen Schritttyp ersetzt werden.

Generalisierungen

- Schritt auf der Seite 58.

Constraints

- UnbestimmterSchritt hat einen Ausgang.

Notation



Terminüberwachung

Überwachung der Start- und Endtermine einer Aufgabe zur Laufzeit. Zu einer Aktivität hinzugefügte Terminüberwachung vom Überwachungstyp = Modelliert erzeugt bei ihr einen zusätzlichen Ausgang. Dieser Pfad wird allerdings nicht am Ende des Blocks mit den anderen zusammengeführt, sondern endet in einem WF-beendet ähnlichem BlockElement, welches quasi Ereignis dieser Aktivität darstellen soll. Da dieses Element aber kein eigenes Block erzeugt, wurde es im Rahmen dieser Arbeit auch nicht als vollwertiges Element der Workflow-Notation betrachtet. Dieser Sachverhalt wird in diesem Metamodell nicht näher spezifiziert aber im Kapitel 4 bei der Transformation vom IntermediateEvent ausführlicher beschrieben (siehe dazu auch Abbildung 4.12 bis Abbildung 4.15).

Attribute

- Termin: TerminTyp
Art des Termins definiert, der überwacht werden soll.
- BezugsZeitPunkt: BezugsZeitPunktTyp
Zeitpunkt, auf den sich die Terminüberwachung bezieht.
- Typ: Überwachungstyp
Handlung, die beim Terminüberschreiten stattfindet.
- Zeitspanne: String [0..1]
Zeit, die zwischen dem Bezugszeitpunkt und der Überschreitung der Frist verstreicht (Zeitausdruck).
- Zeitausdruck: String [0..1]
Ausdruck, der die Zeit zur Terminüberschreitung angibt.

Constraints

- TerminTyp = GewünschterStart kann nur mit dem Überwachungstyp = Einfach kombiniert werden.
- Beim BezugsZeitPunktTyp = WorkflowGestartet und WorkitemGestartet ist Zeitspanne notwendig.
- Beim BezugsZeitPunktTyp = Ausdruck ist Zeitausdruck notwendig.

TerminTyp

TerminTyp ist ein Enumeration Typ, der textuelle Werte zur Beschreibung der Art des Termins definiert, der überwacht werden soll:

- GewünschterStart – Zeit, wann die Aufgabe starten soll.
- SpätesterStart – Zeit, bis wann die Aufgabe starten muß.
- GewünschtesEnde – Zeit, wann die Aufgabe beendet werden soll.
- SpätestesEnde – Zeit, wann die Aufgabe beendet werden muss.

BezugsZeitPunktTyp

BezugsZeitPunktTyp ist ein Enumeration Typ, der textuelle Werte definiert, die den Zeitpunkt beschreiben, wann die Terminüberwachung starten soll:

- WorkflowGestartet – Startzeit des Workflow.
- WorkitemGestartet – Startzeit der Aufgabe.
- Ausdruck – Zeitausdruck.

ÜberwachungsTyp

ÜberwachungsTyp ist ein Enumeration Typ, der definiert was passiert, falls der Termin überschritten wird:

- Einfach – der eingetragene Zuständige wird benachrichtigt; Aufgabe wird nicht unterbrochen.
- Modelliert – laufende Aufgabe wird abgebrochen.

3.3.4 SAP Aktivitaet

Das Diagramm auf der Abbildung 3.14 definiert alle möglichen Aktivitätsschritttypen in der Workflow-Definition.

Benutzerentscheidung

Benutzerentscheidung ist ein Schritttyp, bei dem der Ablauf des aktuellen Workflows zur Laufzeit auf Grund einer Entscheidung des tatsächlichen Bearbeiters gesteuert wird.

Generalisierungen

- Aktivität auf der Seite 61.

Attribute

- Titel: String
Titel der Entscheidung, die der Bearbeiter treffen soll.
- Alternative: String [2..*]
Alternativen, aus welchen der Bearbeiter eine auswählt .

Notation

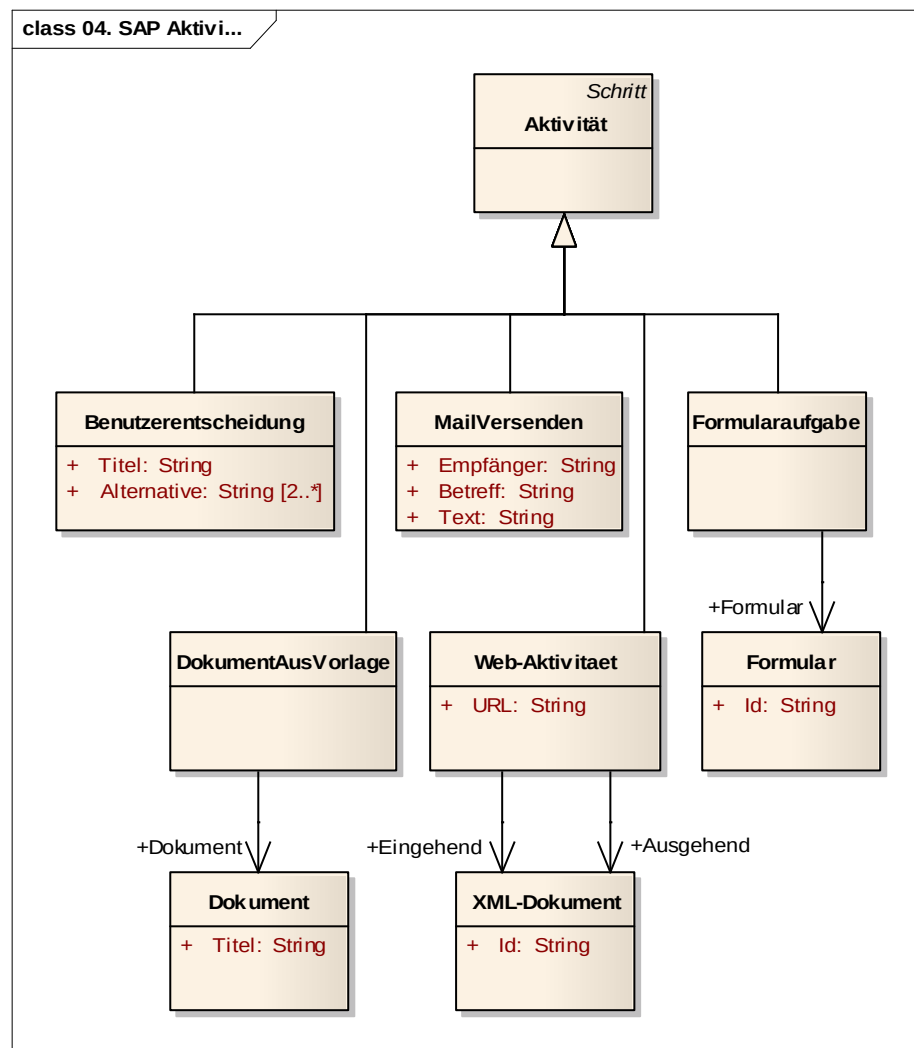


Abbildung 3.14: Diagramm SAP Aktivität

MailVersenden

In diesem Schritt wird eine Email an einen Empfänger versendet.

Generalisierungen

- Aktivität auf der Seite 61.

Attribute

- Empfänger: String
Empfänger der Email.
- Betreff: String
Betreff der Email.
- Text: String
Text der Email.

Constraints

- Dieser Schritt hat keine Terminüberwachung.

Notation



Formularaufgabe

In der Formularaufgabe kann man Daten eines Workflow-Containers anzeigen oder bearbeiten.

Generalisierungen

- Aktivität auf der Seite 61.

Assoziationen

- Formular: Formular
Das Formular, das erstellt wird.

Notation



Formular

Struktur zum Anzeigen der Daten eines Workflow-Containers.

Attribute

- ID: String
Eine eindeutige Bezeichnung eines Formulars.

Web-Aktivität

Im Schritt Web-Aktivität werden zur Laufzeit XML-Dokumente mit anderen Systemen ausgetauscht.

Generalisierungen

- Aktivität auf der Seite 61.

Attribute

- URL: String
URL-Adresse, an die das XML-Dokument gesendet wird.

Assoziationen

- Eingehend: XML-Dokument
zu sendende XML-Datei.
- Ausgehend: XML-Dokument
XML-Datei, die empfangen wird.

Notation



XML-Dokument

Ein Dokument, das dem Datenaustausch zwischen den Systemen dient.

Attribute

- ID: String
Eine eindeutige Bezeichnung der XML-Datei.

DokumentAusVorlage

In diesem Schritt kann mit verschiedenen PC-Anwendungen ein Dokument erstellt werden.

Generalisierungen

- Aktivität auf der Seite 61.

Assoziationen

- Dokument: Dokument
Ein Dokument, das erzeugt wird.

Notation



Dokument

Ein Dokument, das mit einer PC-Anwendung erstellt wird.

Attribute

- Titel: String
Titel des Dokumentes.

3.3.5 SAP Steuerungsschritt

Die Diagramme in Abbildung 3.15 und Abbildung 3.17 definieren alle verwendbaren Steuerungsschritttypen in der Workflow-Definition.

Bedingung

Der Steuerungsschritt Bedingung definiert eine Verzweigung im Workflow-Ablauf. Es wird nur einer von zwei Zweigen durchlaufen. Die Entscheidung trifft das System auf der Grundlage einer definierten Bedingung.

Generalisierungen

- Steuerungsschritt auf der Seite 61.

Attribute

- Bedingung: String
Bedingung, die vom System als wahr oder falsch ausgewertet werden kann.

Constraints

- Hat immer einen Ausgang mit Bezeichnung = "Wahr" und einen Asugang mit Bezeichnung = "Falsch".

Notation



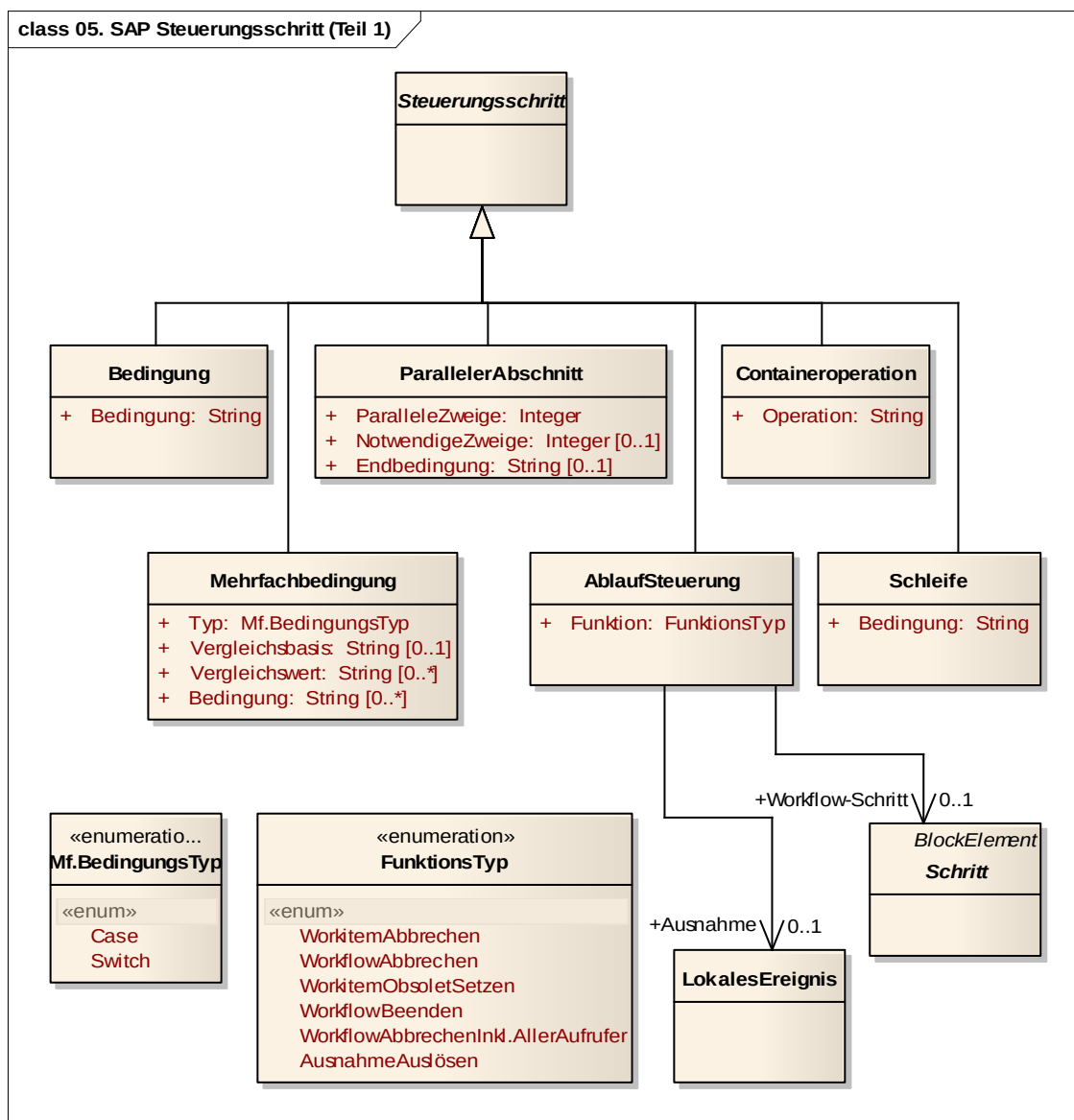


Abbildung 3.15: Diagramm SAP Steuerungsschritt (Teil 1)

Mehrfachbedingung

Der Steuerungsschritt Mehrfachbedingung definiert eine mehrfache Verzweigung im Workflow-Ablauf. Es wird nur einer von mehreren Zweigen durchlaufen. Die Entscheidung trifft das System auf der Grundlage von definierten Bedingungen.

Generalisierungen

- Steuerungsschritt auf der Seite 61.

Attribute

- Typ: Mf.BedingungsTyp
Typ der Mehrfachbedingung.
- Vergleichsbasis: String [0..1]

Vordefinierte Bedingung, die vom System ausgewertet wird.

- Vergleichswert: String [0..*]
Werte, mit welchen das Ergebnis der Auswertung aus einer Vergleichsbasis verglichen wird bis einer übereinstimmt.
- Bedingung: String [0..*]
Vordefinierte Bedingungen, die vom System ausgewertet werden, bis eine als „wahr“ ausgewertet wird.

Constraints

- Hat mindestens 3 Ausgänge.
- Ein Ausgang hat immer die Bezeichnung = "Sonstige Werte" und wird ausgewählt, falls alle anderen Bedingungen nicht zutreffen.
- Mehrfachbedingung vom Typ = Case hat die Attribute Vergleichsbasis und Vergleichswert.
- Mehrfachbedingung vom Typ = Switch hat das Attribut Bedingung.

Notation



Mf.BedingungsTyp

Der Enumeration Typ Mf.BedingungsTyp definiert textuelle Werte, die beschreiben, wie die Entscheidung bei der Mehrfachbedingung getroffen wird:

- Case – nur eine Bedingung wird ausgewertet und das Ergebnis wird mit den vordefinierten Werten verglichen.
- Switch – mehrere vordefinierte Bedingungen werden ausgewertet, bis eine „wahr“ liefert.

ParallelerAbschnitt

Der Schritttyp ParallelerAbschnitt beschreibt, dass die nachfolgende Bearbeitung in parallelen Zweigen stattfindet.

Generalisierungen

- Steuerungsschritt auf der Seite 61.

Attribute

- ParalleleZweige: Integer
Anzahl der Zweige im parallelen Abschnitt.

- NotwendigeZweige: Integer [0..1]
Optionales Attribut ; definiert Anzahl der Zweige, die durchlaufen werden müssen, damit der parallele Abschnitt erfolgreich beendet wird.
- Endbedingung: String [0..1]
Optionale Bedingung, die erfüllt werden muss, damit der parallele Abschnitt erfolgreich beendet wird.

Constraints

- Hat keinen Ausgang.
- Falls das Attribut NotwendigeZweige nicht definiert ist, muss die Endbedingung definiert werden.

Notation

Der ParallelerAbschnitt besitzt grafische Notationen für den Schritt und für den Join:



Ein Beispiel für diesen Schritt ist auch in Abbildung 3.16 zu sehen. Die Aktivität1 und Aktivität2 werden quasi gleichzeitig ausgeführt. Der Parallele Abschnitt ist beendet, wenn die beiden Aktivitäten beendet sind.

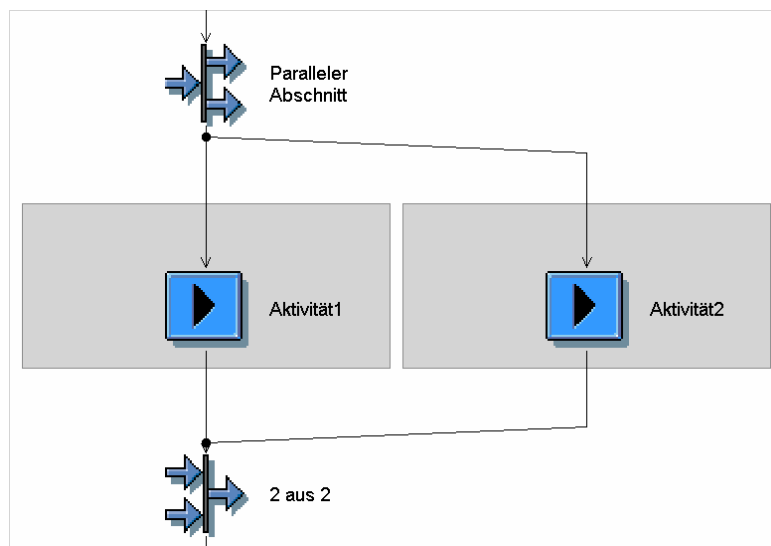


Abbildung 3.16: Paralleler Abschnitt in der SAP Workflow-Definition

AblaufSteuerung

Der Schrittyp AblaufSteuerung wird benötigt, um den laufenden Workflow abubrechen oder eine Ausnahme auszulösen.

Generalisierungen

- Steuerungsschritt auf der Seite 61.

Attribute

- Funktion: Funktionstyp
Handlung, die beim Aktivieren dieses Schrittes ausgeführt wird.

Assoziationen

- Ausnahme: LokalesEreignis [0..1]
Lokale Ausnahme, die ausgelöst wird.
- Workflow-Schritt: Schritt [0..1]
Workflow-Schritt, der beendet oder abgebrochen wird.

Constraints

- Assoziation Ausnahme nur, wenn Attribut Funktion = AusnahmeAuslösen.
- Assoziation Workflow-Schritt nur, wenn Attribut Funktion = WorkitemAbbrechen oder WorkitemObsoletSetzen.
- Als Workflow-Schritt können nur Schrittypen Aktivität, Benutzerentscheidung, DokumentAusVorlage, WartenAufEreignis und Formular auftreten.

Notation



FunktionsTyp

Der Enumeration Typ FunktionsTyp definiert textuelle Werte, die beschreiben, welche Funktion ausgeführt wird, wenn der Schritt AblaufSteuerung aktiviert wird:

- WorkitemAbbrechen – Ausführung eines Schrittes wird abgebrochen.
- WorkflowAbbrechen – Ausführung des ganzen Workflows wird beendet. Falls sich der Schritt Ablaufsteuerung in einem Sub-Workflow befindet, wird der Zweig des aufrufenden Workflows angehalten.
- WorkitemObsoletSetzen – ein Schritt mit der Terminüberwachung mit dem ÜberwachungsTyp = Modelliert wird abgebrochen. Ausführung des Workflows wird an dem Schritt-Ausgang mit der Bezeichnung "Schritt obsolet" fortgesetzt.
- WorkflowBeenden – Ausführung des ganzen Workflows wird beendet. Falls sich der Schritt Ablaufsteuerung in einem Sub-Workflow befindet, wird der Zweig des aufrufenden Workflow fortgesetzt.

- WorkflowAbbrechenInkl.AllerAufrufer – Ausführung des Workflows sowie aller aufrufenden Workflow wird beendet.
- AusnahmeAuslösen – ein lokales Ereignis vom Typ Ausnahme wird ausgelöst.

Schleife

Eine Schleife dient dem wiederholtem Durchlauf einer Folge von Schritten in der Workflow-Definition. Die Entscheidung für den Durchlauf trifft das System auf Grund einer vordefinierten Bedingung.

Generalisierungen

- Steuerungsschritt auf der Seite 61.

Attribute

- Bedingung: String
Bedingung, die vom System als wahr oder falsch ausgewertet werden kann.

Constraints

- Hat immer einen Ausgang mit Bezeichnung = „Wahr“ und einen Ausgang mit Bezeichnung = „Falsch“.

Notation



Containeroperation

Im Schrittyp Containeroperation werden zur Laufzeit einzelne Containerelemente des Workflow-Containers mit Rechenoperationen oder mit Wertzuweisungen beeinflusst.

Generalisierungen

- Steuerungsschritt auf der Seite 61.

Attribute

- Operation: String
Operation, die ausgeführt wird.

Notation





Generalisierungen

- Typ: ErzeugerTyp
Typ des erzeugten Ereignisses.
- EreignisSpec: String [0..1]

Spezifiziert das globale Ereignis, das erzeugt werden kann. Dieses Ereignis wird in dem BusinessObjektyp spezifiziert.

Assoziationen

- LokalesEreignis: LokalesEreignis [0..1]
Lokales Ereignis vom Typ Ereignis, das ausgelöst werden kann.

Constraints

- Attribut EreignisSpec nur bei ErzeugerTyp = EreignisAuslösen.
- Assoziation LokalesEreignis nur bei ErzeugerTyp = LokalesEreignisAuslösen.

Notation



ErzeugerTyp

Der Enumeration Typ ErzeugerTyp definiert textuelle Werte, die den Typ des durch Schritt Ereigniserzeuger ausgelösten Ereignisses festlegen:

- EreignisAuslösen – es wird ein Ereignis ausgelöst, das in einem BusinessObjektyp definiert ist.
- LokalesEreignisAuslösen – es wird ein Ereignis ausgelöst, das in der Workflow-Definition angelegt ist.

WartenAufEreignis

Der Schritt WartenAufEreignis hält die Ausführung des Workflows an, bis ein definiertes Ereignis eintritt.

Generalisierungen

- Steuerungsschritt auf der Seite 61.

Attribute

- Typ: WarteTyp
Typ des Ereignisses, auf das gewartet wird.
- EreignisSpec: String [0..1]
Spezifiziert das Ereignis, auf welches gewartet wird.
- Bedingung: String [0..1]
Spezifiziert die Bedingung, auf welche gewartet wird.

Assoziationen

- Terminüberwachung: Terminüberwachung [0..4]
Überwachung der Start- und Endtermine einer Aufgabe zur Laufzeit.

Constraints

- Attribut EreignisSpec nur bei WarteTyp = AufEreignis, AufEreignisÜberWorkflow oder AufLokalesEreignis.
- Attribut Bedingung nur bei WarteTyp = AufBedingung.

Notation



WarteTyp

Enumeration Typ WarteTyp definiert textuelle Werte, die beschreiben, welche Ereignisse den Schritt WartenAufEreignis beenden:

- AufEreignis – der Schritt wird beendet, wenn ein im BusinessObjektyp definiertes Ereignis eintritt.
- AufBedingung – der Schritt wird beendet, wenn eine definierte Bedingung erfüllt wird.
- AufEreignisÜberWorkflow – hier wird im Schritt WartenAufEreignis auf ein Ereignis gewartet, dass zuerst der Workflow empfängt und an den Schritt weitergibt, wenn dieser aktiv wird.
- AufLokalesEreignis – der Schritt wird beendet, wenn ein in der Workflow-Definition angelegtes Ereignis eintritt.

Ad-hoc-Anker

Der Schritt Ad-hoc-Anker kann zur Laufzeit durch einen vordefinierten Workflow ersetzt werden.

Generalisierungen

- Steuerungsschritt auf der Seite 61.

Assoziationen

- EinfügbareWorkflows: Workflow-Muster [1..*]
Workflows, die zur Laufzeit eingesetzt werden können.

Notation



Block(Schritt)

Der Block(Schritt) (um ihn von dem Strukturelementen Block zu unterscheiden) fasst Schritte zusammen, die hintereinander ausgeführt werden, so dass diese auf dieselben lokalen Daten zugreifen können. Mit Hilfe vom Block(Schritt) kann man eine gemeinsame Frist oder Ausnahmebehandlung für die zusammengefassten Schritte definieren.

Generalisierungen

- Steuerungsschritt auf der Seite 61.

Attribute

- BlockArt: BlockArtTyp
Typ des Blocks.
- Endbedingung: String [0..1]
Bedingung, die, falls wahr ausgewertet wird, die Ausführung des Blocks beenden kann.

Assoziationen

- Frist: Terminüberwachung [0..1]
Optionale gemeinsame Frist für die zusammengefassten Elemente.
- Ausnahme: LokalesEreignis [0..*]
Lokales Ereignis vom Typ Ausnahme, welches der Block(Schritt) behandeln kann.

Constraints

- Attribut Endbedingung kommt nicht bei BlockArt = Standard.
- Als Ausnahme nur Lokales Ereignis mit Typ = Ausnahme.

Notation

Der Block(Schritt) besitzt graphische Notation sowohl für den Schritt als auch für den Join:



Ein Beispiel für den Block(Schritt) ist in Abbildung 3.18 zu sehen.

BlockArtTyp

Der Enumeration Typ BlockArtTyp definiert textuelle Werte, die beschreiben, wie die Ausführung innerhalb eines Block(Schrittes) stattfindet.

- Standard – Block wird einmal durchlaufen.
- DynamischeParallelität – Anhand eines mehrzeiligen Containerelementes pro Zeile eines mehrzeiligen Containerelements wird eine Instanz des Blocks erzeugt. Alle Instanzen werden parallel verarbeitet.
- DynamischeSequenz – Anhand eines mehrzeiligen Containerelementes pro Zeile eines mehrzeiligen Containerelements wird eine Instanz des Blocks erzeugt. Alle Instanzen werden sequenziell verarbeitet.

LokalerWorkflow

In einem lokalen Workflow kann man einen Teil des Workflows modellieren, der durch ein lokales Ereignis gestartet wird.

Generalisierungen

- Steuerungsschritt auf der Seite 61.

Assoziationen

- StartEreignis: LokalesEreignis
Ereignis, das den lokalen Workflow auslöst, falls es eintritt.

Constraints

- Als StartEreignis nur LokalesEreignis mit dem Typ = Ereignis.

Notation

Der lokale Workflow besitzt grafische Notationen sowohl für den Schritt als auch für den Join. Allerdings wird er nicht im Haupt-Pfad der Workflow-Definition, sondern heraus gelöst parallel dazu in demselben Diagramm modelliert.



In Abbildung 3.18 ist ein Beispiel für den LokalerWorkflow zu sehen.

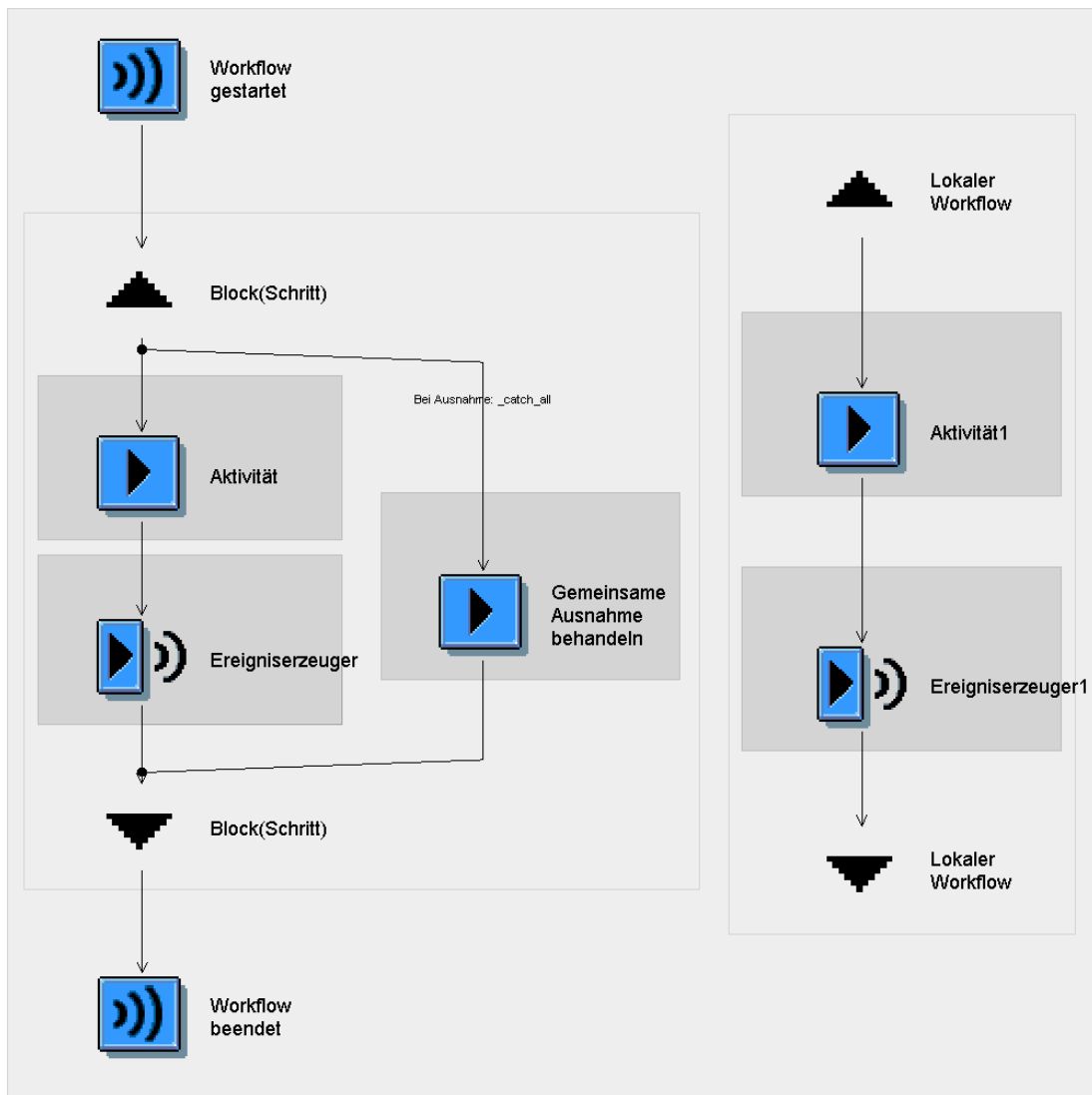


Abbildung 3.18: Block(Schritt) und LokalerWorkflow

4 Modell-zu-Modell Transformation vom BPMN-Modell zum SAP Workflow

4.1 Grundlagen der Modell-zu-Modell Transformation

Modell-zu-Modell Transformation ist ein wichtiges Mittel bei der modellgetriebenen Softwareentwicklung, welches auch bei der Prozessmodellierung angewendet werden kann. Transformationsregeln werden auf der Ebene der Metamodelle definiert, d.h. M2-Schicht der Metamodellierungshierarchie (siehe Abschnitt 3.1). Für die Transformation müssen sowohl das Quell- als auch das Zielmodell Instanzen definierter Metamodelle sein (vgl. [Wik09b]).

Es gibt einige Technologien für Modell-zu-Modell Transformation. Eine der bekanntesten ist Query View Transformation (QVT). QVT ist ein Teil von MOF und somit auch ein Standard von OMG. QVT wird auf die MOF-konformen Metamodelle angewendet und definiert Abbildungen auf Basis von Elementtypen. So können Klassen und die dazugehörigen Attribute eines Modells auf Klassen und passende Attribute eines anderen Modells abgebildet werden.

4.2 Transformationsregeln von BPMN^(SWM)-Metamodell zum SAP Workflow-Metamodell

In dieser Arbeit wurde allerdings das Vorgehen verwendet, welches auch in der BPMN Spezifikation bei der Abbildung von BPMN-Modellen auf BPEL zum Einsatz kam, und zwar die Beschreibung mittels Tabellen. Rechts in der Tabelle steht immer eine Klasse des BPMN^(SWM)-Metamodells bzw. ihre Attribute, und links wird beschrieben, zu welcher Klasse bzw. welchem Attribut des SAP Workflow-Metamodells sie transformiert werden.

Diese Regeln wurden durch die Analyse der Elemente beider Metamodelle und deren Beziehungen, wie in den oberen Abschnitten dieses Kapitels beschrieben, ausgearbeitet.

Die beiden Modelle weisen teilweise sehr große Unterschiede auf, so dass trotz Anpassung des BPMN^(SWM)-Metamodells nicht alle Attribute transformiert werden können. Einige Elemente des SAP Workflows werden abhängig von der Belegung der Attribute in Elementen des BPMN-Modells erzeugt und einige müssen statisch dazu erzeugt werden. Bei einigen besonders komplizierten Fällen wurden zusätzlich Klassendiagramme erstellt.

4.2.1 BPMN Core Transformation

Die in dem Abschnitt BPMN Core definierten Elemente werden wie in Tabelle 4.1 und Tabelle 4.2 beschrieben transformiert.

Tabelle 4.1: Transformation vom BPMNObject

BPMN-Element / Attribut	Transformation zu SAP Workflow
Name	Wird im Allgemeinen zum Attribut Bezeichnung eines passenden Elementes im SAP Workflow transformiert. Näheres siehe unten bei der Transformation von einzelnen Elementen.
Id, Categories und Documentation	Diese Attribute werden zu keinem Element im SAP Workflow transformiert.

Tabelle 4.2: Transformation vom BusinessProcess und Diagram

BPMN-Element / Attribut	Transformation zu SAP Workflow
Name	Dieses Attribut kann zur Bezeichnung von Workflow-Muster transformiert werden.
Language, QueryLanguage, ExpressionLanguage	Werden zu keinem Element im SAP Workflow transformiert.
Diagram	Wird zu keinem Element im SAP Workflow transformiert.

4.2.2 BPMN Activity Transformation

Das Element Activity wird abhängig von der Belegung seiner Attribute und der Unterklasse unterschiedlich transformiert. Grundsätzlich wird die Task auf eine Aktivität transformiert, die eine Standardaufgabe referenziert; Sub-Process vom Typ Independent auf eine Aktivität, die ein Workflow-Muster referenziert. Sub-Process vom Typ Embedded kann unterschiedlich transformiert werden.

Detailliert wird das in den Abschnitten Transformation einer looping Activity, Transformation von Task und Transformation vom Sub-Process beschrieben.

Transformation einer looping Activity

LoopType = None

Im Allgemeinen wird eine BPMN Activity zusammen mit dem ausgehenden SequenceFlow wie in der Tabelle 4.3 beschrieben transformiert. Aus Einfachheitsgründen wird hier angenommen, dass es sich um eine Activity handelt, die auf eine Aktivität abgebildet werden kann.

Tabelle 4.3: Activity Transformation, LoopType = None

BPMN-Element / Attribut	Transformation zu SAP Workflow
Activity	Erzeugt einen Block mit dem Schritt Aktivität.
Activity.Name	Kopiert zu Aktivität.Bezeichnung.
SequenceFlow mit Source = Activity	Transformiert zu Ausgang wo Aktivität.Ausgang = Ausgang.
SequenceFlow.Target	Transformiert zu Aktivität.Nachfolger was gleichzeitig Ausgang.Ziel ist.

LoopType = Standard

Activity mit dem LoopType = Standard wird abhängig vom Attributwert TestTime transformiert (siehe dazu Tabelle 4.4 sowie Abbildungen Abbildung 4.1 bis Abbildung 4.7).

Tabelle 4.4: Activity Transformation, LoopType = Standard

BPMN-Element / Attribut	Transformation zu SAP Workflow
Activity + SequenceFlow mit Source = Activity	Erzeugt ein Block mit dem Schritt Aktivität und einem Ausgang, wie bereits in Tabelle 4.3 beschrieben.
LoopType = Standard	Erzeugt einen Block mit den Elementen Schleife und Join. Schleife hat standardmäßig 2 Ausgänge: – Ausgang zum erneuten Schleifendurchlauf mit Ausgang.Bezeichnung = Falsch – Ausgang, der den Durchlauf beendet mit Aus-

	gang.Bezeichnung = Wahr. Dieser Block dient als ParentBlock für das Aktivität-Block.
TestTime = After	Setzt – Join.Nachfolger = Aktivität – Aktivität.Nachfolger = Schleife – Ausgang.Ziel = Join wo Bezeichnung = „Falsch“
TestTime = Before	Setzt – Join.Nachfolger = Schleife – Aktivität.Nachfolger = Join – Ausgang.Ziel = Aktivität wo Bezeichnung = „Falsch“
Activity.LoopCondition + Activity.LoopMaximum	Transformiert zu Schleife.Bedingung
SequenceFlow.Target	Transformiert zu Ausgang.Ziel des Ausgangs mit Bezeichnung = Wahr, was gleichzeitig ein Schleife.Nachfolger ist

Folgende Abbildungen Abbildung 4.1 bis Abbildung 4.4 veranschaulichen das Vorgehen bei der Transformation einer Activity mit LoopType = Standard und TestTime = After.

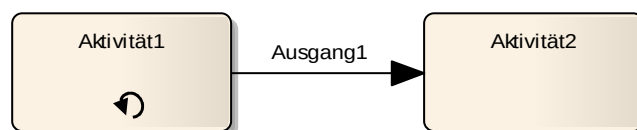


Abbildung 4.1: Looping Activity in BPMN

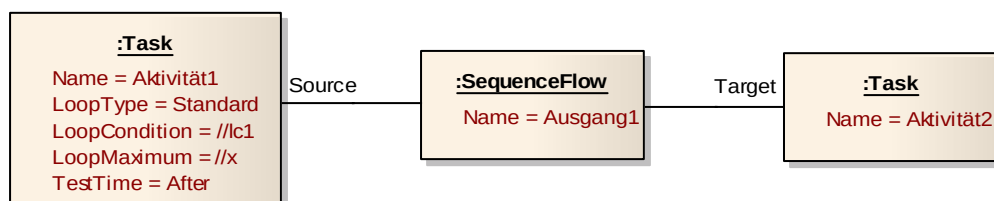


Abbildung 4.2: Looping Activity mit TestTime = After als Objektdiagramm

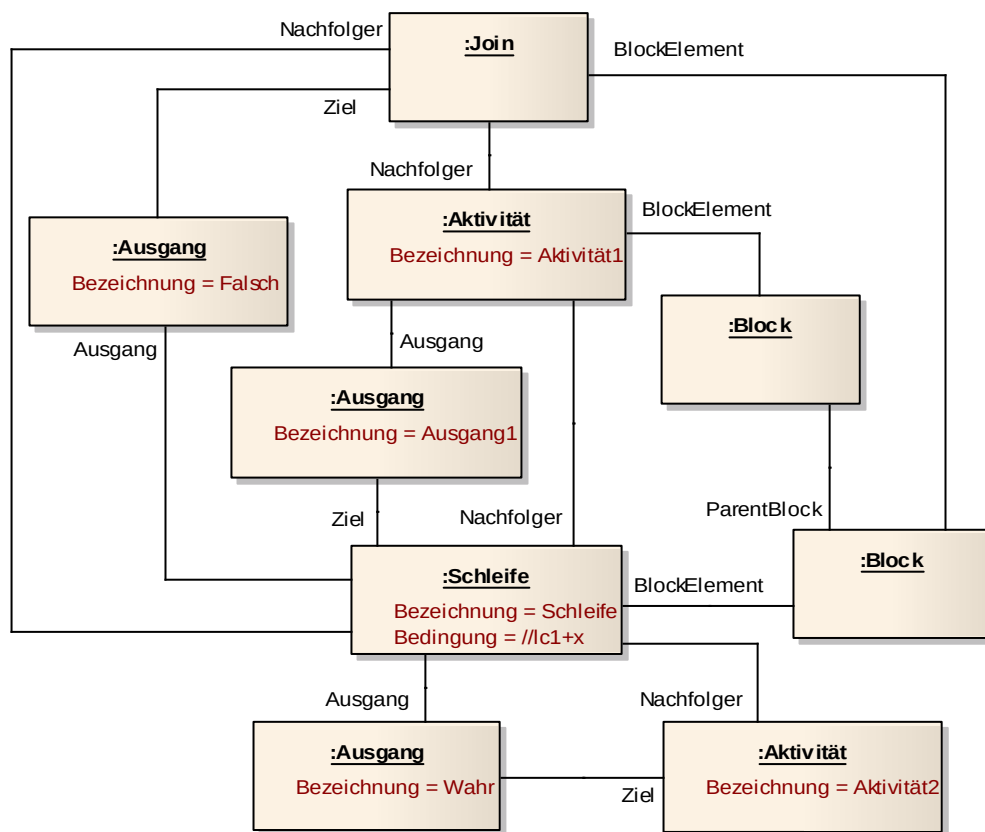


Abbildung 4.3: Looping Activity mit TestTime = After transformiert zu SAP als Objektdiagramm

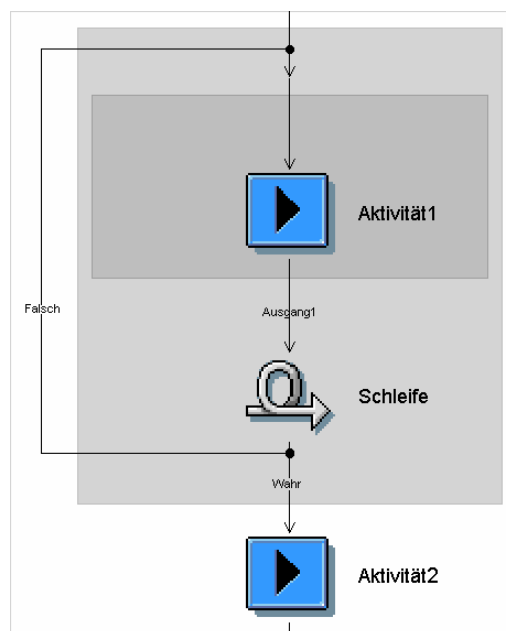


Abbildung 4.4: Looping Activity mit TestTime = After transformiert zu SAP als Schleife in der Workflow-Definition

Die Abbildung 4.5 bis Abbildung 4.7 veranschaulichen das Vorgehen bei der Transformation einer Activity mit LoopType = Standard und TestTime = Before. Modelliert in BPMN entspricht der Abbildung 4.1, da das Attribut TestTime nicht nach außen sichtbar ist.

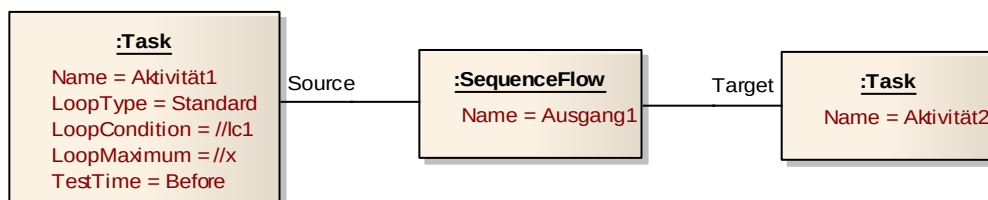


Abbildung 4.5: Looping Activity mit TestTime = Before als Objektdiagramm

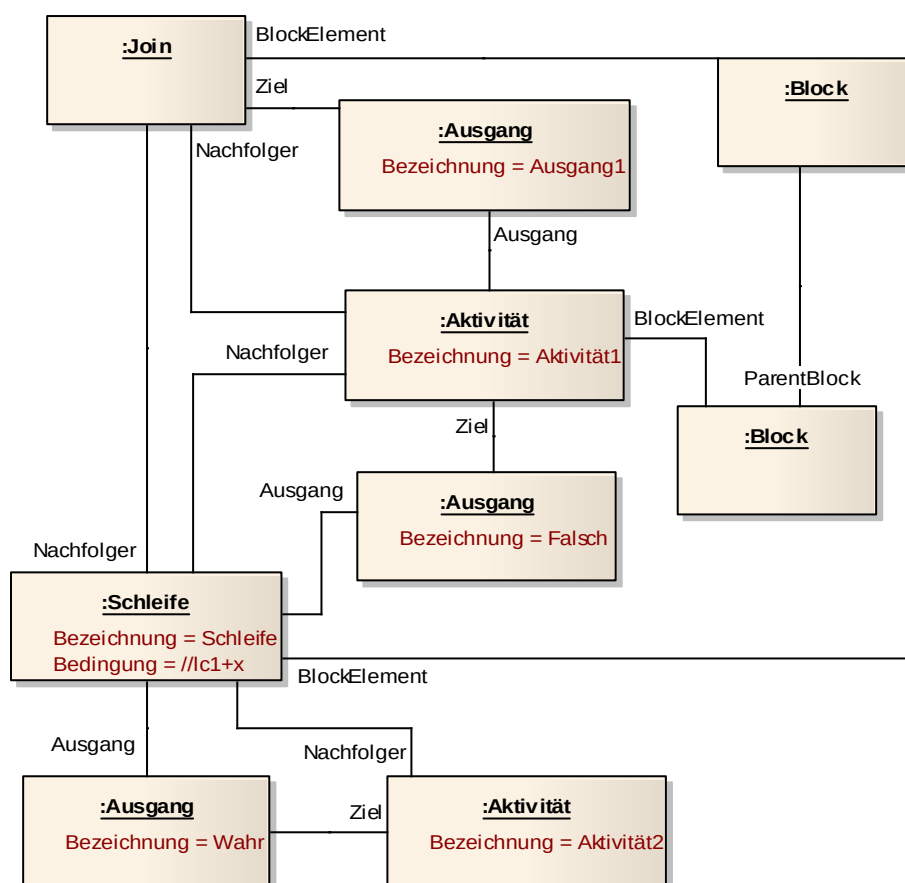


Abbildung 4.6: Looping Activity mit TestTime = Before transformiert zu SAP als Objektdiagramm

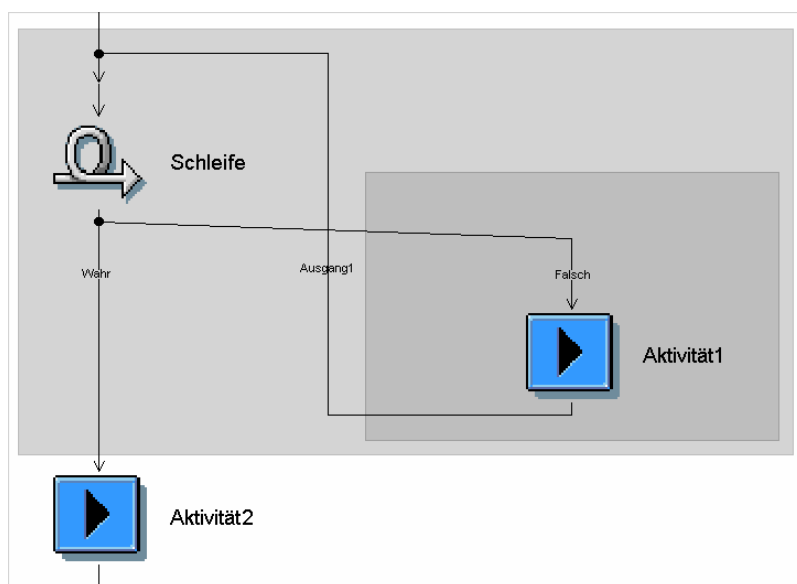


Abbildung 4.7: Looping Activity mit TestTime = Before transformiert zu SAP als Schleife in der Workflow-Definition

LoopType = MultiInstance

Activity mit dem LoopType = MultiInstance wird abhängig vom Attribut MI_Ordering wie in der Tabelle 4.4 beschrieben transformiert.

Tabelle 4.5: Activity Transformation, LoopType = MultiInstance

BPMN-Element / Attribut	Transformation zu SAP Workflow
Activity + SequenceFlow mit Source = Activity	Erzeugt einen Block mit dem Schritt Aktivität und einem Ausgang, wie bereits in der Tabelle 4.3 beschrieben.
LoopType = MultiInstance	Erzeugt einen Block mit dem Element Block(Schritt), das als ParentBlock für das Aktivität-Block dient.
MI_Ordering = Sequential	Setzt Block(Schritt).BlockArt = DynamischeSequenz
MI_Ordering = Parallel	Setzt Block(Schritt).BlockArt = DynamischeParallelität
MI_Condition	Kann momentan nicht transformiert werden.

Transformation von Task

Eine Task wird abhängig vom Attribut TaskType wie in der Tabelle 4.6 beschrieben transformiert.

Tabelle 4.6: Activity Transformation: Task

BPMN-Element / Attribut	Transformation zu SAP Workflow
TaskType = None	Grundsätzlich sollen die None-Tasks auf der 3. Ebene des BPMN-Modells nicht vorkommen. Sie kann aber zu einer Aktivität transformiert werden.
TaskType = Send	Transformiert zum Schritt Aktivität, der eine Standardaufgabe mit dem Typ = Hintergrundaufgabe referenziert. Zu beachten ist hier, dass der Typ einer Standardaufgabe sich nicht im Workflow-Builder definieren lässt. Es wäre aber durch eine Code-Transformation möglich, die in der parallelen Diplomarbeit ausgearbeitet wurde.
TaskType = Receive	Transformiert zum Schritt wartenAufEreignis mit Typ = AufEreignis.
TaskType = User	Transformiert zum Schritt Aktivität, der eine Standardaufgabe mit dem Typ = Dialogaufgabe referenziert.
TaskType = Manual	Diese Task wird in der vorgeschlagenen Modellierung teilweise zweckentfremdet verwendet und wird zusammen mit dem nachfolgenden SplittingGateway mit dem GatewayType = XOR und XORType = Data zu einer Benutzerentscheidung transformiert werden. Benutzerentscheidung.Titel wird gleich Schritt.Bezeichnung gesetzt. Näheres siehe im Abschnitt BPMN Gateway Transformation.
TaskType = Service	Transformiert zum Schritt Aktivität, der eine Standardaufgabe mit dem Typ = Hintergrundaufgabe referenziert.
TaskType = Script	Kann zum Schritt Containeroperation transformiert werden.

Performers	Kann zu Aktivität.Standardaufgabe.Bearbeiter transformiert werden.
Message	Wird zu keinem Element des SAP Workflow transformiert.
Script	Kann zu Containeroperation.Operation transformiert werden. Script wird momentan in den BPMN Modellen nicht verwendet. Sinn und Mehrwert seiner Definition und Transformation sind fraglich.

Transformation vom Sub-Process

Der Sub-Process wird abhängig vom Attribut Sub-ProcessType wie in der Tabelle 4.7 beschrieben transformiert.

Tabelle 4.7: Activity Transformation: Sub-Process

BPMN-Element / Attribut	Transformation zu SAP Workflow
Sub-ProcessType = Independent	Wird immer zu einer Aktivität transformiert, die ein workflow-Muster referenziert. Der eigentliche Prozess wird als externes workflow-Muster mit eigener Workflow-Definition angelegt.
ProcessRef	Wird zu keinem Element des SAP Workflow transformiert.
Sub-ProcessType = Embedded	Kann zu einer Aktivität, die ein workflow-Muster referenziert, oder zu einem Block(Schritt) transformiert werden. Der Embedded Sub-Process vom LoopType = Standard ohne weitere Besonderheiten (wie z.B. die angehefteten IntermediateEvents) kann auch in den SAP Workflow integriert werden, wie in der Tabelle 4.4 beschrieben ist. Der Unterschied besteht nur darin, dass statt eines Blocks mit dem Schritt Aktivität der entsprechende interne Prozess vom Embedded Sub-Process eingesetzt wird.
GraphicalElements	Bilden den internen Aufbau eines workflow-Musters, eines Block(Schrittes), oder, falls der SubProcess

	den LoopType = Standard hat, werden in den Schritt Schleife integriert.
--	---

4.2.3 BPMN Event Transformation

Transformation vom StartEvent

Transformation vom StartEvent ist in der Tabelle 4.8 beschrieben.

Tabelle 4.8: StartEvent Transformation

BPMN-Element / Attribut	Transformation zu SAP Workflow
Trigger mit Typ = None Message, Timer und Rule	StartEvent von einem Workflow-Gerüst wird zu einem Schritt WF-gestartet im SAP Workflow transformiert. StartEvent von einem Sub-Process wird abhängig von der Transformation dieses Sub-Process behandelt. Falls der Sub-Process zum Schritt Aktivität transformiert wird – wie oben. Falls der Sub-Process zum Block(Schritt) transformiert wird – wird der StartEvent zu keinem Element im SAP Workflow transformiert.
Trigger mit Typ = Link	Wird zum Schritt wartenAufEreignis mit dem Typ = AufLokalesEreignis transformiert (siehe dazu das Beispiel in Abbildung 4.8 bis Abbildung 4.10).
Trigger.LinkId	Wird bei der Definition von Lokalen Ereignissen verwendet; zur Zeit noch nicht näher spezifiziert.
Andere Attribute vom Trigger	Werden zu keinem Element des SAP Workflow transformiert.

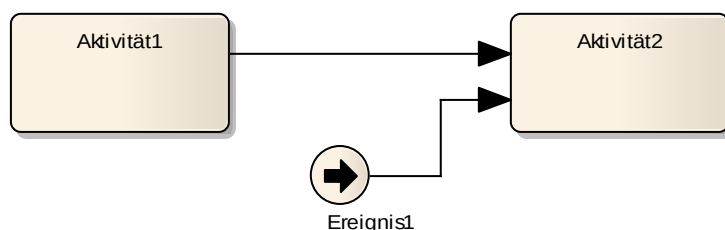


Abbildung 4.8: Link StartEvent in BPMN

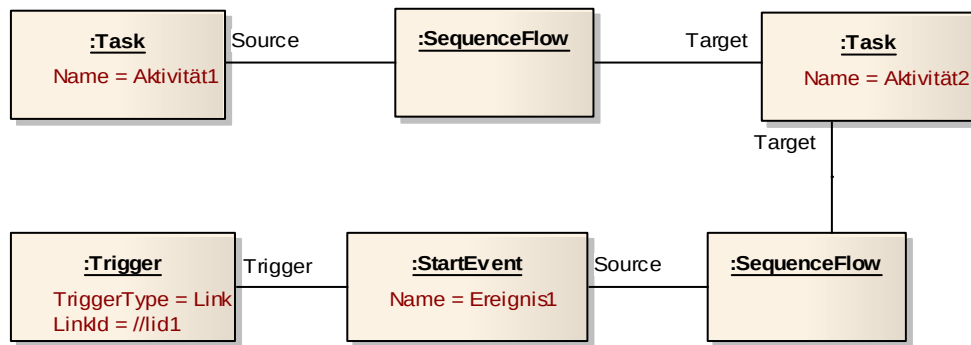


Abbildung 4.9: Link StartEvent als Objektdiagramm

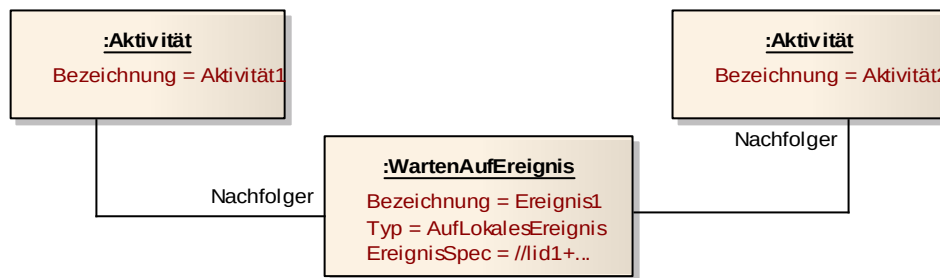


Abbildung 4.10: Link StartEvent transformiert zu SAP Workflow als Objektdiagramm

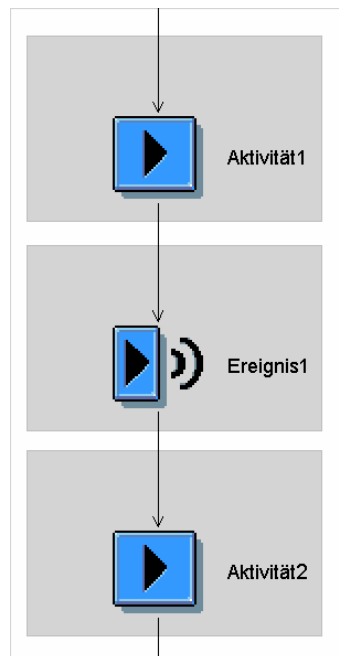


Abbildung 4.11: Ereigniserzeuger in der Workflow-Definition

Transformation vom EndEvent

Transformation vom EndEvent ist in der Tabelle 4.9 beschrieben.

Tabelle 4.9: EndEvent Transformation

BPMN-Element / Attribut	Transformation zu SAP Workflow
Result mit Typ = None oder Message	EndEvent von einem Workflow-Gerüst wird zu einem Schritt WF-beendet im SAP Workflow transformiert. EndEvent von einem Sub-Process wird abhängig von der Transformation von diesem Sub-Process behandelt. Falls der Sub-Process zum Schritt Aktivität transformiert wird – wie oben. Falls der Sub-Process zum Block(Schritt) oder als interner Aufbau der Schleife transformiert wird – wird der EndEvent zu keinem Element im SAP Workflow transformiert.
Result mit Typ = Link	Wird zum Schritt Ereigniserzeuger mit dem Typ = LokalesEreignisAuslösen transformiert. Analog zum Link StartEvent.
Result mit Typ = Error	Wird zum Schritt AblaufSteuerung mit Funktion = AusnahmeAuslösen transformiert.
Result mit Typ = Terminate	Wird zum Schritt AblaufSteuerung mit Funktion = workflowAbbrechenInkl.AllerAufrufer transformiert.
Result.LinkId	Wird bei der Definition von Lokalen Ereignissen verwendet (LokalesEreignis, Typ = Ereignis); zur Zeit noch nicht näher spezifiziert.
Result.ErrorCode	Wird bei der Definition von Lokalen Ausnahmen verwendet (LokalesEreignis, Typ = Ausnahme); zur Zeit noch nicht näher spezifiziert.
Andere Attribute vom Result	Werden zu keinem Element des SAP Workflow transformiert.

Transformation vom IntermediateEvent

IntermediateEvent wird abhängig vom Trigger und davon, ob er im SequenceFlow oder am Rande einer Activity vorkommt, transformiert. Die entsprechenden Regeln werden in Tabelle 4.10 und Tabelle 4.11 definiert.

Tabelle 4.10: Transformation von IntermediateEvent im SequenceFlow

BPMN-Element / Attribut	Transformation zu SAP Workflow
Trigger mit Typ = Rule	Wird zum Schritt wartenAufEreignis mit Typ = Auf-Bedingung transformiert.
Trigger.RuleName	Wird zur wartenAufEreignis.Bedingung transformiert.
Trigger mit Typ = Timer	Als Teil eines SplittingGateways mit GatewayType = XOR und XORType = Event, siehe den Abschnitt BPMN Gateway Transformation Im normalen SequenceFlow vor einer Activity wird bei dem Schritt Aktivität zur Terminüberwachung mit Termin = GewünschterStart transformiert.
Trigger.TimeDate oder Trigger.TimeCycle	Wird zu Terminüberwachung.BezugsZeitPunkt + ggf. Terminüberwachung.Zeitspanne transformiert.
Trigger mit Typ = Link	Ein Link IntermediateEvent dient zur Erzeugung des virtuellen SequenceFlow und ist für die Transformation irrelevant.

Tabelle 4.11: Transformation von IntermediateEvent am Rand einer Activity

BPMN-Element / Attribut	Transformation zu SAP Workflow
Trigger mit Typ = Message	Transformation von diesem Event ist noch offen, da die Stornobehandlung im SAP Workflow als Schritt wartenAufEreignis im parallelen zum normalen Ablauf Pfad modelliert wird.
Trigger mit Typ = Timer	Am Rand vom Sub-Process, falls dieser als Block(Schritt) transformiert wird, wird der IntermediateEvent zum Block(Schritt).Frist mit Typ =

	Modelliert transformiert. Am Rand einer Activity wird er bei dem Schritt Aktivität zur Terminüberwachung mit Termin = SpätestesEnde und Typ = Modelliert transformiert. Diese Terminüberwachung erzeugt zwei neue Ausgänge dieser Aktivität mit Bezeichnungen „Frist erreicht“ und „Obsolet“. Als Ziel des Ausgangs „Frist erreicht“ soll dann ein Schritt AblaufSteuerung mit – Funktion = WorkitemObsoletSetzen und – Workflow-Schritt = Aktivität.Id eingefügt werden. Wie bereits erwähnt wurde, wird dieser Pfad nicht am Ende des Blocks mit den anderen zusammengeführt, sondern endet in einem WF-beendet ähnlichem BlockElement. In diesem Fall wurde dieses Element als Join für den Block der AblaufSteuerung betrachtet und mit sich selbst als Nachfolger assoziiert. Dies ist zwar keine korrekte aber mögliche Lösung. Activity, die nach dem Auslösen von IntermediateEvent ausgeführt wird, wird zum Ziel des Ausgangs mit der Bezeichnung „Obsolet“ transformiert. Dieser Sachverhalt wird in Abbildung 4.12 bis Abbildung 4.15 noch einmal deutlich gemacht.
Trigger.TimeDate oder Trigger.TimeCycle	Wird zu Terminüberwachung.BezugsZeitPunkt + ggf. Terminüberwachung.Zeitspanne transformiert.
Trigger mit Typ = Error oder Rule	Am Rand vom Sub-Process, falls dieser als Block(Schritt) transformiert wird, wird zum Block(Schritt).Ausnahme transformiert. Am Rand von einer Task wird zu einem Ausgang entsprechender Aktivität.
Trigger.ErrorCode und Trigger.RuleName	Bleibt noch offen.
Trigger.Message	Wird zu keinem Element des SAP Workflow transformiert.

In Abbildung 4.12 bis Abbildung 4.15 wird die Transformation von einem angehefteten Timer IntermediateEvent in BPMN auf die modellierte Terminüberwachung im SAP Workflow veranschaulicht. Bei dem Objektdiagramm zum SAP Workflow wurde aus Gründen der Übersichtlichkeit auf die Darstellung von allen Elementen, insbesondere vom Block, verzichtet.

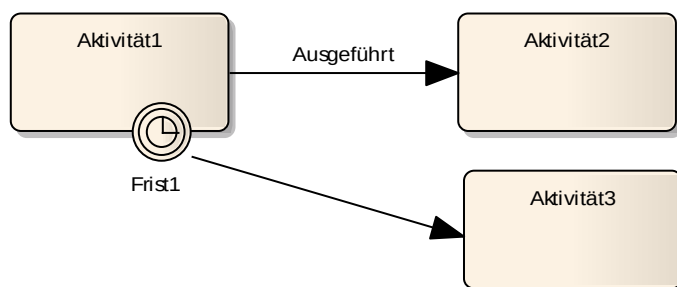


Abbildung 4.12: Terminüberwachung in BPMN

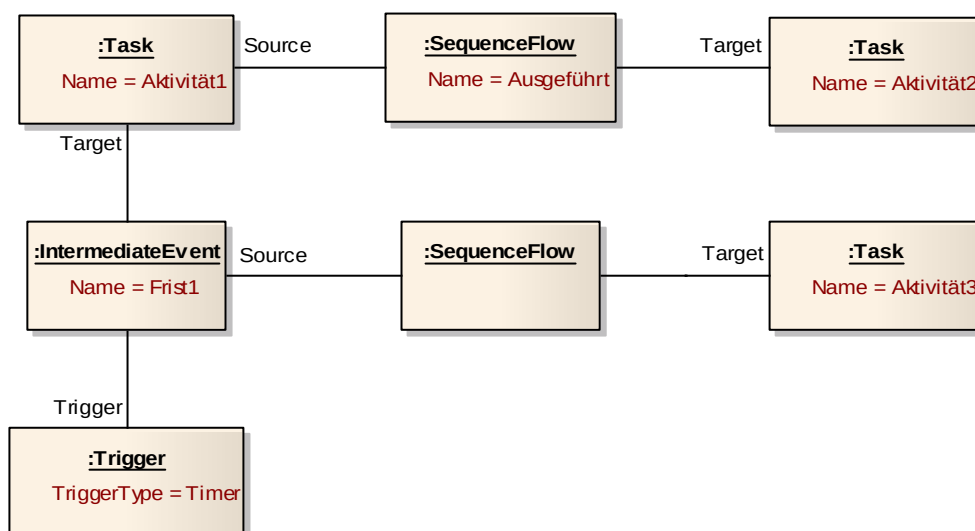


Abbildung 4.13: Terminüberwachung in BPMN als Objektdiagramm

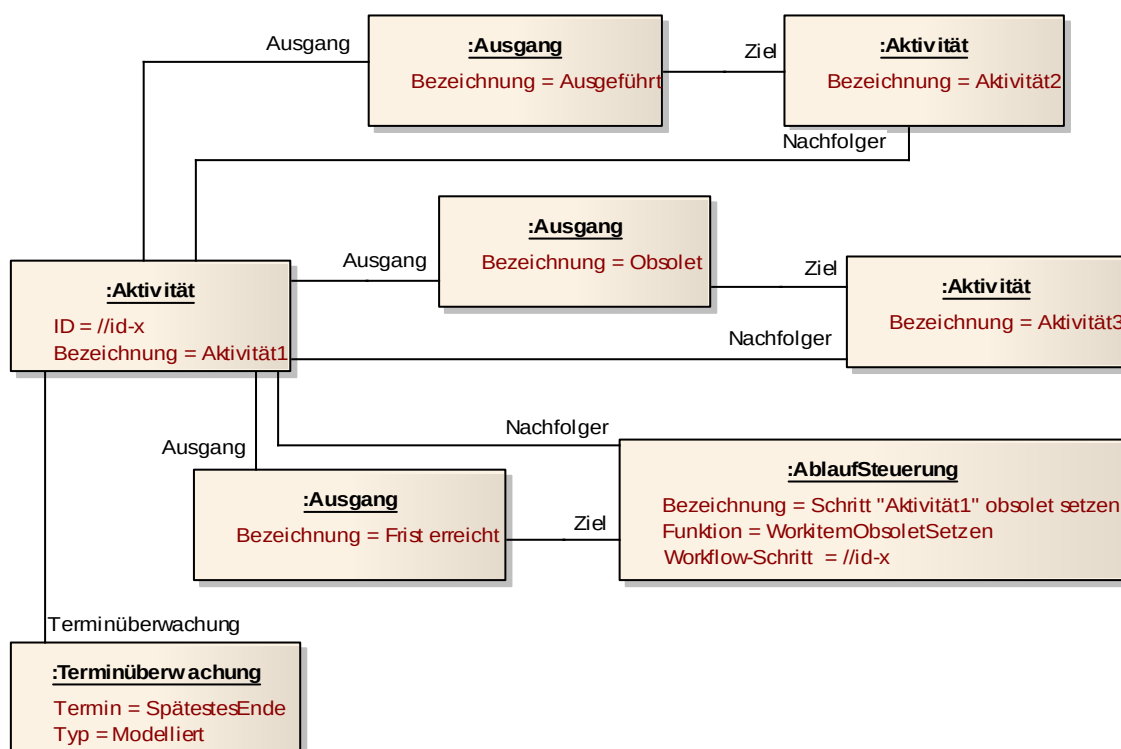


Abbildung 4.14: Modellerte Terminüberwachung im SAP Workflow als Objekt-diagramm

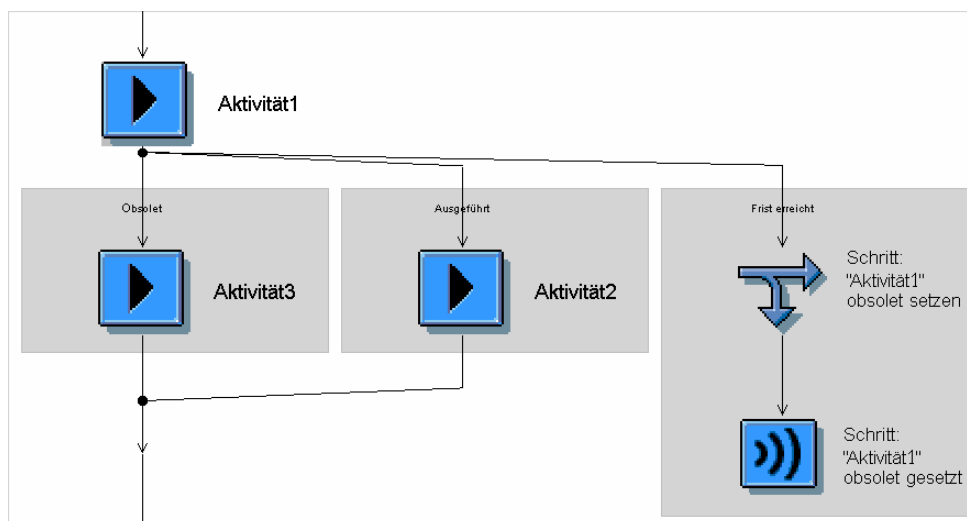


Abbildung 4.15: Modellerte Terminüberwachung im SAP Workflow

4.2.4 BPMN Gateway Transformation

Transformation vom BPMN Element Gateway ist in der Tabelle 4.12 bis Tabelle 4.14 beschrieben.

Tabelle 4.12: Transformation vom SplittingGateway, GatewayType = XOR, XORType = Data

BPMN-Element / Attribut	Transformation zu SAP Workflow
SplittingGateway mit 2 OutgoingSequenceFlows	Wird zum Schritt Bedingung transformiert. Dabei gilt Folgendes: (siehe die nächsten 3 Zeilen)
OutgoingSequenceFlow. ConditionExpression	Nicht negierte ConditionExpression von einem SequenceFlow kann zu Bedingung.Bedingung transformiert werden.
OutgoingSequenceFlow	Mit nicht negierter ConditionExpression wird zum Ausgang mit der Bezeichnung = Wahr transformiert. Mit der negierten ConditionExpression – zum Ausgang mit der Bezeichnung = Falsch.
SequenceFlow.Name	Vom SequenceFlow mit nicht negierten ConditionExpression wird zu Ausgang.Bezeichnung kopiert.
SplittingGateway mit mehr als 2 OutgoingSequenceFlows	Wird zum Schritt Mehrfachbedingung vom Typ = Switch transformiert. Einzelheiten siehe die nächsten 3 Zeilen
OutgoingSequenceFlow mit ConditionType=Expression	Erzeugt jeweils einen Ausgang.
OutgoingSequenceFlow. Name	Wird zu Ausgang.Bezeichnung kopiert.
OutgoingSequenceFlow. ConditionExpression	Wird dabei zu Mehrfachbedingung.Bedingung transformiert.
OutgoingSequenceFlow mit ConditionType = Default	Wird zu einem Ausgang mit der Bezeichnung = Sonstige Werte transformiert.
Nach einer Task vom TaskType = User	Kann mit dem Schritt Aktivität (als Ergebnis einer Task-Transformation) zusammengefasst werden.
Nach einer Task vom TaskType = Manual	Wird zum Schritt Benutzerentscheidung zusammengefasst.

OutgoingSequenceFlow	Erzeugt jeweils einen Ausgang.
OutgoingSequenceFlow. Name	Wird zu Ausgang.Bezeichnung kopiert. Im Fall von Benutzerentscheidung wird auch zu Benutzerentscheidung.Alternative kopiert.

Tabelle 4.13: Transformation vom SplittingGateway, GatewayType = XOR, XORType = Event

BPMN-Element / Attribut	Transformation zu SAP Workflow
Gateway gefolgt vom Timer IntermediateEvent und einem Rule IntermediateEvent oder einer Receive Task.	Kann folgenderweise transformiert werden: (siehe die nächsten 3 Zeilen)
Rule IntermediateEvent	Auf einen Schritt wartenAufEreignis vom Typ = AufBedingung transformiert werden, wie in der Tabelle 4.10 beschrieben.
Receive Task.	Kann auf einen wartenAufEreignis mit Typ = AufEreignis transformiert werden.
Timer IntermediateEvent	Auf wartenAufEreignis.Terminüberwachung wie bereits in der Tabelle 4.11 beschrieben ist.
Gateway gefolgt vom Timer IntermediateEvent und 2 oder mehr Rule IntermediateEvents bzw Receive Tasks.	Wird zu einem Schritt ParallelerAbschnitt transformiert, wobei Folgendes gilt: – ParalleleZweige = Anzahl von OutgoingSequenceFlows minus 1 – NotwendigeZweige = 1
Rule IntermediateEvent und Receive Task.	Werden wie bereits oben beschrieben Transformiert.
Timer IntermediateEvent	Transformiert zur Terminüberwachung von nur einem Schritt wartenAufEreignis wie der Tabelle 4.11 beschrieben.

Tabelle 4.14: Transformation vom SplittingGateway, GatewayType = AND

BPMN-Element / Attribut	Transformation zu SAP Workflow
SplittingGateway mit GatewayType = AND	Wird zum Schritt ParallelerAbschnitt transformiert, wobei gilt: ParalleleZweige = Anzahl von SplittingGateway. OutgoingSequenceFlow
Zusammen mit dem MergingGateway vom GatewayType = AND	NotwendigeZweige = ParalleleZweige.
Zusammen mit dem MergingGateway vom GatewayType = Complex	MergingGateway.OutgoingCondition wird zu ParallelerAbschnitt.Endbedingung transformiert oder definiert NotwendigeZweige < ParalleleZweige.

MergingGateway mit GatewayType = XOR wird zum Element Join transformiert.

4.2.5 BPMN ConnectingObject Transformation

ConnectingObjects werden wegen der Blockorientierung von SAP Workflow nur im Zusammenhang mit den FlowObjects transformiert; im Allgemeinen aber zum Ausgang, wobei ConnectingObject.Name zum Ausgang.Bezeichnung kopiert wird.

4.2.6 BPMN Swimlane Transformation

Werden zu keinem Element der SAP Workflow-Definition transformiert.

4.2.7 BPMN Artifact Transformation

Im Allgemeinen werden sie zu keinem Element der SAP Workflow-Definition transformiert. In der zukünftigen Arbeit kann die Möglichkeit betrachtet werden, Artifacts zusammen mit den FlowObjects zu transformieren und somit die zusätzliche Information zu gewinnen.

5 Anwendung des Konzepts anhand des ausgewählten Beispiels

Als Modellierungsbeispiel diene die gesetzliche Vorlage für das Vorgehen bei dem Gaslieferantenwechsel: GeLi Gas, siehe [GeLi07]. Die in diesem Dokument identifizierten Geschäftsprozesse wurden gemäß den, an die SWM Bedürfnisse angepassten BPMN Regeln, modelliert.

Als Beispiel für diese Arbeit wurde das Geschäftsprozess „Lieferantenwechsel“ ausgewählt, deren textuelle Beschreibung in der Tabelle 5.1 zu finden ist.

Die dabei verwendeten Abkürzungen:

L – Letztverbraucher

A – Altlieferant

N – Neulieferant

NB – Netzbetreiber

Tabelle 5.1: Beschreibung des Prozesses "Lieferantenwechsel" (Quelle [GeLi07])

Nr.	Sender	Empfänger	Beschreibung des Prozessschrittes	Frist	Anmerkungen
1	N	L	Neulieferant erstellt Angebot für einen Gasliefervertrag an Letztverbraucher	-	Neulieferant prüft ggf. vorab durch eine Geschäftsdaten-anfrage beim Netzbetreiber, welchem Marktgebiet die Entnahmestelle bislang zugeordnet ist. Neulieferant klärt mit Letztverbraucher, ob Lieferantenwechsel oder Lieferbeginn vorliegt
2	L	N	Letztverbraucher und Neulieferant schließen einen Gasliefervertrag.	-	Prozess kann ggf. auch durch Kündigung des Altlieferanten oder durch ein Vertragsende eingeleitet werden. In diesen Fällen entfallen die Prozessschritte 3a, 4 und 5a/b
3a	N	A	Kündigung des bestehenden Gasliefervertrages beim Altlieferanten durch den Neulieferanten im Auftrag des Letztverbrauchers.	Gemäß Gasliefervertrag, aber spätestens vor Beginn des Fristenmonats.	Der Letztverbraucher muss dem Neulieferanten eine Vollmacht erteilt haben. Die Kündigung kann sich auf einen fixen Zeitpunkt oder auf den nächstmöglichen Termin beziehen. Der Letztverbraucher kann den Gasliefervertrag auch selbst kündigen.
3b	N	NB	Anmeldung Netznutzung.	Spätestens	Die Anmeldung zur Netznut-

			Mitteilung des Neulieferanten an Netzbetreiber, ob Lieferantenwechsel oder Lieferbeginn vorliegt.	einen Monat vor beabsichtigtem Lieferbeginn	zung und die Kündigung beim Altlieferanten können gleichzeitig erfolgen. Der Neulieferant teilt in der Anmeldung u.a. mit, ob der Letztverbraucher ein „Haushaltskunde“ ist und welchem Marktgebiet die Entnahmestelle künftig zugeordnet werden soll.
4	A	A	Prüfung der Kündigung des Gasliefervertrages durch den Altlieferanten.	Unverzüglich, jedoch spätestens am 5. Werktag nach Eingang der Kündigung	Der Altlieferant prüft u.a. Mindestvertragslaufzeiten und Kündigungsfristen. Ist der Kündigung des Gasliefervertrages bereits eine andere Kündigung vorausgegangen, so können sich folgende Fälle ergeben: Fall 1: Der Kündigungstermin B liegt vor dem Kündigungstermin A. Normale Bearbeitung der Kündigung B entsprechend dem Geschäftsprozess Lieferantenwechsel (Prüfung Kündigungsfrist, Vertragslaufzeit). Fall 2: Der Kündigungstermin B stimmt mit Kündigungstermin A überein. Kunde oder anderer Lieferant haben bereits gekündigt. Ablehnung mit Antwortkategorie „Transaktion schon stattgefunden“. Fall 3: Der Kündigungstermin B liegt nach dem Kündigungstermin A. Die Kündigung B wird mit dem Ablehnungsgrund „kein Vertragsverhältnis“ abgelehnt.
5a	A	N	Bestätigung der Kündigung durch den Alt- an den Neulieferanten bzw. Letztverbraucher.	Unverzüglich, spätestens jedoch am 5. Werktag nach Eingang der Kündigung.	Bestätigung zum gewünschten Kündigungstermin.
5b	A	N	Zurückweisung der Kündigung zum gewünschten Termin vom Altlieferanten gegenüber dem Neulieferanten und ggf. Bestätigung der Kündigung zum nächstmöglichen Kündigungstermin.	Unverzüglich, spätestens jedoch am 5. Werktag nach Eingang der Kündigung	Wenn die Kündigung zum gewünschten Termin zurückgewiesen wird, weil Fristen (z.B. Kündigungsfrist, Mindestvertragslaufzeit) nicht eingehalten wurden, lehnt der Altlieferant die Kündigung zum gewünschten Termin ab und bestätigt die Kündigung zum nächstmöglichen Kündigungstermin, sofern der ursprünglich gewünschte Termin nicht als fixer Termin gekennzeichnet war. Die Kündigung zum

					nächstmöglichen Termin teilt er dem Neulieferanten mit. Sofern ein fixer Kündigungstermin mitgeteilt wurde, ist der nächstmögliche Kündigungstermin lediglich mitzuteilen.
6	A	NB	Abmeldung der Netznutzung durch den Altlieferanten beim Netzbetreiber der Entnahmestelle.	Unverzüglich, jedoch spätestens am 5. Werktag nach Eingang der Kündigung	Zur zügigen Abwicklung hat die Abmeldung der Netznutzung möglichst gleichzeitig mit der Kündigungsbestätigung zu erfolgen.
7	NB	NB	Prüfung der Anmeldung durch Netzbetreiber. Der Netzbetreiber prüft die eingegangenen Meldungen in drei Schritten: 1. Prüfung der Kapazitäten bei Marktgebietswechsel. 2. Prüfung auf Lieferantenkonkurrenz (siehe Abschnitt „Auflösung Konfliktszenarien“). 3. Prüfung aller sonstigen Voraussetzungen, z.B. Prüfung aller gemeldeten Lieferantenwechsel. Er hat alle bis zum 14. Werktag, 10.00 Uhr des Fristenmonats korrekt eingegangenen Abmeldungen zu berücksichtigen. Hierzu gehören auch die Meldungen gemäß dem Abschnitt „Auflösung Konfliktszenarien“.	zu 1.: Unverzüglich zu 2.: Unverzüglich, jedoch spätestens am 10. Werktag des Fristenmonats. zu 3.: Unverzüglich, jedoch spätestens am 15. Werktag des Fristenmonats.	Die ggf. erforderliche Kapazitätsprüfung hat nach geeigneten Branchenstandards zu erfolgen.
8a	NB	N	Übermittlung der Antwort des Netzbetreibers auf die Anmeldung an den Neulieferanten. Bei Ablehnung ist der Grund anzugeben.	Unverzüglich, spätestens bis zum Ablauf des 15. Werktags des Fristenmonats.	Lehnt der Netzbetreiber die Anmeldung zum Anmeldedatum ab, kann der Lieferantenwechsel jedoch zu einem späteren Zeitpunkt durchgeführt werden, so bestätigt der Netzbetreiber außerdem die Anmeldung zu diesem Zeitpunkt. Ist die Auflösung der Lieferantenkonkurrenz für den Neulieferanten gescheitert, wird die Anmeldung des Neulieferanten abgelehnt und der Ablehnungsgrund mitgeteilt. Hierbei ist insbesondere anzugeben, ob ein Fall der Lieferantenkonkurrenz wegen fehlender Abmeldung oder wegen Mehrfachanmeldung vorliegt.
8b	NB	A	Die Abmeldung wird durch den Netzbetreiber gegenüber dem Altlieferanten bestätigt	Unverzüglich, spätestens jedoch bis zum	Der Netzbetreiber muss auch auf nachträglich gemeldete Abmeldungen gem. Abschnitt

			oder abgelehnt.	Ablauf des 15. Werktags des Fristenmonats	B.1.5 „Auflösung Konfliktszenarien bei Lieferantenkonkurrenz“ antworten.
9a/b	NB	A/N	Versand von Bestandslisten durch den Netzbetreiber an die Lieferanten.	Am 16. Werktag des Fristenmonats	Die Bestandsliste enthält die für den Folgemonat relevanten Lieferverhältnisse. Informationen zu Lieferverhältnissen, die nach dem Folgemonat beginnen, sind in dieser Bestandsliste nicht enthalten.
10	NB	NB	Ermittlung der Messwerte durch Netzbetreiber (durch Selbstablesung des Letztverbrauchers, Ablesung durch den Netzbetreiber, Ablesung durch vom Netzbetreiber beauftragten Dritten, rechnerische Abgrenzung/ Schätzung durch den Netzbetreiber, etc.).	Gemäß Prozess „Messwertübermittlung“ (Abschnitt D.1.)“	Inhalt und Umfang der zu ermittelnden Messwerte und Daten ergeben sich aus dem Prozess „Messwertübermittlung“ (Abschnitt D.1.).
11a/b	NB	N, A	Übermittlung der Messwerte oder anderer abrechnungs- oder bilanzierungsrelevanter Daten durch den Netzbetreiber an den Neulieferanten und an den Altlieferanten.	Gemäß Prozess „Messwertübermittlung“ (Abschnitt D.1.)“	Inhalt und Umfang der zu übertragenden Messwerte und Daten ergeben sich aus dem Prozess „Messwertübermittlung“ (Abschnitt D.1.).
12	NB	A	Endabrechnung des Netzzugangs zu der betroffenen Entnahmestelle zwischen Altlieferant und Netzbetreiber.	Unverzüglich, jedoch spätestens am 10. Werktag nach Übermittlung der abrechnungsrelevanten Informationen	Siehe Prozess „Netznutzungsabrechnung“ (Abschnitt D.4.).

Dieser Prozess wurde in der Arbeitsgruppe entsprechend den ausgearbeiteten Regeln in BPMN modelliert. Aus Platzgründen wird in dieser Arbeit nur der Teil des Modells für die SWM Rolle als alter Lieferant vorgestellt. D.h. auch wenn SWM bei einem konkreten Fall des Lieferantenwechsel-Prozesses gleichzeitig als Altlieferant, Neulieferant und Netzbetreiber auftritt, wird der Prozess getrennt für jede Rolle durchlaufen. Dies ist jetzt auch aus Marktliberalisierungsgründen gesetzlich vorgeschrieben.

In Abbildung 5.1 ist das Modell der 2. Ebene zu sehen.

Viele Modellierungs-Tools verstehen die Tasks und Sub-Processes mit eindeutigen graphischen Kennzeichnungen, die deren Typ (z.B. User, Manual etc. bei der Task) angeben. Bei Enterprise Architect wird das nicht angeboten. Aus Übersichtsgründen wurden den Namen von Independent Sub-Processes das „SP.“ vorangesetzt.

Ein Vorschlag für ein Workflow-Gerüst ist in Abbildung 5.8 und Abbildung 5.9 zu sehen. Dieses Workflow-Gerüst wurde in Rahmen dieser Arbeit nicht weiter optimiert und verfeinert, da man hierzu tiefere Kenntnisse für die Unternehmensabläufe benötigt.

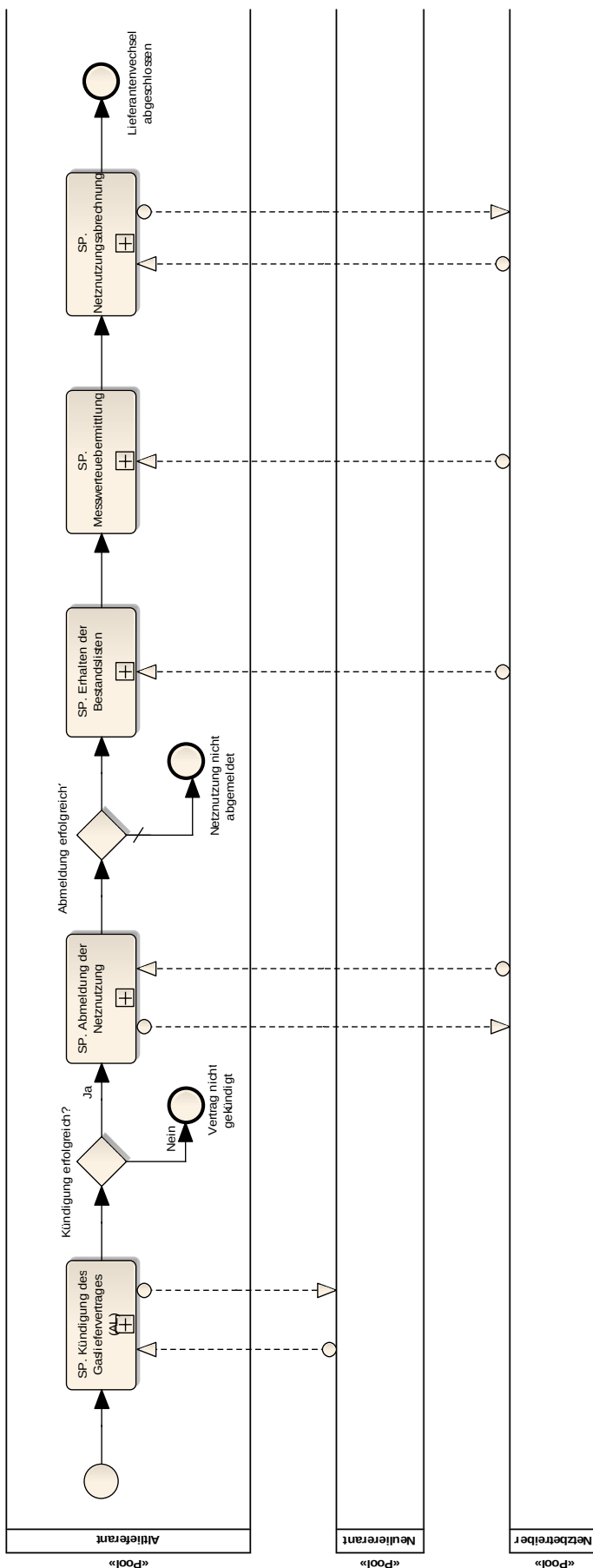


Abbildung 5.1: Geschäftsprozess „Lieferantenwechsel“ 2. Ebene

In Abbildung 5.2 bis Abbildung 5.7 sind die ausführlichen Modellierungen der Sub-Prozesse, d.h. die 3. Modell-Ebene dargestellt:

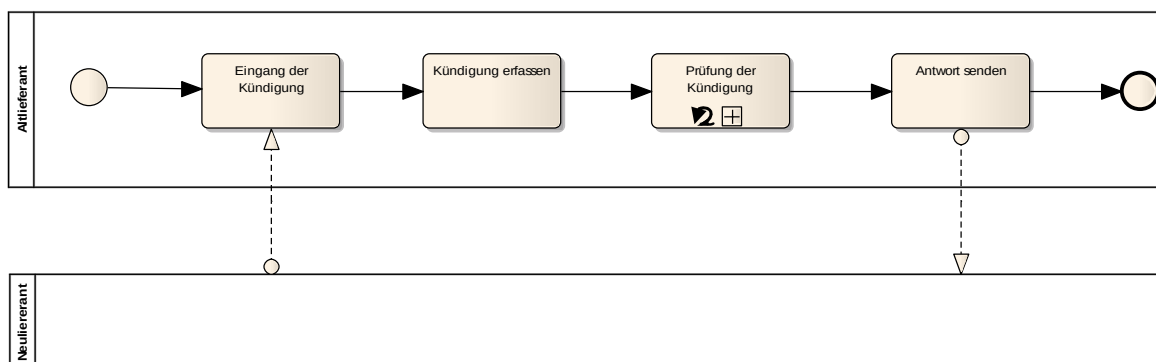


Abbildung 5.2: Independent Sub-Process "Kündigung des Gaslieferungsvertrages"

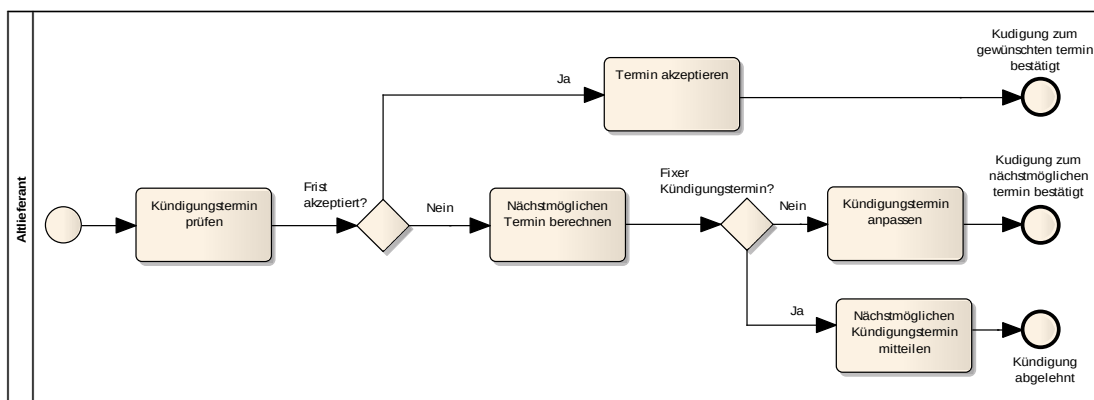


Abbildung 5.3: Embedded Sub-Process "Prüfung der Kündigung"

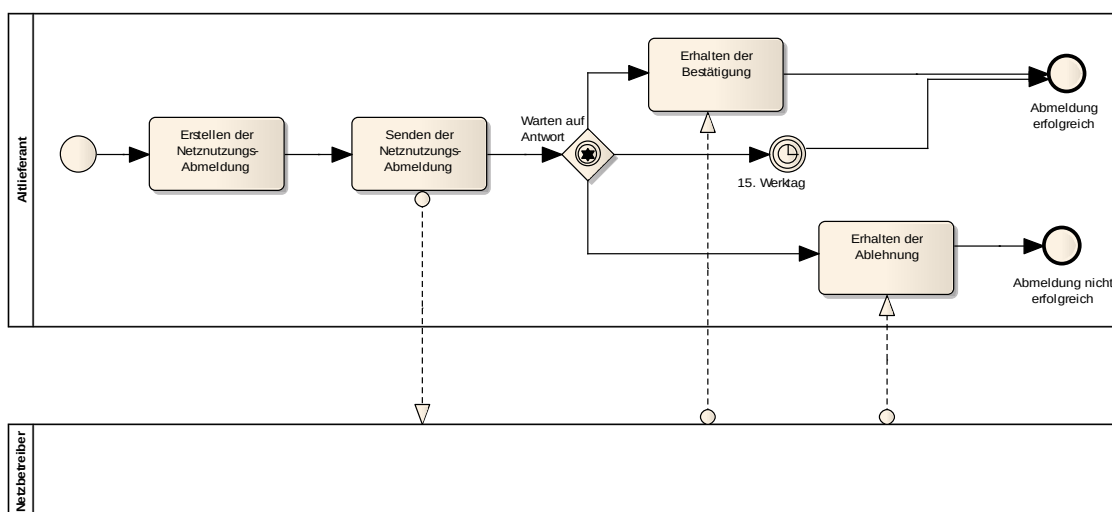


Abbildung 5.4: Independent Sub-Process "Abmeldung der Netznutzung"

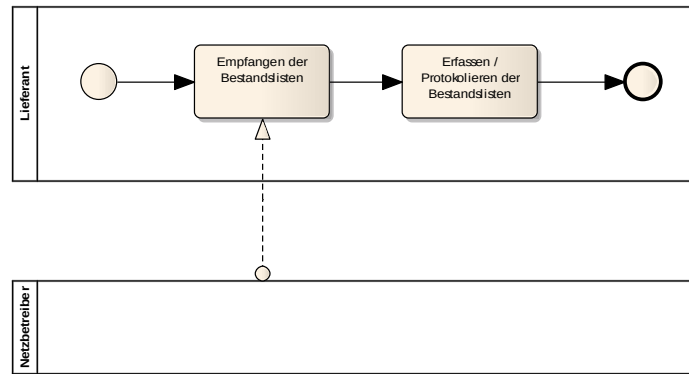


Abbildung 5.5: Independent Sub-Process "Erhalten der Bestandslisten"

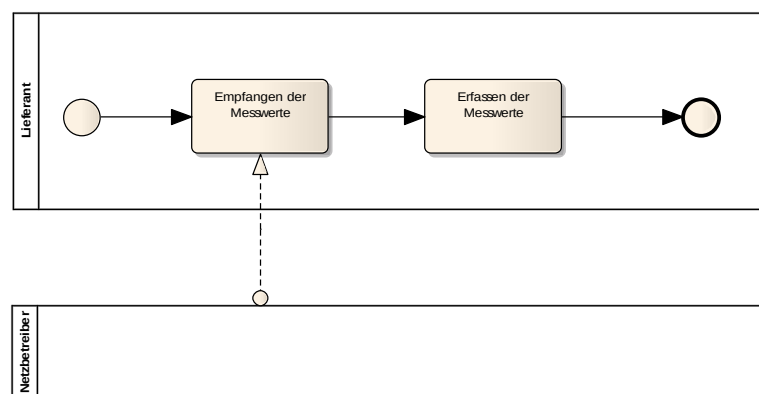


Abbildung 5.6: Independent Sub-Process „Messwerteübermittlung“

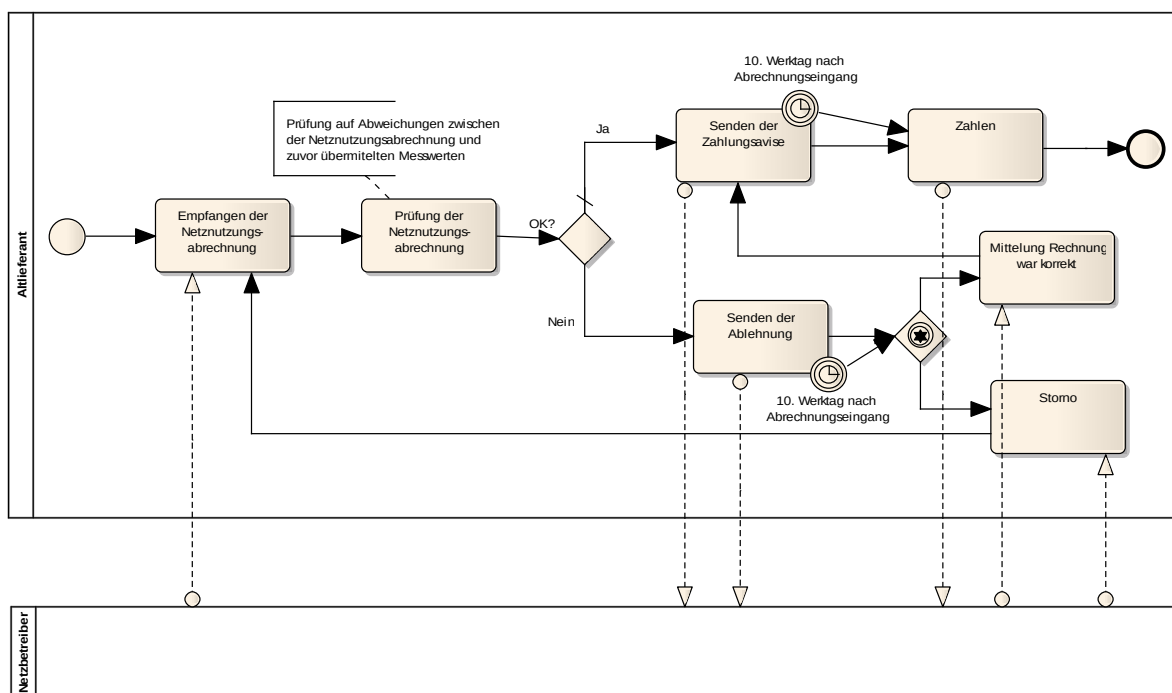


Abbildung 5.7: Independent Sub-Process " Netznutzungsabrechnung "

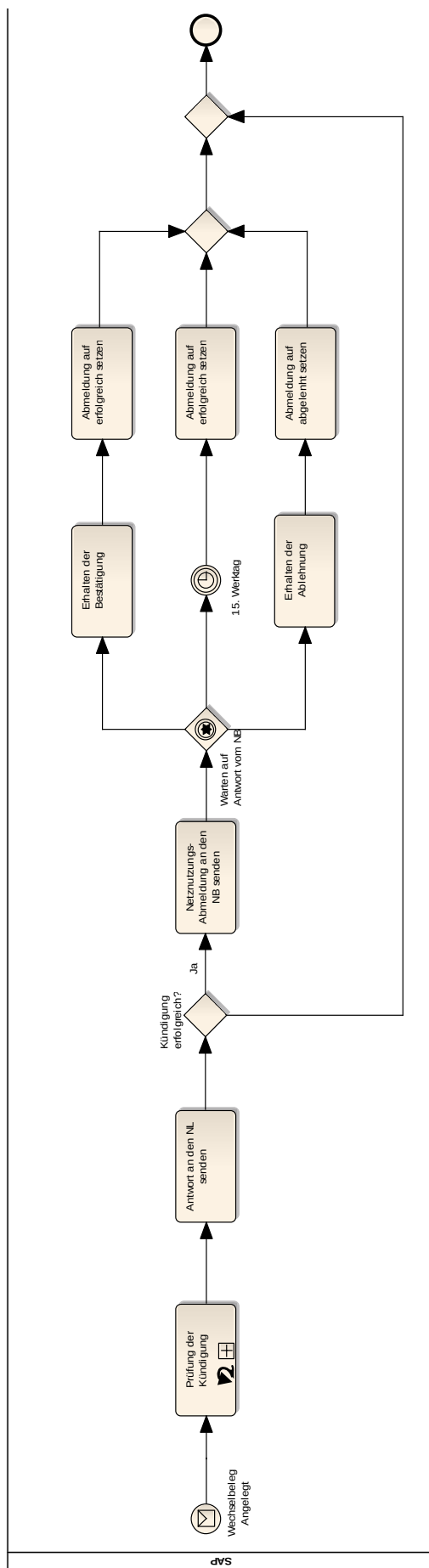


Abbildung 5.8: Workflow-Gerüst zum Geschäftsprozess „Lieferantenwechsel“

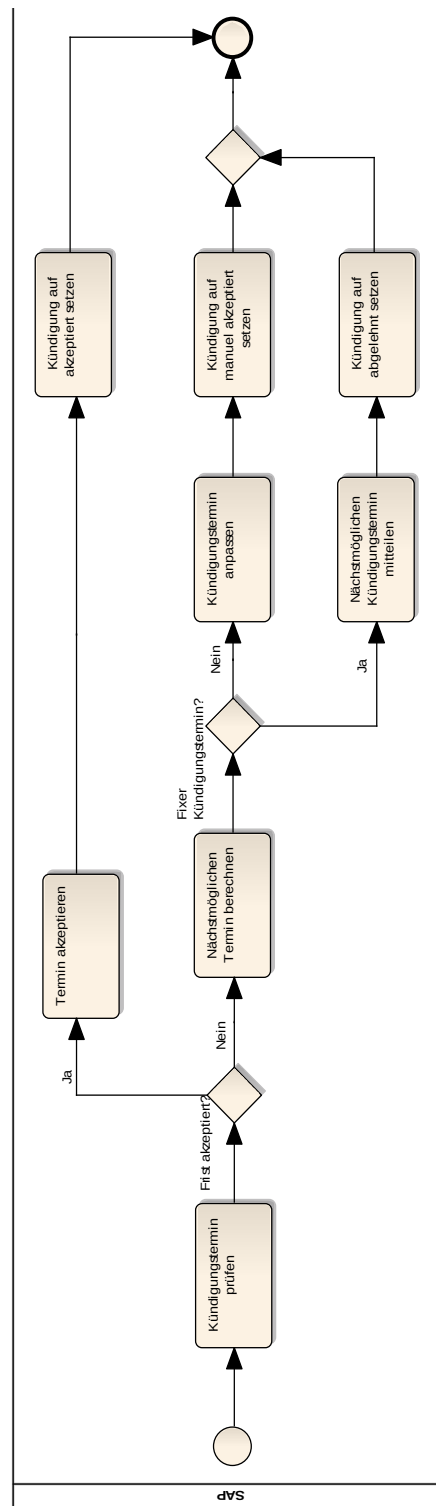


Abbildung 5.9: Workflow-Gerüst, Embedded Sub-Process "Prüfung der Kündigung"

In Abbildung 5.10 und Abbildung 5.11 wird eine mögliche Vorlage für den SAP Workflow dargestellt, die als Ergebnis der Transformation entstanden ist. Diese Abbildungen stammen allerdings nicht von einem automatisch transformierten Workflow. Da die Transformation des ganzen BPMN Modells mit Enterprise Architect, welche in der zweiten Diplomarbeit erarbeitet wurde, noch Schwierigkeiten bereitete, wurde dieser händisch im SAP Workflow Builder modelliert.

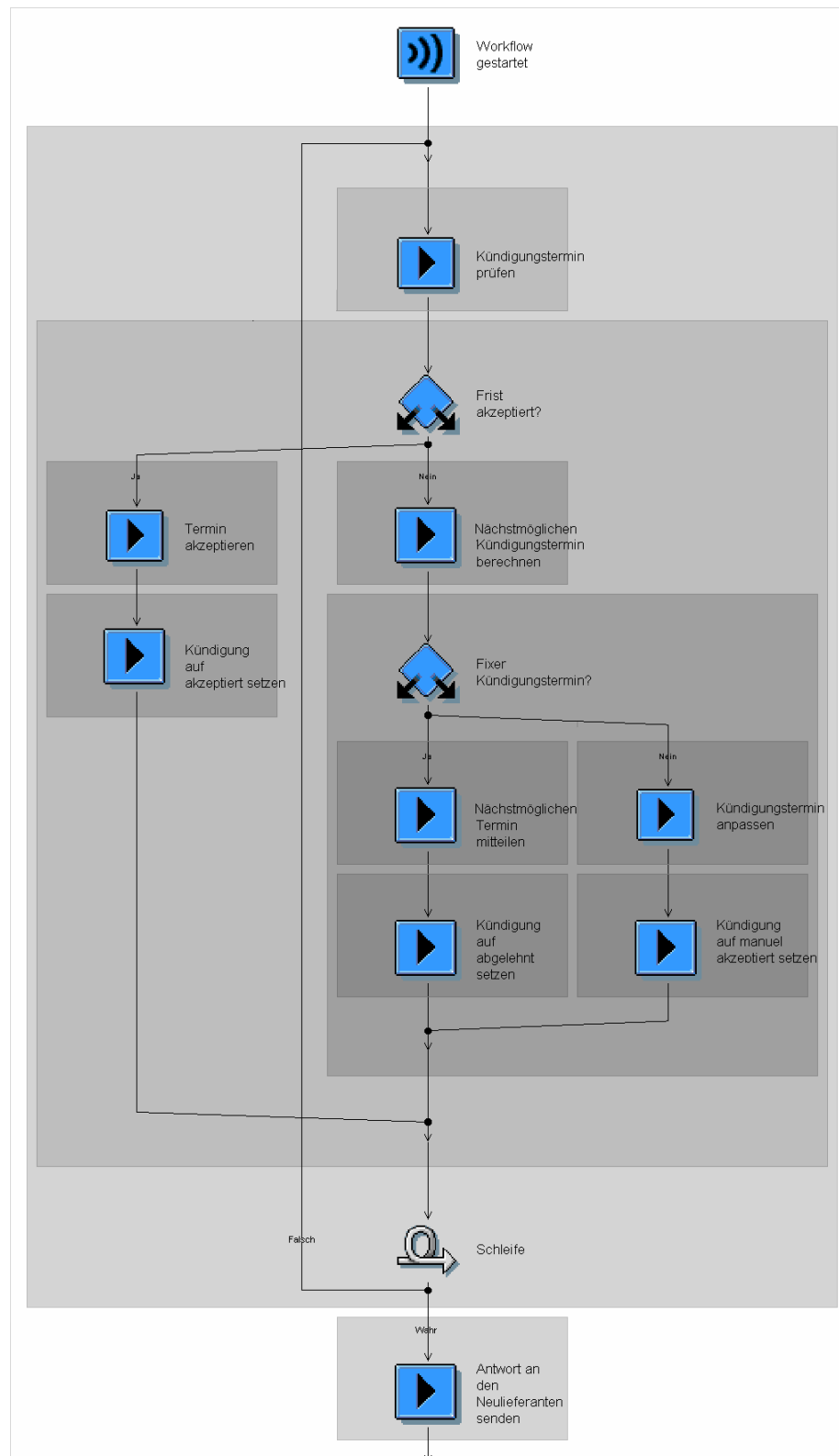


Abbildung 5.10: Vorlage für den SAP Workflow (Teil 1)

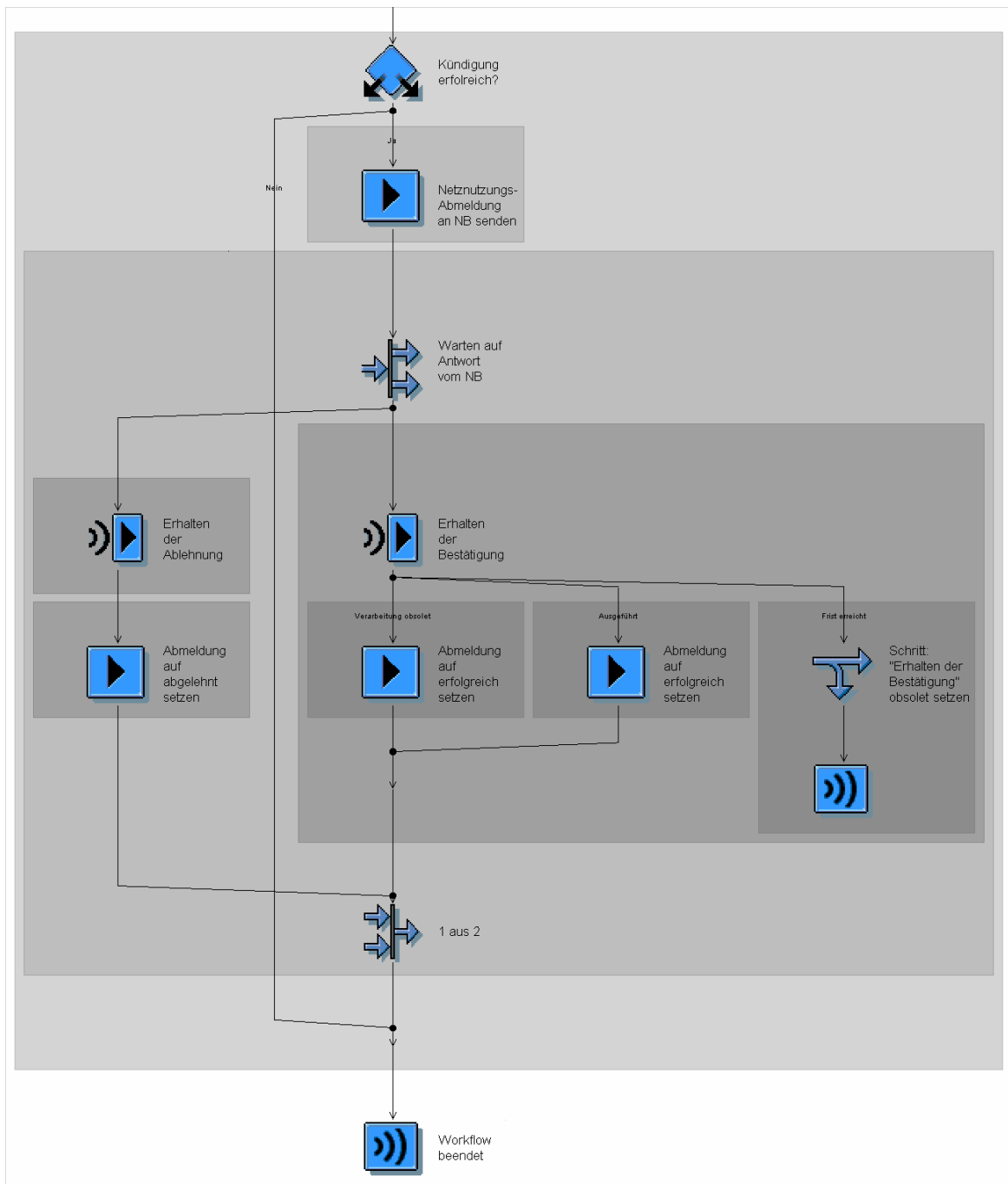


Abbildung 5.11: Vorlage für den SAP Workflow (Teil 2)

6 Bewertung und Ausblick

Im Rahmen dieser Arbeit wurde ein Konzept für die Ableitung eines SAP Workflows aus einem mittels BPMN modellierten Geschäftsprozesses ausgearbeitet. Zu diesem Zweck wurden einige von SWM vorgeschlagene Geschäftsprozesse mit BPMN auf zwei Detaillierungsebenen modelliert und verfeinert. Im Anschluss wurde aus dem Geschäftsprozessmodell ein Modell für das so genannte Workflow-Gerüst (ebenfalls in BPMN) abgeleitet. Dieses Workflow-Gerüst dient als Ausgangsbasis für die automatisierte Transformation nach SAP und bildet somit eine Vorlage für den ablauffähigen Workflow.

Dafür wurden – sowohl für die BPMN-Beschreibung als auch für den SAP Workflow – MOF-konforme Metamodelle erstellt und Transformationsregeln von dem BPMN^(SWM)-Metamodell zum SAP Workflow-Metamodell ausgearbeitet.

Während der Arbeit mit den beiden Notationen wurde festgestellt, dass sie sich doch sehr stark unterscheiden. Aus diesem Grund musste die BPMN eingeschränkt und um einige Elemente reduziert und an die Bedürfnisse des SAP Workflows angepasst werden. Die Tatsache, dass ein SAP Workflow blockstrukturiert ist, muss auch bei der Modellierung der Geschäftsprozesse und insbesondere bei der Herleitung des Workflow-Gerüsts berücksichtigt werden.

Dazu kommt auch, dass die interne Funktionalität von SAP mächtiger ist als die von BPMN. Außerdem ist sie nicht (nur) in der graphischen Notation vorhanden. Eine vollautomatische Erzeugung eines lauffähigen SAP Workflows ist daher aus BPMN unmöglich. Der vorgeschlagene Ansatz kann aber durchaus eine Vorlage für den SAP Workflow erzeugen, die dann in SAP verfeinert und mit interner SAP Funktionalität angereichert werden kann.

In der zukünftigen Weiterentwicklung dieses Ansatzes können die neuen Möglichkeiten der kommenden BPMN Spezifikation miteinbezogen werden. Auch die bisher ausgelassenen BPMN Elemente können nochmals im Hinblick auf die Wiederaufnahme in das Modell bewertet werden. SAP Entwicklungen schreiten ebenfalls regelmäßig voran, so dass auch von dieser Seite bessere Anknüpfungsmöglichkeiten geschaffen werden können.

Außerdem kann das Vorgehen für die Herleitung eines Workflow-Gerüsts aus dem Geschäftsprozessmodell noch detaillierter ausgearbeitet werden. Ein viel versprechender Ansatz für diese Problematik wurde von Allweyer in [All08a] beschrieben. Ein praktisches Beispiel, ebenfalls mit BPMN modelliert, ist in [All08b] zu finden.

Anhang

Abbildungsverzeichnis

Abbildung 2.1: Top-down-Prozessgestaltung (Quelle [FG08])	9
Abbildung 2.2: BPMN Prozess.....	13
Abbildung 2.3: Geschäftsprozessmodell Ebene 2	16
Abbildung 2.4: Geschäftsprozessmodell Ebene 3	17
Abbildung 2.5: Schritte, die sich auf eine Aktivität beziehen	20
Abbildung 2.6: Schritte, die der Ablauf-Steuerung dienen	20
Abbildung 2.7: Beispiel einer Workflow-Definition.....	21
Abbildung 2.8: BPMN Prozess nicht SAP-konform strukturiert	23
Abbildung 2.9: BPMN Prozess SAP-konform strukturiert	24
Abbildung 3.1: Hierarchie der Metamodellierung (Quelle [Wik09a]).....	25
Abbildung 3.2: Diagramm BPMN Core	28
Abbildung 3.3: Diagramm BPMN GraphicalObject	31
Abbildung 3.4: Diagramm BPMN Activity.....	33
Abbildung 3.5: Diagramm BPMN Event	37
Abbildung 3.6: Diagramm BPMN Gateway.....	41
Abbildung 3.7: Diagramm BPMN ConnectingObject.....	44
Abbildung 3.8: Diagramm BPMN Swimlane	46
Abbildung 3.9: Diagramm BPMN Artifact	48
Abbildung 3.10: Diagramm BPMN SupportingType.....	49
Abbildung 3.11: Diagramm SAP Core	51
Abbildung 3.12: Diagramm SAP Workflow-Definition.....	56
Abbildung 3.13: Diagramm SAP Schritt	60
Abbildung 3.14: Diagramm SAP Aktivität.....	64
Abbildung 3.15: Diagramm SAP Steuerungsschritt (Teil 1).....	68
Abbildung 3.16: Paralleler Abschnitt in der SAP Workflow-Definition.....	70
Abbildung 3.17: Diagramm SAP Steuerungsschritt (Teil 2).....	73
Abbildung 3.18: Block(Schritt) und LokalerWorkflow	78
Abbildung 4.1: Looping Activity in BPMN	82

Abbildung 4.2: Looping Activity mit TestTime = After als Objektdiagramm.....	82
Abbildung 4.3: Looping Activity mit TestTime = After transformiert zu SAP als Objektdiagramm	83
Abbildung 4.4: Looping Activity mit TestTime = After transformiert zu SAP als Schleife in der Workflow-Definition.....	83
Abbildung 4.5: Looping Activity mit TestTime = Before als Objektdiagramm	84
Abbildung 4.6: Looping Activity mit TestTime = Before transformiert zu SAP als Objektdiagramm	84
Abbildung 4.7: Looping Activity mit TestTime = Before transformiert zu SAP als Schleife in der Workflow-Definition.....	85
Abbildung 4.8: Link StartEvent in BPMN	88
Abbildung 4.9: Link StartEvent als Objektdiagramm	89
Abbildung 4.10: Link StartEvent transformiert zu SAP Workflow als Objektdiagramm	89
Abbildung 4.11: Ereigniserzeuger in der Workflow-Definition.....	89
Abbildung 4.12: Terminüberwachung in BPMN	93
Abbildung 4.13: Terminüberwachung in BPMN als Objektdiagramm.....	93
Abbildung 4.14: Modellierter Terminüberwachung im SAP Workflow als Objektdiagramm	94
Abbildung 4.15: Modellierter Terminüberwachung im SAP Workflow	94
Abbildung 5.1: Geschäftsprozess „Lieferantenwechsel“ 2. Ebene.....	102
Abbildung 5.2: Independent Sub-Process "Kündigung des Gasliefervertrages"	103
Abbildung 5.3: Embedded Sub-Process "Prüfung der Kündigung"	103
Abbildung 5.4: Independent Sub-Process "Abmeldung der Netznutzung"	103
Abbildung 5.5: Independent Sub-Process "Erhalten der Bestandslisten"	104
Abbildung 5.6: Independent Sub-Process „Messwerteübermittlung“	104
Abbildung 5.7: Independent Sub-Process " Netznutzungsabrechnung "	104
Abbildung 5.8: Workflow-Gerüst zum Geschäftsprozess „Lieferantenwechsel“	105
Abbildung 5.9: Workflow-Gerüst, Embedded Sub-Process "Prüfung der Kündigung"	106
Abbildung 5.10: Vorlage für den SAP Workflow (Teil 1).....	107
Abbildung 5.11: Vorlage für den SAP Workflow (Teil 2).....	108

Tabellenverzeichnis

Tabelle 2.1: BPMN Grundkonstrukte	12
Tabelle 4.1: Transformation vom BPMNObject	80
Tabelle 4.2: Transformation vom BusinessProcess und Diagram	80
Tabelle 4.3: Activity Transformation, LoopType = None	81
Tabelle 4.4: Activity Transformation, LoopType = Standard	81
Tabelle 4.5: Activity Transformation, LoopType = MultiInstance	85
Tabelle 4.6: Activity Transformation: Task	86
Tabelle 4.7: Activity Transformation: Sub-Process	87
Tabelle 4.8: StartEvent Transformation.....	88
Tabelle 4.9: EndEvent Transformation.....	90
Tabelle 4.10: Transformation von IntermediateEvent im SequenceFlow	91
Tabelle 4.11: Transformation von IntermediateEvent am Rand einer Activity	91
Tabelle 4.12: Transformation vom SplittingGateway, GatewayType = XOR, XORType = Data.....	95
Tabelle 4.13: Transformation vom SplittingGateway, GatewayType = XOR, XORType = Event.....	96
Tabelle 4.14: Transformation vom SplittingGateway, GatewayType = AND.....	97
Tabelle 5.1: Beschreibung des Prozesses "Lieferantenwechsel" (Quelle [GeLi07]).....	98

Abkürzungsverzeichnis und Glossar

ABAP	Advanced Business Application Programming – proprietäre SAP Programmiersprache.
BPEL	Business Process Execution Language. ist eine XML-basierte ausführbare Sprache zur Beschreibung von Geschäftsprozessen.
BPDM	Business Process Definition Metamodel. Ein einheitliches Metamodell zur Beschreibung von Geschäftsprozessen.
BPMN	Business Process Modelling Notation. Graphische Notation zur Modellierung der Geschäftsprozesse.
EA	Enterprise Architect. Ein umfassendes UML Modellierungswerkzeug. Unterstützt auch BPMN.
EPK	Ereignisgesteuerte Prozessketten. Graphische Notation zur Modellierung der Geschäftsprozesse.
MOF	Meta Object Facility. Eingeführt von der OMG und ist zum Standard bei der Definition von Metamodellen geworden.
OMG	Object Management Group. Ein internationales Konsortium, welches Standards für die Objektorientierte Programmierung entwickelt.
SAP	Systeme, Anwendungen und Produkte in der Datenverarbeitung. Die SAP AG ist der größte europäische und weltweit viertgrößte Softwarehersteller.
SAP ERP	Enterprise-Resource-Planning. SAP ERP ist das Hauptprodukt der SAP AG, Nachfolger des bekannten SAP R/3. ERP ist ein Unternehmens-Informationssystem, das entwickelt wurde, um alle geschäftsrelevanten Bereiche eines Unternehmens im Zusammenhang zu betrachten.
SWM	Stadtwerke München GmbH. Der größte deutsche kommunale Dienstleister.
UML	Unified Modeling Language. Vereinheitlichte Modellierungssprache.

Literaturverzeichnis

- [All08a] Allweyer, Thomas: Erzeugung detaillierter und ausführbarer Geschäftsprozessmodelle durch Modell-zu-Modell-Transformationen
<http://kurze-prozesse.de/blog/wp-content/uploads/2007/12/m2m-transformationen-allweyer-epk2007.pdf> - zuletzt besucht am 31.01.2009.
- [All08b] Allweyer, Thomas: Vom fachlichen Modell zum ausführbaren Workflow. Am Beispiel von ARIS und Intalio|BPMS. 2008.
<http://kurze-prozesse.de/blog/wp-content/uploads/2008/02/Vom-fachlichen-Modell-zum-Workflow.pdf> - zuletzt besucht am 31.01.2009.
- [BPMNa] BPMN auf gut Deutsch. 2008
http://kurze-prozesse.de/?page_id=78 - zuletzt besucht am 31.01.2009.
- [BPMN10] Business Process Modeling Notation (BPMN) Specification. Final Adopted Specification.
<http://www.bpmn.org/Documents/OMG%20Final%20Adopted%20BPMN%201-0%20Spec%2006-02-01.pdf> - zuletzt besucht am 31.01.2009
- [BPMN11] Business Process Modeling Notation, V1.1. OMG Available Specification.
<http://www.bpmn.org/Documents/BPMN%201-1%20Specification.pdf> - zuletzt besucht am 01.01.2009
- [ConS] ConSense IMS | QMS | PMS.
<http://www.consense-gmbh.de/software/consense-ims-qms-pms.html> - zuletzt besucht am 31.01.2009
- [EntA71] Enterprise Architect. Programmversion 7.1.
<http://www.sparxsystems.com/products/ea/index.html>. - zuletzt besucht am 31.01.2009.
- [FG08] Freund, Jakob; Götzer, Klaus: Vom Geschäftsprozess zum Workflow. Ein Leitfaden für die Praxis. November 2008. ISBN 978-3-446-41482-2
- [Fre08] Freund, Jakob: Prozessmodelle verfeinern kein Problem? Oktober 2008.
<http://www.bpmn.info/?p=44> - zuletzt besucht am 06.02.2009
- [Gad05] Gadatsch, Andreas: Grundkurs Geschäftsprozess-Management. Methoden und Werkzeuge für die IT-Praxis: Eine Einführung für Studenten und Praktiker. 4., erweiterte Auflage. September 2005 - ISBN 3-8348-0039-2
- [GeLi07] Geschäftsprozesse Lieferantenwechsel Gas (GeLi Gas). Anlage zu dem Beschluss BK7-06-067 vom 20. August 2007.
<http://www.bundesnetzagentur.de/media/archive/11200.pdf> - zuletzt besucht am

20.12.2008

- [OAS07] OASIS Web Services Business Process Execution Language (WSBPEL) TC. Standard 2007
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel. - zuletzt besucht am 31.01.2009.
- [OMG06] OMG: Business Process Definition Metamodel. Stand 2006.
<http://www.omg.org/cgi-bin/doc?bei/03-01-06>
- [OMG07] OMG: Unified Modeling Language (OMG UML) V2.1.2. Stand 2007.
<http://www.omg.org/spec/UML/2.1.2/Infrastructure/PDF/> - zuletzt besucht am 31.01.2009
- [OMG09] OMG: MetaObject Facility.
<http://www.omg.org/mof/> - zuletzt besucht am 31.01.2009.
- [SAPBF] SAP Bibliothek - SAP Business Workflow.
http://help.sap.com/saphelp_nw04/helpdata/de/0f/80243b4a66ae0ce10000000a11402f/frameset.htm - zuletzt besucht am 31.01.2009
- [Sil07] Silver, Bruce: BPMN and the Business Process Expert, Part 1 bis Part 6 2007.
<http://www.brsilver.com/wordpress/category/bpmn/>
- [SWM09] Stadtwerke München GmbH
<http://www.swm.de/> - zuletzt besucht am 06.02.2009
- [Wik09a] Wikipedia: Meta Object Facility.
http://de.wikipedia.org/wiki/Meta_Object_Facility - zuletzt besucht am 31.01.2009.
- [Wik09b] Wikipedia: Modelltransformation
<http://de.wikipedia.org/wiki/Modelltransformation> - zuletzt besucht am 31.01.2009.
- [WM08] White, Stephen A. PhD; Miers, Derek: BPMN Modeling and Reference Guide: Understanding and Using BPMN. September 2008. ISBN 978-097775272-0