

Institut für Architektur von Anwendungssystemen
Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Diplomarbeit Nr. 3237

Ein Modellierungswerkzeug für Produktionsprozesse auf Basis einer BPMN-Erweiterung

Monir Zouaoui

Studiengang: Informatik

Prüfer: Prof. Dr. Frank Leymann

Betreuer: Dipl.-Inf. Sema Zor

Dipl.-Inf. Uwe Breitenbücher

Dipl.-Inf. Oliver Kopp

Dipl.-Inf. Daniel Schleicher

begonnen am: 1. September 2011

beendet am: 2. März 2012

CR-Klassifikation: D.2.11, H.4.1

Kurzfassung

Um die Business Manufacturing Gap zu schließen, sollten auch Produktionsunternehmen in einer Service Oriented Architecture kollaborieren können. Ein Teil des BPM Lebenszyklus ist die Phase Modellierung. Zur Umsetzung der Phase Modellierung wird in dieser Arbeit ein Modellierungswerkzeug für Produktionsprozesse auf Basis einer BPMN-Erweiterung vorgestellt.

Das Konzept der Erweiterung BPMN4Manu erfüllt die formalen Bedingungen an eine Erweiterung der BPMN. Es werden neue Symbole eingeführt und die statische Struktur des zusätzlichen Metamodells als UML-Klassendiagramme dargestellt. Die direkten und die indirekten Beziehungen sind in Form von Matrizen aufgeführt. Die Umsetzung erfolgt im Modellierungswerkzeug Oryx.

Abstract

For closing the business manufacturing gap manufacturing companies aim to collaborate in a service oriented architecture. One part of the BPM lifecycle is the phase modeling. To implement the phase modeling this thesis presents a modeling tool for production processes on the basis of a BPMN extension.

The concept of the extension BPMN4Manu achieves the formal restrictions on an extension of BPMN. New symbols are introduced and the static structure of the additional metamodel are represented as UML class diagrams. The direct and indirect relationships are given in the form of matrices. The implementation takes place in the modeling tool Oryx.

Inhaltsverzeichnis

1. Modellierung von Produktionsprozessen	9
1.1. Motivation	9
1.2. Problemstellung	11
1.3. Aufbau der Arbeit	12
2. Szenario: Prüfen einer Zündkerze	13
3. Business Process Model and Notation (BPMN)	15
3.1. Aufbau der Spezifikation	15
3.2. Trennung von Notation und Metamodel	16
3.3. Erweiterbarkeit	18
4. Konzept der Erweiterung BPMN4Manu	21
4.1. Notation	21
4.2. Metamodel	24
4.3. Beziehungen	27
4.4. Kritische Betrachtung	29
5. Umsetzung im Modellierungswerkzeug Oryx	31
5.1. Einführung in Oryx	31
5.2. Notation	32
5.3. Metamodel und Beziehungen	32
6. Erweitertes Szenario	37
7. Verwandte Arbeiten	39
8. Zusammenfassung und Ausblick	41
Literaturverzeichnis	43
A. Anhang	49
A.1. README.txt	49
A.2. Oryx Setup Guide	57
A.3. Tests	68

Abbildungsverzeichnis

1.1.	BPM Lifecycle	9
2.1.	Prüfen einer Zündkerze in BPMN	14
3.1.	BPMN Core und Layer Structure	15
3.2.	Trennung von Notation und Metamodel sowie Verfeinerung	16
3.3.	Extension Class Diagram	19
4.1.	Sequence Flow Metamodel	24
4.2.	Machines & Tools Flow Metamodel	25
4.3.	Parts Flow Metamodel	26
5.1.	Architektur von Oryx	31
5.2.	Laden der Stencil Set Extension BPMN4Manu	33
6.1.	Test der containmentRules	37
6.2.	Prüfen einer Zündkerze in BPMN4Manu innerhalb von Oryx	38
A.1.	Test der Sequence Flow connectionRules	68
A.2.	Test der Machines & Tools Flow connectionRules	69
A.3.	Test der Parts Flow connectionRules	69

Tabellenverzeichnis

4.1.	Symbole der Erweiterung BPMN4Manu	21
4.2.	Symbole für Werkzeuge und Einzelteile	23
4.3.	Direkte Beziehungen aller Elemente der Erweiterung BPMN4Manu	27
4.4.	Sequence Flow Beziehung	28
4.5.	Machines & Tools Flow Beziehung	28
4.6.	Parts Flow Beziehung	28

Verzeichnis der Listings

5.1.	Registrierung der Stencil Set Extension BPMN4Manu	33
5.2.	Definition des Stencils ManuTask	34
5.3.	Definition der roles des Stencils ManuTask	34
5.4.	Definition der connectionRules	35
5.5.	Definition der containmentRules	35

Abkürzungsverzeichnis

BPDM	Business Process Definition MetaModel
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Model and Notation
BPMS	Business Process Management System
BPR	Business Process Reengineering
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CSF	Critical Success Factor
ERP	Enterprise Resource Planning
FMC	Fundamental Modeling Concepts
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LTL	Linear Temporal Logic
MDA	Model Driven Architecture
ME	Manufacturing Execution
MOF	Meta Object Facility
MTM	Methods Time Measurement
OMG	Object Management Group
PNG	Portable Network Graphics
PPS	Produktionsplanung und -steuerung
SCM	Supply Chain Management
SOA	Service Oriented Architecture
SVG	Scalable Vector Graphics
UML	Unified Modeling Language
VSM	Value Stream Mapping
WfMS	Workflow Management System
XML	Extensible Markup Language
XPDL	XML Process Definition Language
XSD	XML Schema Definition

1. Modellierung von Produktionsprozessen

1.1. Motivation

Eine der Aufgaben von Unternehmen ist die Organisation der wertschöpfenden Prozesse. *Business Process Management* (BPM) [EHLB95] konzentriert sich auf die in den Parametern Durchlaufzeit, Qualität, Wirtschaftlichkeit und Flexibilität optimale Gestaltung dieser Prozesse. Hierbei werden alle Entscheidungen weitestgehend nach dem ökonomischen Prinzip gefällt [Töpo07]. Es wird kontinuierlich das Ziel verfolgt, eine Reihe von *Critical Success Factors* (CSF) zu erfüllen. Im Sinne des klassischen *Business Process Reengineering* (BPR) [HC93], werden die Prozesse fortlaufend identifiziert, analysiert/synthetisiert und implementiert.

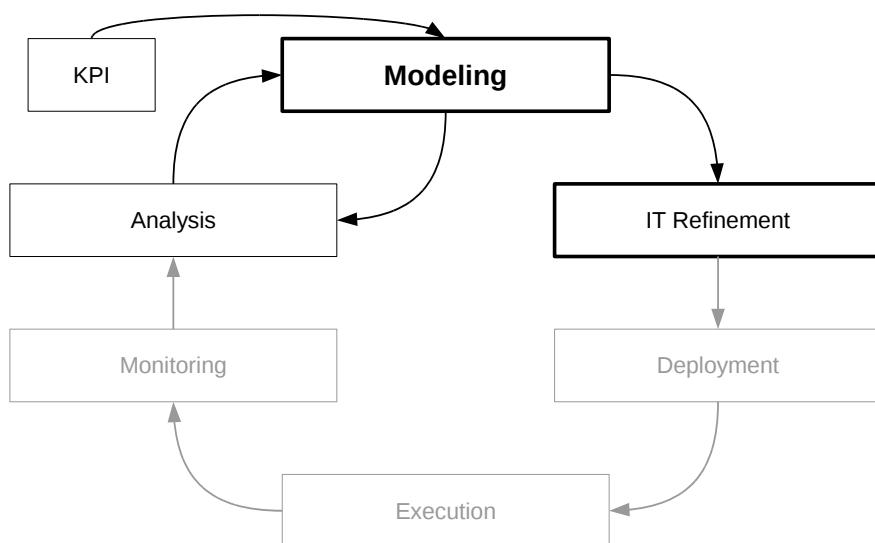


Abbildung 1.1.: BPM Lifecycle [Ley10]

Der *BPM Lifecycle* [Ley10] (Abbildung 1.1) ist ein Prozessregelkreis mit den Phasen *Modeling*, *IT Refinement*, *Deployment*, *Execution*, *Monitoring* und *Analysis* sowie mit *Key Performance Indicators* (KPI). Die Problemstellung (Abschnitt 1.2) dieser Arbeit bewegt sich hauptsächlich innerhalb der Phase *Modeling*.

1. Modellierung von Produktionsprozessen

Während dem **Modeling** entsteht das reproduzierbare Abbild eines Prozesses – ein **Prozessmodell**. Das Prozessmodell identifiziert die Güter und Dienstleistungen eines Unternehmens („*Process Model = Product*“ [Ley10]). Es wird in ein ausführbares *Workflow Model* [WFM95] transformiert und so die Kluft zwischen Business und IT, die *Business IT Gap*, geschlossen. Um die Erfüllung der CSF sicherzustellen, realisiert die Phase Analysis die Messung sowie Aus- und Bewertung einer Reihe von KPI.

Im globalisierten Markt sind Wertschöpfungsprozesse typischerweise über die Grenzen eines Unternehmens hinweg miteinander verzahnt („*Virtual Enterprises*“ [LRoo]). Um die Prozesse dennoch optimal gestalten zu können, verwenden die Unternehmen *Business Process Management Systems* (BPMS) sowie *Workflow Management Systems* (WfMS), orchestrieren *Web Services* und kollaborieren in einer *Service Oriented Architecture* (SOA) [WCL⁺08].

In Produktionsunternehmen sind die wertschöpfenden Prozesse vorwiegend **Fertigungs- und Produktionsprozesse** [BBD⁺08]. Unter dem Einsatz von Produktionsfaktoren [Gut83] werden aus mehr oder weniger komplexen Inputgütern wertgesteigerte Outputgüter erzeugt [GT12]. Zählt Information (einschließlich dem Umfeld und der Umwelt) zu den Potentialfaktoren [Bod93], bedingt die optimale Gestaltung der Prozesse den effizienten Einsatz von Informationssystemen für betriebswirtschaftliche und technische Aufgaben. Betriebswirtschaftliche Aufgaben sind z. B. *Enterprise Resource Planning* (ERP), *Supply Chain Management* (SCM) sowie *Produktionsplanung und -steuerung* (PPS). Technische Aufgaben sind z. B. *Computer Aided (Design | Engineering | Manufacturing)* (CA(D|E|M)) sowie *Manufacturing Execution* (ME) [SHo4, VWB⁺09].

Produktionsunternehmen befinden sich in einem turbulenten Umfeld. Die Prozesse müssen stetig flexibler [SGNWP08] und wandlungsfähiger [WZ09] werden. Die Kluft zwischen den Informationssystemen und dem *Production Shop Floor* [RS99], die *Business Manufacturing Gap* [Gif07], verhindert die optimale Gestaltung der Prozesse und muss daher geschlossen werden. Auch Produktionsunternehmen sollen in einer SOA kollaborieren können [ICCo8].

Für die Modellierung von Geschäftsprozessen hat sich die *Business Process (Modeling | Model and) Notation* (**BPMN V(1.0 | 1 | (1.2 | 2.0))**) [OMG06, OMG08c, OMG09, OMG11a] durchgesetzt. Die BPMN¹ zeichnet sich durch den Aufbau der Spezifikation sowie durch die konsequente Trennung von *Notation* und *Metamodel* aus.

Die Spezifikation der BPMN definiert einen *BPMN Core* und eine *Layer Structure*. Der BPMN Core ist grundlegend, präzise und robust. Die Layer Structure macht die BPMN flexibel, transparent und (abwärts)kompatibel.

¹Zum Zeitpunkt der Erstellung dieser Arbeit lag die BPMN in der Version 2.0 vor. Ist die Version nicht ausdrücklich angegeben, bedeutet **BPMN** immer **Business Process Model and Notation Version 2.0** [OMG11a].

Die konsequente Trennung von Notation und Metamodel ist das Ergebnis von Bestrebungen der *Object Management Group* (OMG), die BPMN besser mit der *Model Driven Architecture* (MDA) [OMG03] in Einklang zu bringen. Es kann für spezielle fachliche Interessengruppen sowie für unterschiedliche Aufgaben mit differenzierterem Detaillierungsgrad modelliert werden. Das Metamodel der BPMN ist erweiterbar, austauschbar und bestimmt eine eindeutige sowie verständliche(re) Ausführungssemantik.

Durch die erhöhte Qualität der Notation und die Erweiterbarkeit, kommt die BPMN prinzipiell für die Modellierung von Produktionsprozessen in Frage. Die Vereinigung der gegensätzlichen Eigenschaften Robustheit und Flexibilität, macht sie schließlich zur Sprache der Wahl.

1.2. Problemstellung

Damit Produktionsunternehmen in einer SOA kollaborieren können, muss die Business Manufacturing Gap geschlossen werden.

Hierzu muss eine Erweiterung der BPMN zur Modellierung von Produktionsprozessen präsentiert werden. Zor et al. [ZLS11] schlagen *BPMN4Manu* vor. Die Erweiterung muss zunächst konzipiert und in einem Modellierungswerkzeug umgesetzt werden.

In einem weiteren Schritt sollte ein Produktionsprozessmodell dann alle fachlichen Details sowie die für ein BPMS zusätzlich notwendigen, technischen Details exakt abbilden können. Schließlich sollte die Transformation einschließlich der Erweiterung BPMN4Manu in ein ausführbares Workflow Model in BPEL umgesetzt werden.

1. Modellierung von Produktionsprozessen

1.3. Aufbau der Arbeit

Kapitel 1 – Modellierung von Produktionsprozessen

Kapitel 1 motiviert die Kollaboration von Produktionsunternehmen in einer SOA und stellt die schrittweise Problemstellung zur Schließung der Business Manufacturing Gap dar.

Kapitel 2 – Szenario: Prüfen einer Zündkerze

Kapitel 2 konkretisiert anhand eines beispielhaften Prozessmodells die Notwendigkeit von BPMN4Manu.

Kapitel 3 – Business Process Model and Notation (BPMN)

Kapitel 3 stellt die Vorteile des Aufbaus der Spezifikation, der konsequenten Trennung von Notation und Metamodel sowie der Erweiterbarkeit von BPMN grundlegend dar.

Kapitel 4 – Konzept der Erweiterung BPMN4Manu

Kapitel 4 beschreibt die Erweiterung von Notation und Metamodel sowie die Beziehungen der neuen Elemente unter Beachtung der allgemeinen und formalen Bedingungen. Hierzu werden die neuen Symbole eingeführt, die Erweiterung des Metamodells in UML dargestellt und die Beziehungen in Form von Matrizen aufgeführt. Das Ende dieses Kapitels enthält eine kritische Betrachtung.

Kapitel 5 – Umsetzung im Modellierungswerkzeug Oryx

Kapitel 5 beschreibt die Umsetzung des Konzepts der Erweiterung BPMN4Manu. Hierzu wird die Architektur des Modellierungswerkzeugs Oryx, die Realisierung und Bereitstellung der Notation wie auch die Definition des Metamodells und der Beziehungen erklärt.

Kapitel 6 – Erweitertes Szenario

Kapitel 6 verifiziert anhand eines beispielhaften Prozessmodells die Umsetzung des Konzepts der Erweiterung BPMN4Manu.

Kapitel 7 – Verwandte Arbeiten

Kapitel 7 stellt Arbeiten mit einer Erweiterung der BPMN V(1.(0|1)|(1.2|2.0)) sowie mit einer Umsetzung im Modellierungswerkzeug Oryx vor.

Kapitel 8 – Zusammenfassung und Ausblick

Kapitel 8 fasst die Ergebnisse zusammen und bietet einen Ausblick auf die weiteren Schritte zur Schließung der Business Manufacturing Gap.

2. Szenario: Prüfen einer Zündkerze

Das im Folgenden beschriebene, beispielhafte Prozessmodell in BPMN dient zur Konkretisierung der Notwendigkeit von *BPMN4Manu*.

Die Abbildung 2.1 zeigt das Prozessmodell der Prüfung einer Zündkerze in BPMN. Es ist das Abbild eines mit *Methods Time Measurement* (MTM) [Deu] analysierten Prozesses.

Die einzelnen Aktivitäten sind die Folgenden:

1. *Open and remove the closure cap.* 1. **Verschlusskappe** öffnen und abnehmen.
2. *Remove nuts with a wrench.* 2. **Muttern** mit einer **Ratsche** entfernen.
3. *Remove cover.* 3. **Abdeckung** abnehmen.
4. *Remove ignition fuse and disconnect ignition cable.* 4. **Zündkabelsicherung** öffnen und **Zündkabel** abziehen.
5. *Take screw out, use screwdriver.* 5. **Schraube** mit einem **Schraubenzieher** herausdrehen.
6. *Pull out ignition coil.* 6. **Zündspule** herausziehen.
7. *Unscrew the ignition plug, use wrench.* 7. **Zündkerze** mit einer **Ratsche** herausdrehen.
8. *Check ignition plug.* 8. **Zündkerze** prüfen.

Die weiteren Aktivitäten in Abbildung 2.1 bilden die Aktivitäten 1-8 in umgekehrter Reihenfolge ab.

Die **hervorgehobenen Begriffe** sind die während der jeweiligen Aktivität verwendeten **Maschinen**, **Werkzeuge** und **Einzelteile**. Sie können in BPMN nicht abgebildet werden, denn es existieren keine Symbole und Elemente. Zur Abbildung des **Wertstroms** existieren in BPMN auch keine Symbole und Elemente. Dies ist jedoch zur optimalen Gestaltung eines Produktionsprozesses im Sinne des *Value Stream Mapping* (VSM) [RS99] von entscheidender Bedeutung.

Zor et al. [ZLS11] schlagen zur Abbildung von Maschinen, Werkzeugen, Einzelteilen und Wertströmen die Erweiterung **BPMN4Manu** vor.

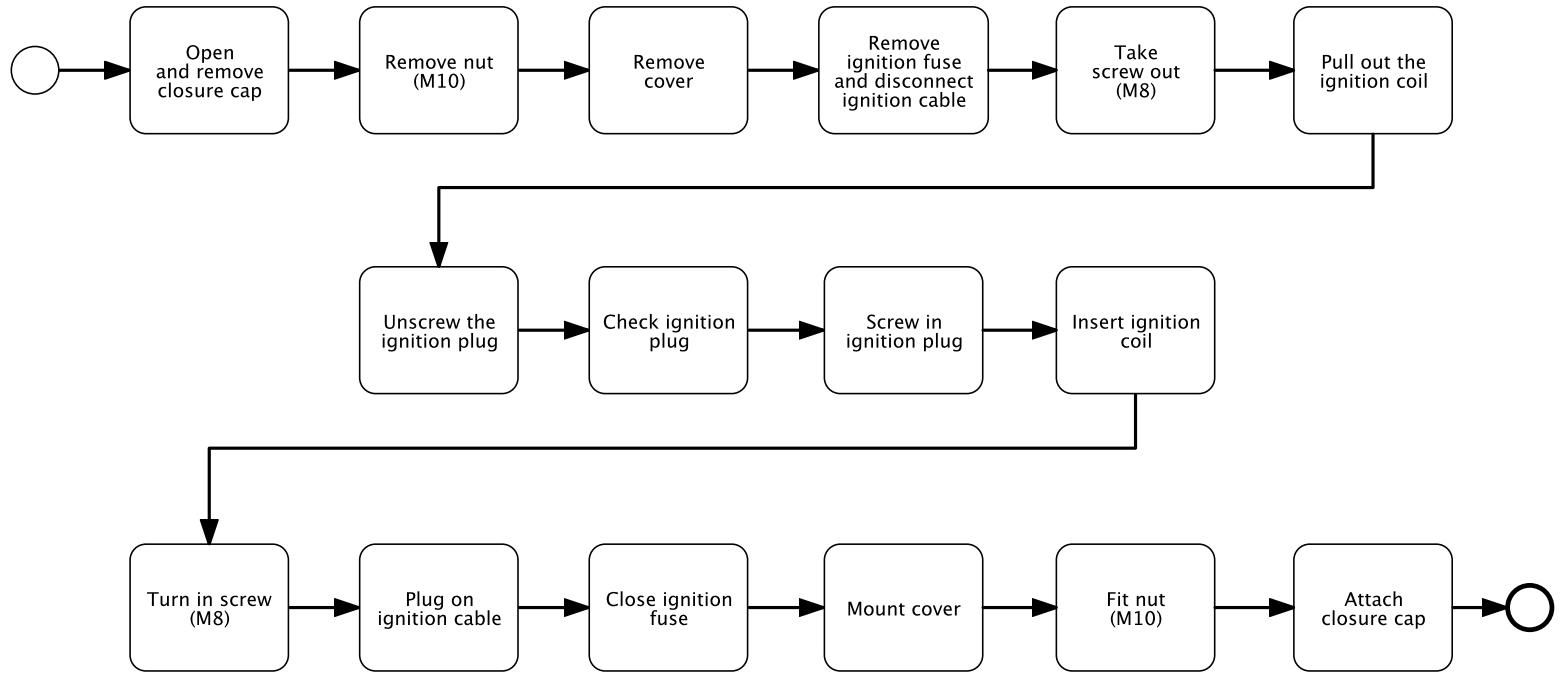


Abbildung 2.1.: Prüfen einer Zündkerze in BPMN [ZLS11]

3. Business Process Model and Notation (BPMN)

Wie in der Motivation (Abschnitt 1.1) erwähnt, hat sich die BPMN für die Modellierung von Geschäftsprozessen durchgesetzt. Sie zeichnet sich durch den **Aufbau der Spezifikation** sowie durch die konsequente **Trennung von Notation und Metamodell** aus. An eine Erweiterung der BPMN werden **allgemeine** und **formale Bedingungen** gestellt. Zur Bearbeitung des ersten Schritts der Problemstellung (Abschnitt 1.2) sind in Unterabschnitt 3.3.2 die formalen Bedingungen an eine Erweiterung des Metamodells der BPMN dargestellt.

3.1. Aufbau der Spezifikation

Die Abbildung 3.1 zeigt den Aufbau der Spezifikation der BPMN mit dem **BPMN Core** und der **Layer Structure**.

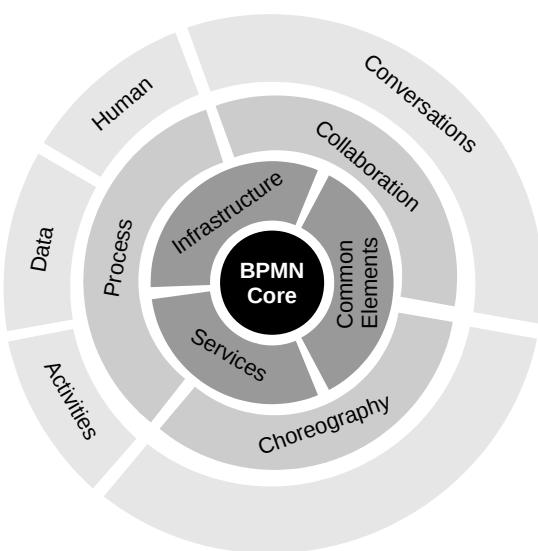


Abbildung 3.1.: BPMN Core und Layer Structure [OMG11a]

Der **BPMN Core** ist grundlegend, präzise und robust. Er enthält alle zur Modellierung von *Processes (orchestrations)*, *Choreographies (interaction)* und *Collaborations (interconnection)* notwendigen Konstrukte.

3. Business Process Model and Notation (BPMN)

Der BPMN Core ist in die folgenden *Packages* unterteilt:

1. **Foundation** Die grundlegenden Konstrukte der BPMN und Erweiterbarkeit.
2. **Common** Für Processes, Choreographies und Collaborations gemeinsame Konstrukte.
3. **Service** Konstrukte von Schnittstellen und Diensten.

Ein weiteres BPMN Core Package, das Package 4. **Infrastructure**, enthält die Klassen Definitions und Import. Die Implementierung dieser Klassen ist für die Modellierung in BPMN sowie für die Realisierung der Austauschbarkeit unerlässlich. Sie legen die *Concrete Syntax* und die *Abstract Syntax* als XML Schema Definition (XSD) [Wora] fest. Durch das Infrastructure Package hebt sich die BPMN V2.0 [OMG11a] deutlich von der BPMN V(1.(0|1)) [OMG06, OMG08c] ab.

Durch die **Layer Structure** kann die Object Management Group (OMG) die Schichten und Packages auf flexible Art und Weise zueinander in Beziehung setzen und erweitern. Um Transparenz gegenüber den Implementierungen von Modellierungswerkzeugen und (Abwärts-)Kompatibilität zur BPMN V(1.(0|1)|1.2) [OMG06, OMG08c, OMG09] sowie zum *Business Process Definition MetaModel* (BPDM) [OMG08a, OMG08b] zu gewährleisten, definiert die OMG Schnittstellen.

3.2. Trennung von Notation und Metamodell

Die konsequente Trennung von **Notation** und **Metamodell** ist das Ergebnis von Bestrebungen der OMG, die BPMN besser mit der MDA in Einklang zu bringen.

Die Abbildung 3.2 zeigt die Trennung von Notation und Metamodell sowie die inkrementelle Verfeinerung von fachlichen zu technischen Prozessmodellen.

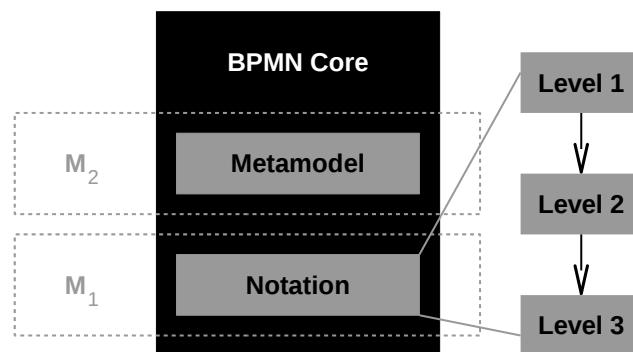


Abbildung 3.2.: Trennung von Notation und Metamodell sowie Verfeinerung

M_1 und M_2 sind die entsprechenden Ebenen der MOF¹. Auf der Ebene M_1 befindet sich die **Notation** – die **Symbole**. Auf der Ebene M_2 ist das **Metamodel**.

Die **Notation** ist in drei *Level* aufgeteilt. Die Aufteilung in Level erhöht die Qualität der Notation, denn es kann für spezielle fachliche Interessengruppen sowie für unterschiedliche Aufgaben mit differenzierterem Detaillierungsgrad modelliert werden.

Leymann [Ley10] sowie Freund und Rücker [FR10] charakterisieren die Level wie folgt:

Level 1

- „Descriptive Modeling“ [Ley10]
- „Strategische Prozessmodelle“ [FR10]

Level 2

- „Analytical Modeling“ [Ley10]
- „Operative Prozessmodelle“ [FR10]

Level 3

- „Executable Modeling“ [Ley10]
- „Technische Prozessmodelle“ [FR10]

Ein Prozessmodell auf *Level 1* ist das grobe Abbild eines strategischen fachlichen Ablaufs. Es ist informativ und dient dem Verständnis sowie der Kommunikation. Ein Prozessmodell auf *Level 2* bildet den Ablauf exakt mit allen fachlichen Details ab. Es dient als Vorlage für ein Prozessmodell auf *Level 3*. Ein Prozessmodell auf *Level 3* enthält die für ein BPMS zusätzlich notwendigen, technischen Details. Es dient der Ausführbarkeit.

Das **Metamodel** der BPMN ist erweiterbar, austauschbar und bestimmt eine eindeutige sowie verständliche(re) Ausführungssemantik. Zur Bearbeitung des ersten Schritts der Problemstellung (Abschnitt 1.2) ist im folgenden Abschnitt die Erweiterbarkeit der BPMN dargestellt.

¹Die *Meta Object Facility* (MOF) [OMG11b] ist eine Spezifikation der MDA. Sie sieht die hierarchisch organisierten Modellierungsebenen M_3 (*Metametamodel*), M_2 (*Metamodel*), M_1 (*Model*) und M_0 (*Data*) vor. M_3 ist die oberste und M_0 die unterste der Ebenen. Die Instanz einer Ebene ist die Extension der Instanz der jeweils übergeordneten und die Intension der Instanz der jeweils untergeordneten Ebene. So ist ein Model die Instanz eines Metamodells.

3.3. Erweiterbarkeit

Die Notation und das Metamodell der BPMN sind unter gewissen Bedingungen erweiterbar. Es gibt **allgemeine** und **formale Bedingungen**. An eine Erweiterung der Notation werden ausschließlich allgemeine Bedingungen gestellt. An eine Erweiterung des Metamodells werden sowohl allgemeine als auch formale Bedingungen gestellt.

3.3.1. Allgemeine Bedingungen

Die allgemeinen Bedingungen sind die Folgenden:

Notation

Die Erweiterung der Notation durch neue Symbole darf den „*look-and-feel*“ [OMG11a] der bereits existierenden Symbole der BPMN nicht beeinträchtigen. Unabhängig von der modellierenden Person, sollte ein erweitertes Prozessmodell weiterhin leicht verständlich sein.

Metamodell

Was das Metamodell angeht, dürfen existierende Elemente durch zusätzliche Attribute erweitert werden. Macht die Abbildung eines speziellen fachlichen Ablaufs („*vertical domain*“ [OMG11a]) das Hinzufügen neuer Elemente erforderlich, dürfen diese die Ausführungssemantik existierender Elemente nicht verletzen. Damit bleibt der BPMN Core valide. Abhängig von der Implementierung des verwendeten Modellierungswerkzeugs, können bei einem Austausch die neuen Elemente (samt Attribute) verloren gehen.

Der Wortlaut der OMG hierzu ist:

„BPMN 2.0 introduces an extensibility mechanism that allows extending standard BPMN elements with additional attributes. It can be used by modelers and modeling tools to add non-standard elements or Artifacts to satisfy a specific need, such as the unique requirements of a vertical domain, and still have valid BPMN Core. Extension attributes MUST NOT contradict the semantics of any BPMN element. In addition, while extensible, BPMN Diagrams should still have the basic look-and-feel so that a Diagram by any modeler should be easily understood by any viewer of the Diagram. Thus the footprint of the basic flow elements (Events, Activities, and Gateways) MUST NOT be altered.“ [OMG11a, S. 44]

„This approach results in more interchangeable models, because the standard elements are still intact and can still be understood by other BPMN adopters. It's only the additional attributes and elements that MAY be lost during interchange.“ [OMG11a, S. 57]

3.3.2. Formale Bedingungen

Zur Bearbeitung des ersten Schritts der Problemstellung (Abschnitt 1.2) sind im Folgenden die **formalen Bedingungen** an eine Erweiterung des Metamodells der BPMN dargestellt.

Die Abbildung 3.3 zeigt das *Extension Class Diagram* aus dem Package Foundation.

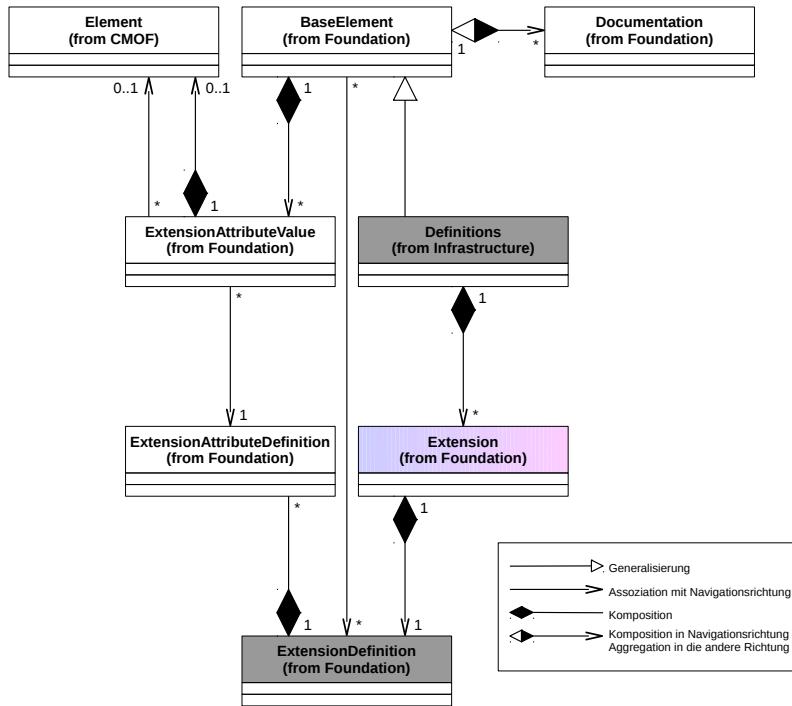


Abbildung 3.3.: Extension Class Diagram [OMG11a]

Das Extension Class Diagram ist das UML-Klassendiagramm [OMG11c] des Metamodells einer Erweiterung.

Definitions, ExtensionDefinition, Extension und ExtensionAttributeDefinition sind die für eine Erweiterung der BPMN wesentlichen Klassen. Sie erlauben es existierende Elemente für die Erweiterung durch zusätzliche Attribute zu nutzen oder das Metamodell der BPMN durch ein zusätzliches Metamodell zu erweitern. Das Erweitern des Metamodells kommt dem Hinzufügen neuer Elemente gleich.

Zusätzliche Attribute existierender Elemente werden einer Instanz der Klasse Definitions aus dem Package Infrastructure hinzugefügt.

Um das Metamodell der BPMN durch ein zusätzliches Metamodell zu erweitern, müssen Instanzen der Klassen Extension und ExtensionAttributeDefinition als Spezialisierung der Klasse ExtensionDefinition aus dem Package Foundation konzipiert werden.

Für die Erweiterung BPMN4Manu erfolgt dies in Abschnitt 4.2 des folgenden Kapitels.

4. Konzept der Erweiterung BPMN4Manu

Das Konzept der Erweiterung BPMN4Manu stellt die Vorbereitung der Umsetzung im Modellierungswerkzeug Oryx (Kapitel 5) dar.

Zunächst werden die **Symbole** der Erweiterung BPMN4Manu eingeführt. Anschließend wird die Erweiterung des **Metamodels** als UML-Klassendiagramme dargestellt und die **Beziehungen** der neuen Elemente in Form von Matrizen aufgeführt. In einer **kritischen Betrachtung** wird der zweite Schritt der Problemstellung (Abschnitt 1.2) hinterfragt und vorbereitet.

4.1. Notation

Im Folgenden werden die von Zor et al. [ZLS11] vorgestellten **neun neuen Symbole** eingeführt. Die Symbole der Erweiterung BPMN4Manu erfüllen die an eine Erweiterung der Notation gestellten, allgemeinen Bedingungen (Abschnitt 3.3.1).

Die Tabelle 4.1 zeigt die Symbole mit deren Name und Kategorie.

Symbol	Name	Kategorie
	<i>Manufacturing Task</i>	<i>Activities</i>
	<i>Machines & Tools</i>	<i>Resource Containers</i>
	<i>Parts</i>	<i>Resource Containers</i>
	<i>Machines & Tools Flow</i>	<i>Connecting Objects</i>
	<i>Parts Flow</i>	<i>Connecting Objects</i>
	<i>Material Route</i>	<i>Gateways</i>
	<i>Material Select</i>	<i>Gateways</i>
	<i>Material Split</i>	<i>Gateways</i>
	<i>Material Join</i>	<i>Gateways</i>

Tabelle 4.1: Symbole der Erweiterung BPMN4Manu

4. Konzept der Erweiterung BPMN4Manu

Die Symbole sind im Folgenden näher beschrieben:

Manufacturing Task

Ein **Manufacturing Task** ist eine einzelne **Aktivität** innerhalb des gesamten Produktionsprozesses. Das Symbol wird der in BPMN bereits existierenden Kategorie *Activities* zugeordnet.

Machines & Tools

Machines & Tools ist ein Behälter für **Maschinen** und **Werkzeuge**. Das Symbol wird der Kategorie *Resource Containers* zugeordnet.

Parts

Parts ist ein Behälter für **Einzelteile** innerhalb des Wertstroms. Das Symbol wird der Kategorie *Resource Containers* zugeordnet.

Machines & Tools Flow

Ein **Machines & Tools Flow** bildet die **Bewegung** von **Maschinen** und **Werkzeugen** ab. Das Symbol wird der in BPMN bereits existierenden Kategorie *Connecting Objects* zugeordnet.

Parts Flow

Ein **Parts Flow** bildet die **Bewegung** von **Einzelteilen** innerhalb des Wertstroms ab. Das Symbol wird der in BPMN bereits existierenden Kategorie *Connecting Objects* zugeordnet.

Material Route

Material Route ist ein **Entscheidungspunkt** für die **Weiterleitung** von **Einzelteilen** innerhalb des Wertstroms. Das Symbol wird der in BPMN bereits existierenden Kategorie *Gateways* zugeordnet.

Material Select

Material Select ist ein **Entscheidungspunkt** für die **Auswahl** von **Einzelteilen** innerhalb des Wertstroms. Das Symbol wird der in BPMN bereits existierenden Kategorie *Gateways* zugeordnet.

Material Split

Material Split ist ein **Entscheidungspunkt** für die **Trennung** von **Einzelteilen** innerhalb des Wertstroms. Das Symbol wird der in BPMN bereits existierenden Kategorie *Gateways* zugeordnet.

Material Join

Material Join ist ein **Entscheidungspunkt** für die **Zusammenführung** von **Einzelteilen** innerhalb des Wertstroms. Das Symbol wird der in BPMN bereits existierenden Kategorie *Gateways* zugeordnet.

Die Tabelle 4.2 zeigt die **Symbole** für einige der **Werkzeuge** und **Einzelteile** aus dem Prozessmodell der Prüfung einer Zündkerze (Kapitel 2).

Symbol	Name	Kategorie
	Wrench	Artifacts
	Ignition Coil	Artifacts
	Ignition Plug	Artifacts

Tabelle 4.2.: Symbole für Werkzeuge und Einzelteile

Die Symbole sind im Folgenden näher beschrieben:

Wrench

Eine **Wrench** ist das **Werkzeug Ratsche**. Das Symbol wird der in BPMN bereits existierenden Kategorie *Artifacts* zugeordnet.

Ignition Coil

Ein **Ignition Coil** ist das **Einzelteil Zündspule** innerhalb des Wertstroms. Das Symbol wird der in BPMN bereits existierenden Kategorie *Artifacts* zugeordnet.

Ignition Plug

Ein **Ignition Plug** ist das **Einzelteil Zündkerze** innerhalb des Wertstroms. Das Symbol wird der in BPMN bereits existierenden Kategorie *Artifacts* zugeordnet.

4.2. Metamodel

In diesem Abschnitt wird die **statische Struktur** des zusätzlichen Metamodells als UML-Klassendiagramme dargestellt. Das zusätzliche Metamodell der Erweiterung BPMN4Manu erfüllt die an eine Erweiterung des Metamodells gestellten, formalen Bedingungen (Abschnitt 3.3.2). Was die Erfüllung der allgemeinen Bedingungen (Abschnitt 3.3.1) angeht, wird an dieser Stelle auf die kritische Betrachtung (Abschnitt 4.4) verwiesen.

Das zusätzliche Metamodell ist als Spezialisierung der Klasse ExtensionDefinition aus dem Package Foundation konzipiert. Es ist nach der Kategorie **Connecting Objects** unterteilt.

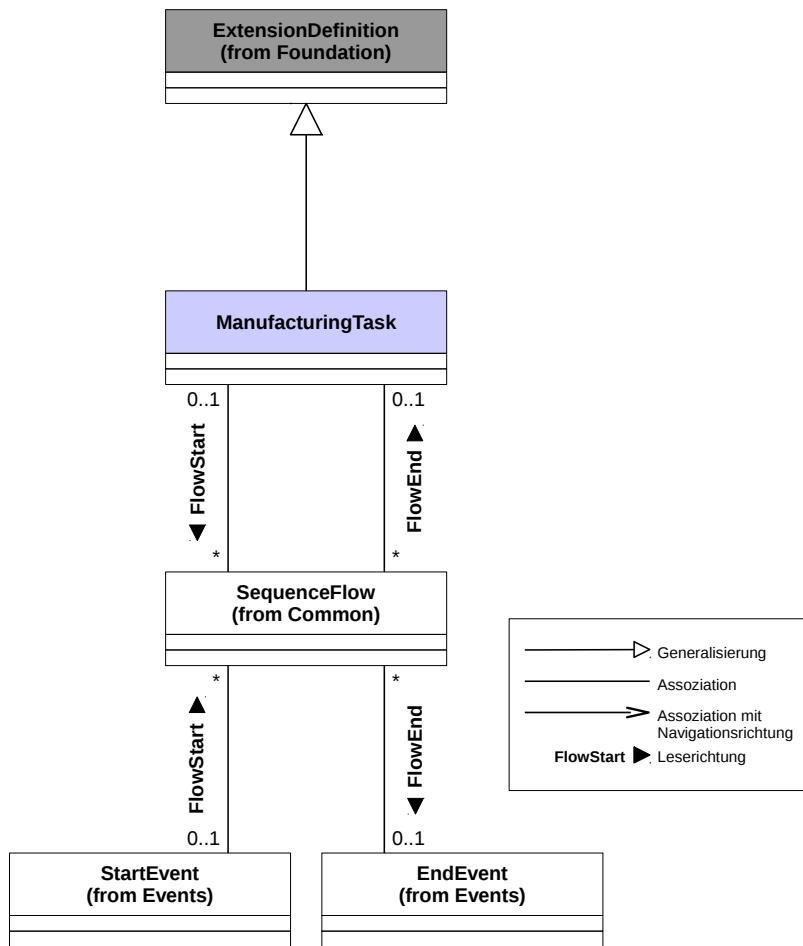


Abbildung 4.1.: Sequence Flow Metamodel

Eine Instanz der Klasse SequenceFlow (Abbildung 4.1) geht die Assoziationen **FlowStart** und **FlowEnd** ein. FlowStart und FlowEnd können mit einer oder keiner Instanz der Klasse ManufacturingTask assoziiert sein. Als Beispiel für Assoziationen zu existierenden Elementen der BPMN, sind die Klassen StartEvent und EndEvent aus dem Package Events dargestellt. FlowStart kann mit einer oder keiner Instanz der Klasse StartEvent assoziiert sein. FlowEnd kann mit einer oder keiner Instanz der Klasse EndEvent assoziiert sein.

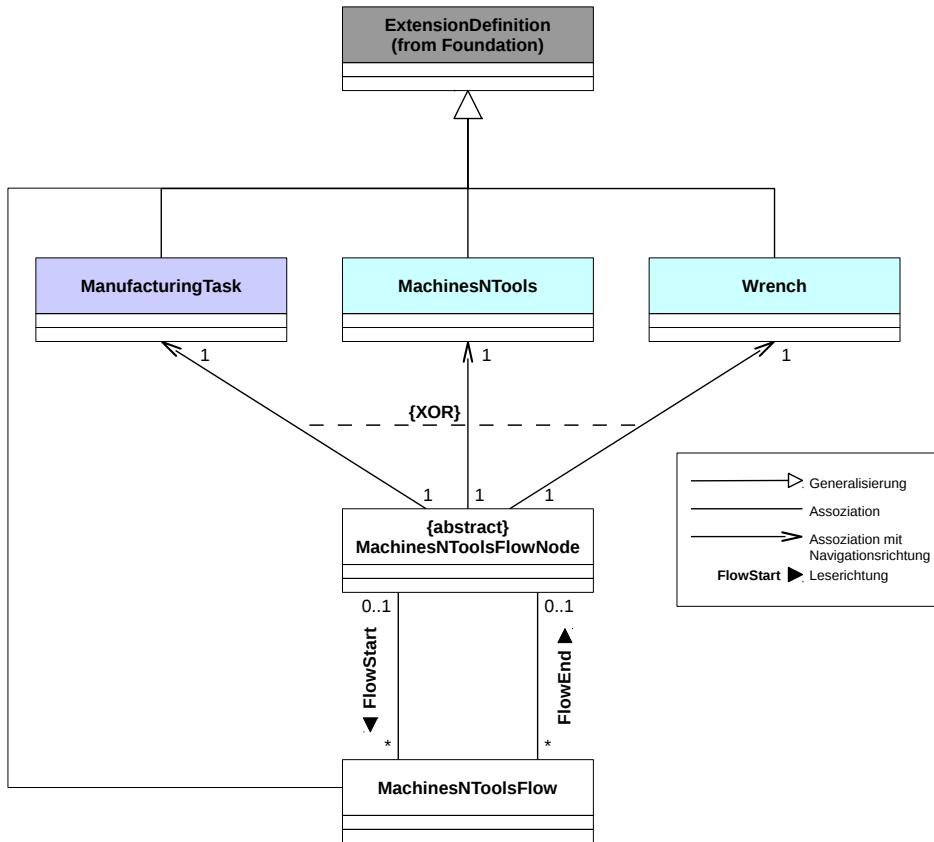


Abbildung 4.2.: Machines & Tools Flow Metamodel

Strukturell verhält sich die Klasse **MachinesNToolsFlow** (Abbildung 4.2) wie die Klasse **SequenceFlow**. Die Assoziation von **FlowStart** und **FlowEnd** zu Instanzen der Klassen **ManufacturingTask**, **MachinesNTools** und **Wrench** erfolgt über die abstrakte Klasse **MachinesNToolsFlowNode**. Das **{XOR}** über der gestrichelten Linie sagt aus, dass die Assoziation von **FlowStart** und **FlowEnd** entweder zu **ManufacturingTask** oder zu **MachinesNTools** oder zu **Wrench** erfolgt. Es dient zur Verbesserung der Lesbarkeit und hat keine strukturelle Bedeutung.

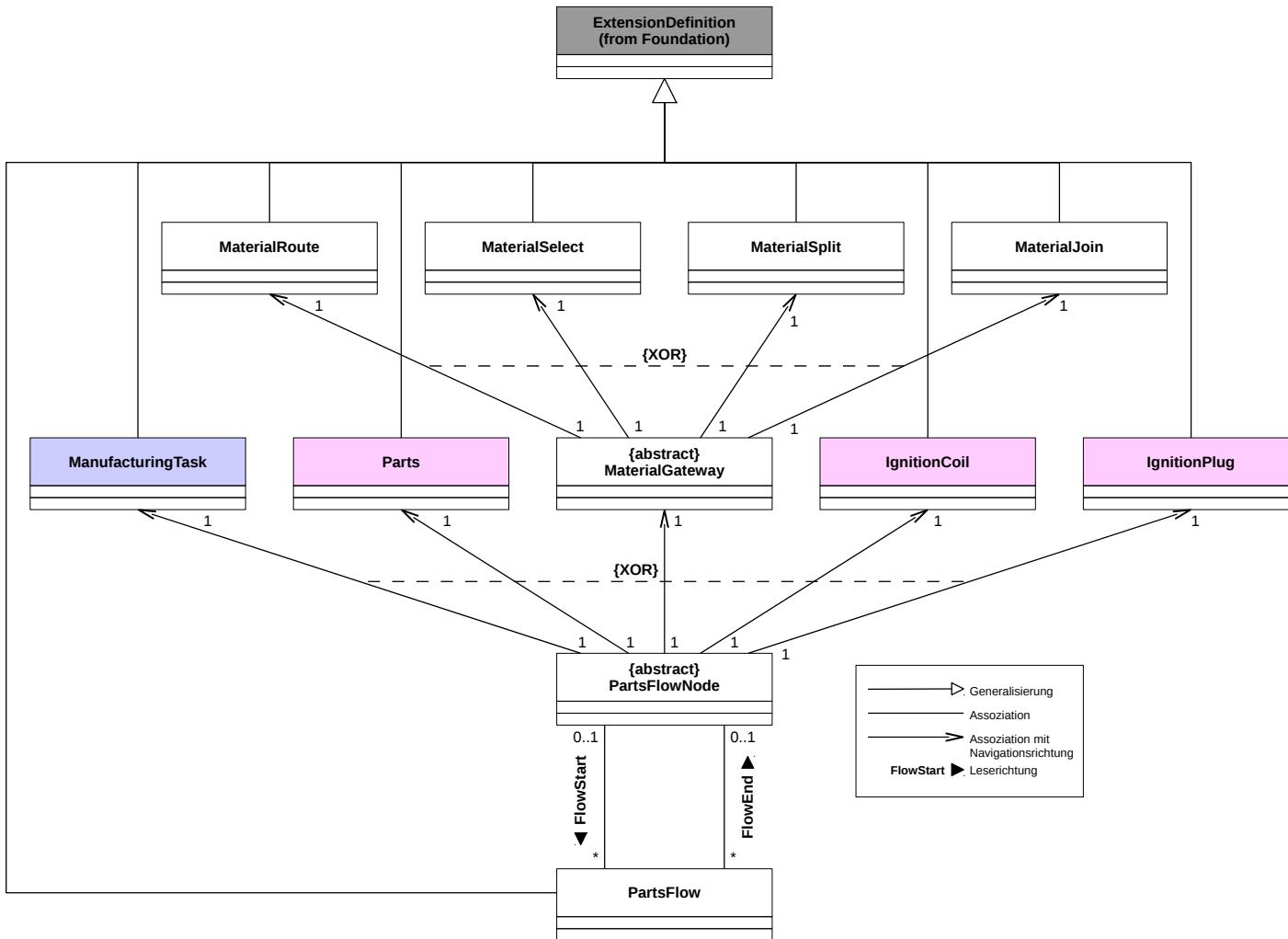


Abbildung 4.3.: Parts Flow Metamodell

Die Klasse PartsFlow (Abbildung 4.3) verhält sich strukturell wie die Klassen SequenceFlow und MachinesNToolsFlow.

Die statische Struktur des zusätzlichen Metamodells spiegelt die direkten Beziehungen der Elemente der Erweiterung BPMN4Manu (Tabelle 4.3) wieder. Die indirekten Beziehungen (Tabellen 4.4 4.5 und 4.6) sind im folgenden Abschnitt in Form von Matrizen aufgeführt.

Die Attribute der Elemente der Erweiterung BPMN4Manu sind in README.txt (Abschnitt A.1) als *Properties* enthalten.

4.3. Beziehungen

Im Folgenden werden die Beziehungen der neuen Elemente in Form von Matrizen aufgeführt. Die Elemente der Erweiterung BPMN4Manu können **direkt** oder **indirekt** zueinander in Beziehung stehen.

	█	█	█	█	█	█	█	█
█				x	x			
█				x				
█					x			
█	x	x						
█	x		x			x	x	x
█					x			
█					x			
█					x			
█					x			

Tabelle 4.3.: Direkte Beziehungen aller Elemente der Erweiterung BPMN4Manu

Die **direkten Beziehungen** (Tabelle 4.3) entsprechen den statischen Strukturen der Erweiterung des Metamodells (Abbildungen 4.1, 4.2 und 4.3).

Die **indirekten Beziehung** sind im Folgenden in Form von Matrizen aufgeführt. Neben der in BPMN bereits existierenden *Sequence Flow* Beziehung (Tabelle 4.4) sind das Machines & Tools Flow (Tabelle 4.5) sowie Parts Flow (Tabelle 4.6).

4. Konzept der Erweiterung BPMN4Manu

	○	○	□	■
○	/	/	/	/
○	/	/	/	/
□	/	/	/	/
■	/	/	/	/

Tabelle 4.4.: Sequence Flow Beziehung

Wie *Start Event* (1. Zeile, 1. Spalte), *End Event* (2. Zeile, 2. Spalte) und *Task* (3. Zeile, 3. Spalte), können auch die weiteren, in BPMN bereits existierenden Elemente, über den Sequence Flow zum Manufacturing Task in Beziehung stehen.

Die Tabellen 4.5 und 4.6 zeigen die Machines & Tools Flow Beziehung sowie die Parts Flow Beziehung.

	■	□
■	/	/
□	/	/

Tabelle 4.5.: Machines & Tools Flow Beziehung

	■	○	◊	◊	◆	◆
■	/	/	/	/	/	/
○	/	/	/	/	/	/
◊	/	/	/	/	/	/
◊	/	/	/	/	/	/
◆	/	/	/	/	/	/
◆	/	/	/	/	/	/

Tabelle 4.6.: Parts Flow Beziehung

4.4. Kritische Betrachtung

Durch die folgende kritische Betrachtung der Bearbeitung des ersten Schritts der Problemstellung (Abschnitt 1.2) wird der zweite Schritt der Problemstellung hinterfragt und vorbereitet.

Der zweite Schritt der Problemstellung ist die inkrementelle Verfeinerung von fachlichen zu technischen Prozessmodellen. Dabei muss die Transformation einschließlich der Erweiterung BPMN4Manu in ein ausführbares Workflow Model in der *Business Process Execution Language* (BPEL) [OAS02, OAS03, OAS07] berücksichtigt werden.

Zunächst ist anzumerken, dass die Transformation in ein ausführbares Workflow Model in BPEL notwendig und hinreichend ist, denn:

- Produktionsunternehmen sollen zur Schließung der Business Manufacturing Gap in einer SOA kollaborieren können und BPEL ist zur Orchestrierung von Web Services vorgesehen und fest etabliert.
- Ersten Recherchen zufolge existieren auf dem Markt keine fest etablierten BPMS, die technische Prozessmodell (unmittelbar) in BPMN ausführen können. Wenn ein technisches Prozessmodell in BPMN von einem BPMS ausgeführt werden kann, dann findet intern (meist) die Transformation in ein ausführbares Workflow Model in BPEL statt.

Zur Vorbereitung der Transformation einschließlich der Erweiterung BPMN4Manu in ein ausführbares Workflow Model in BPEL, ist die Durchführung der folgenden Schritte nötig:

1. Es muss untersucht werden, ob eines der Elemente der Erweiterung BPMN4Manu die Ausführungssemantik existierender Elemente der BPMN verletzt; Allgemeine Bedingungen an eine Erweiterung des Metamodells der BPMN (Abschnitt 3.3.1).
2. Des Weiteren muss untersucht werden, ob und inwieweit ein Produktionsprozessmodell in BPMN4Manu vor der Transformation in ein Workflow Model in BPEL verfeinert werden muss. Sind Produktionsprozessmodelle in BPMN4Manu strategische (Level 1), operative (Level 2) oder technische (Level 3) Prozessmodelle?
3. Ist eine Verfeinerung nötig, dann dürfen auch die Elemente des verfeinerten Produktionsprozessmodells die Ausführungssemantik existierender Elemente der BPMN nicht verletzen.

5. Umsetzung im Modellierungswerkzeug Oryx

Die Umsetzung des Konzepts der Erweiterung BPMN4Manu im Modellierungswerkzeug Oryx schließt die Bearbeitung des ersten Schritts der Problemstellung (Abschnitt 1.2) ab. In einer Einführung wird zunächst ein Teil der Architektur von Oryx beschrieben. Anschließend wird die Realisierung der Notation wie auch die Definition des Metamodells und der Beziehungen erklärt.

5.1. Einführung in Oryx

Oryx [Tsc07, Czu07, Polo07, Peto07, Has] ist ein webbasiertes Modellierungswerkzeug. Es wurde 2006/2007 am Hasso-Plattner-Institut entwickelt. Der Quellcode von Oryx ist frei verfügbar. Dieser kann anhand der Beschreibung in Abschnitt A.2 installiert werden.

Die Abbildung 5.1 zeigt einen Teil der Architektur von Oryx als *Block Diagram* der *Fundamental Modeling Concepts* (FMC) [KGT06, Ren10].

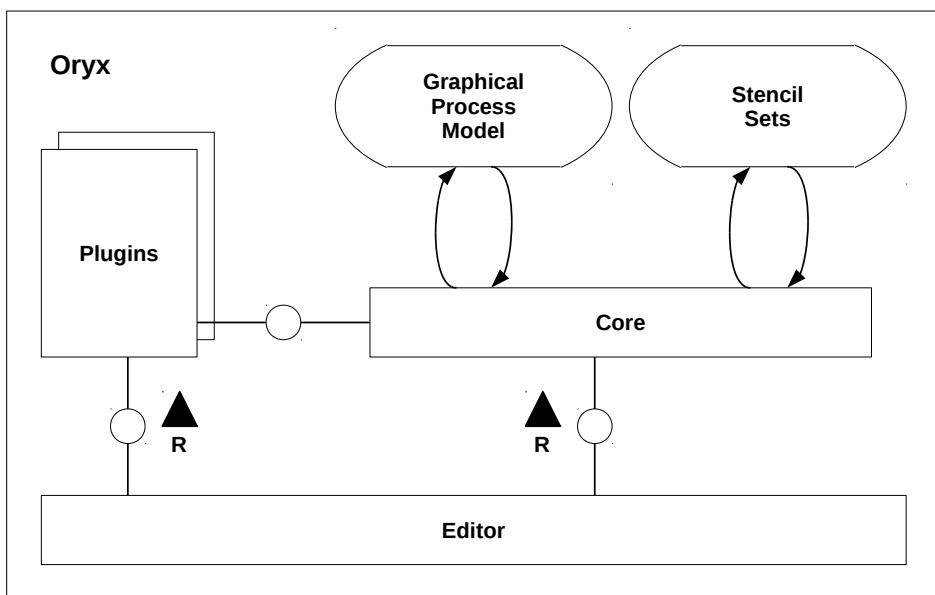


Abbildung 5.1.: Architektur von Oryx [Tsc07]

5. Umsetzung im Modellierungswerkzeug Oryx

Stencil Sets sind eine Menge von Elementen und Beziehungen. Sie beschreiben Modellierungssprachen in der *JavaScript Object Notation (JSON)* [Int11]. Der **Core** von Oryx setzt das **Graphical Process Model** eines Stencil Sets um und präsentiert es der **Editor** Komponente. Die Komponente Editor initialisiert die Stencil Sets, die Zeichenfläche und die **Plugins**. Der Editor ermöglicht es den **Plugins** außerdem, über eine API auf Kernfunktionen des Editors zuzugreifen. Die **Plugins** setzen die grafische Benutzerschnittstelle um und erweitern die Funktionalität der Anwendung.

Für die folgenden Abschnitte wird festgelegt, dass \$inst der Dateisystempfad über einer Installation von Oryx ist. Um Pfadangaben zu verkürzen, wird weiterhin festgelegt, dass \$instoedse der Pfad \$inst/oryx/editor/data/stencilsets/extensions ist.

5.2. Notation

Zur Umsetzung der Notation der Erweiterung BPMN4Manu müssen die **Symbole der Zeichenfläche (View)** als *Scalable Vector Graphics (SVG)* [Word] realisiert werden. Mit SVG können Vektorgrafiken in XML beschrieben werden. Die XML Struktur ermöglicht die Definition individueller *Namespaces* und damit individueller Tags. Somit können in einer durch den Browser interpretierten Sprache wie *JavaScript* [Int11] individuelle Funktionen der Zeichenfläche zur Verfügung gestellt werden. Zusätzlich zur Umsetzung der Notation im View als SVG müssen *Portable Network Graphics (PNG)* [Worc] für die **Symbole neben der Zeichenfläche (Icon)** bereitgestellt werden.

Im Rahmen dieser Arbeit wurden 12 neue Symbole in SVG realisiert und als PNG bereitgestellt. Alle in dieser Arbeit dargestellten Symbole basieren auf diesen SVG. Details hierzu sind in README.txt (Abschnitt A.1) enthalten.

5.3. Metamodel und Beziehungen

Zur Umsetzung des Metamodells und der Beziehungen der Erweiterung BPMN4Manu muss die Datei \$instoedse/extensions.json modifiziert sowie ein Verzeichnis \$instoedse/bpmn4manu und eine Datei \$instoedse/bpmn4manu/bpmn4manu.json hinzugefügt werden. Die Dateien werden zur **Registrierung der Erweiterung eines Stencil Sets (Extension)** sowie zur **Definition von neuen Elementen (Stencils), direkten Beziehungen (Rollen)** und **indirekten Beziehungen (Regeln)** durch die Komponente Core gelesen.

Das Listing 5.1 zeigt den Auszug aus der Datei \$instoedse/extensions.json zur Registrierung der Stencil Set Extension BPMN4Manu.

Listing 5.1 Registrierung der Stencil Set Extension BPMN4Manu

```
1  {
2      "title": "BPMN4Manu",
3      "namespace": "http://oryx-editor.org/stencilsets/extensions/bpmn4manu#",
4      "description": "Manufacturing extension of BPMN 2.0.",
5      "definition": "bpmn4manu/bpmn4manu.json",
6      "extends": "http://b3mn.org/stencilset/bpmn2.0#"
7 }
```

Liest der Core diesen Auszug, steht im Editor für ein Prozessmodell in BPMN die Extension BPMN4Manu zur Verfügung. Der Wert von definition verweist auf die Datei \$instoedse/bpmn4manu/bpmn4manu.json.

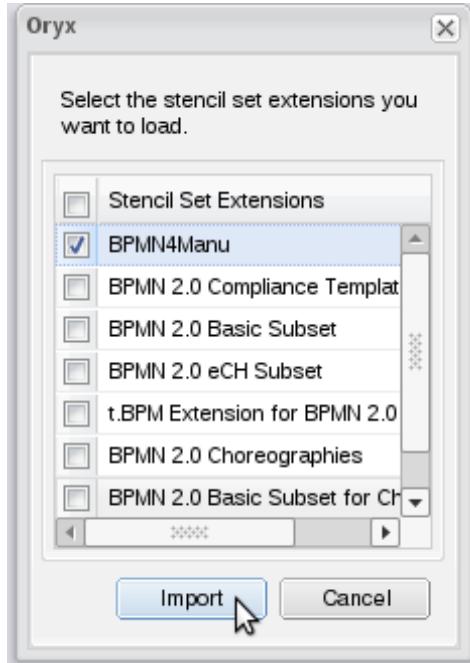


Abbildung 5.2.: Laden der Stencil Set Extension BPMN4Manu

Nach dem Laden der Stencil Set Extension BPMN4Manu (Abbildung 5.2) stehen die Symbole der Erweiterung BPMN4Manu aus den Tabellen 4.1 und 4.2 als View und als Icon zur Verfügung.

5. Umsetzung im Modellierungswerkzeug Oryx

Das Listing 5.2 zeigt einen Auszug aus der Datei \$instoedse/bpmn4manu/bpmn4manu.json zur Definition des Stencils ManuTask.

Listing 5.2 Definition des Stencils ManuTask

```
1  {
2      "type": "node",
3      "id": "ManuTask",
4      "title": "Manufacturing Task",
5      "groups": ["Activities"],
6      "view": "../../extensions/bpmn4manu/view/activity/manuTask.svg",
7      "icon": "../../extensions/bpmn4manu/icons/activity/manuTask.png",
8      "properties": [/*...*/]
9 }
```

Das Stencil ManuTask ist vom Typ node. Auch alle weiteren Stencils der Erweiterung sind vom Typ node. Einzig die Stencils MachinesNToolsFlow und PartsFlow sind vom Typ edge. Die Werte von view und icon verweisen auf die Realisierung des Symbols der Zeichenfläche und auf die Bereitstellung des Symbols neben der Zeichenfläche.

Listing 5.3 Definition der roles des Stencils ManuTask

```
1  "roles": [
2      "sequence_start",
3      "sequence_end",
4
5      "MachinesNToolsFlow_start",
6      "MachinesNToolsFlow_end",
7
8      "PartsFlow_start",
9      "PartsFlow_end"
10 ]
```

Die Definition der roles (Rollen) des Stencils ManuTask (Listing 5.3) spiegelt die direkten Beziehungen aller Elemente der Erweiterung BPMN4Manu (Tabelle 4.3) wieder.

Das Listing 5.4 zeigt den Auszug aus der Datei \$instoedse/bpmn4manu/bpmn4manu.json zur Definition der connectionRules (Regeln) der Stencil Set Extension BPMN4Manu.

Listing 5.4 Definition der connectionRules

```

1 "connectionRules": [
2   {
3     "role": "MachinesNToolsFlow",
4     "connects": [
5       {
6         "from": "MachinesNToolsFlow_start",
7         "to": ["MachinesNToolsFlow_end"]
8       }
9     ]
10 },
11 {
12   "role": "PartsFlow",
13   "connects": [
14     {
15       "from": "PartsFlow_start",
16       "to": ["PartsFlow_end"]
17     }
18   ]
19 }
20 ]

```

Wird einem Stencil eine role zugeordnet, dann kann dieses Stencil indirekte Beziehungen zu all den Stencils eingehen, denen roles dieser rules zugeordnet sind. Die Tabellen 4.4, 4.5 und 4.6 verdeutlichen diesen Zusammenhang.

Die Tests der connectionRules sind in Abschnitt A.3 dargestellt.

Listing 5.5 Definition der containmentRules

```

1 "containmentRules": [
2   {
3     "role": "MachinesNTools",
4     "contains": ["Wrench"]
5   },
6   {
7     "role": "Parts",
8     "contains": [
9       "IgnitionCoil",
10      "IgnitionPlug"
11    ]
12  }
13 ]

```

Durch die containmentRules (Listing 5.5) können die Resource Containers Werkzeuge und Einzelteile beinhalten.

6. Erweitertes Szenario

Die Abbildung 6.2 zeigt das Prozessmodell der Prüfung einer Zündkerze aus Kapitel 2 mit der Erweiterung BPMN4Manu innerhalb des Modellierungswerkzeugs Oryx. Es enthält die konzipierten (Kapitel 4) und umgesetzten (Kapitel 5) Erweiterungen.

Der Schriftzug **BPMN 2.0 / BPMN4Manu** unterhalb der *Shape Repository* links neben der Zeichenfläche weist darauf hin, dass die Stencil Set Extension BPMN4Manu registriert und geladen ist.

Innerhalb der Kategorie Activities ist das Stencil ManuTask samt Icon, innerhalb der Kategorie Resource Containers sind die Stencils MachinesNTools und Parts beide samt Icon, innerhalb der Kategorie Connecting Objects sind die Stencils MachinesNToolsFlow und PartsFlow beide samt Icon und innerhalb der Kategorie Gateways sind die Stencils MaterialRoute, MaterialSelect, MaterialSplit und MaterialJoin ebenfalls alle samt Icon.

Zieht man ein Stencil des Stencil Sets BPMN4Manu aus der Shape Repository in die Zeichenfläche, wird aus dem Icon ein View.

Das Prozessmodell (Abbildung 6.2) enthält mehrere Instanzen der als *roles* definierten direkten Beziehungen und der als *connectionRules* definierten indirekten Beziehungen.

Die Abbildung 6.1 zeigt das Ergebnis eines Tests der *containmentRules* aus Listing 5.5.

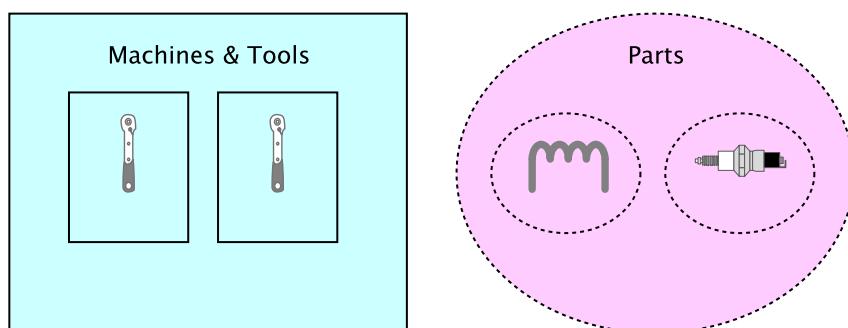


Abbildung 6.1.: Test der *containmentRules*

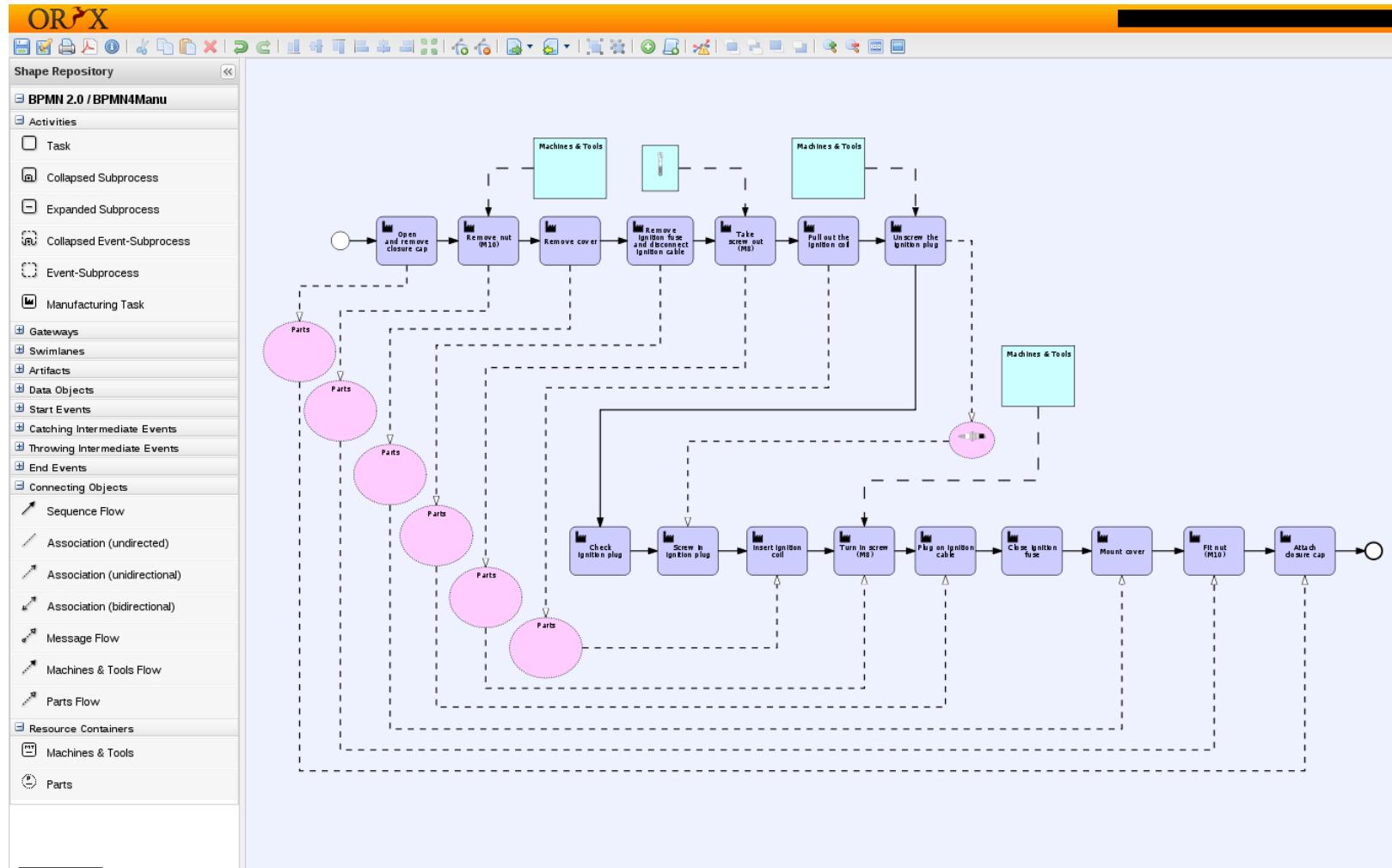


Abbildung 6.2.: Prüfen einer Zündkerze in BPMN4Manu [ZLS11] innerhalb von Oryx

7. Verwandte Arbeiten

Die im Folgenden vorgestellten Arbeiten besitzen eine Erweiterung der BPMN V1.0 [OMG06], BPMN V1.1 [OMG08c], BPMN V1.2 [OMG09] oder der BPMN V2.0 [OMG11a] sowie eine Umsetzung im Modellierungswerkzeug Oryx [Tsc07, Czu07, Polo07, Pet07, Has].

Initiiert von Decker und Puhlmann [DP07] stellt Pfitzner [Pfi07] die **Erweiterung BPMN⁺** der **BPMN V1.0** vor. BPMN⁺ erlaubt eine **graphische Modellierung der BPEL4Chor** [DKLW07] in der BPMN V1.0. Es wird das Ziel verfolgt BPMN⁺ so zu gestalten, dass keine Verfeinerung der WS-BPEL V2.0 notwendig ist. Für die BPMN V1.0 ist in der Spezifikation kein Austauschformat vorgesehen. Zur Verbindung der BPMN⁺ und der BPEL4Chor wird daher die *XML Process Definition Language Version 2.0* (XPDL V2.0) [WFM05] zu der *XPDL4Chor* erweitert und anschließend die Verbindung der XPDL4Chor zu der BPEL4Chor definiert. BPMN⁺ ist in Oryx umgesetzt. Pfitzner et al. [PDKL07] und Decker et al. [DKL⁺08] veröffentlichen Teile der Ergebnisse.

Rodríguez et al. [RFMP07] **erweitern** die **BPMN V1.0** durch Elemente zur Modellierung von **Sicherheitsanforderungen** in Geschäftsprozessen. Da die BPMN in der Version 1.0 kein Metamodel enthält, entwickeln sie es grundlegend und führen eine Spezialisierung durch. Im Ausblick erwähnen Rodríguez et al., dass der nächste Schritt die Umsetzung einer Transformation einschließlich der Sicherheitsanforderungen in ein ausführbares Workflow Model im Sinne der Model Driven Architecture (MDA) [OMG03] sein sollte.

Decker und Barros [DB07] stellen die **Erweiterung iBPMN** der **BPMN V1.0** vor. iBPMN ermöglicht es, **Choreographien** auf einer **konzeptionellen Ebene** zu modellieren. In der BPMN V1.0 ist es vorgesehen, Choreographien anhand einer Beschreibung der Interconnection von Schnittstellen (collaboration) zu modellieren. Bei dieser Methode entsteht ein redundanter Kontrollfluss und es besteht die Gefahr, Prozesse mit inkompatiblem Verhalten, z. B. *deadlocks*, zu modellieren. Mit der iBPMN können die meisten *Service Interaction Patterns* [BDH05] ohne die genannten Schwierigkeiten modelliert werden.

Meyer [Mey09] **erweitert** die **BPMN V1.1** durch eine **Ressourcen-Perspektive**. Die Ressourcen-Perspektive soll die Verteilung und Zuweisung von Aufgaben zu Ressourcen ermöglichen. Hierzu werden Kontrollflussaspekte der BPMN V1.1 mit Aspekten aus dem Bereich der Organisationsmodelle kombiniert, die Anforderungen gegen die Spezifikation der BPMN V1.1 evaluiert und eine Verbindung zu *Workflow Resource Patterns* [RHEA04] identifiziert. Die formale Repräsentation der Ressourcen-Perspektive bedingt eine Überarbeitung des Metamodells der BPMN V1.1 und des Aufgaben-Lebenszyklusses. Den Abschluss bildet eine prototypische, auf ausgewählten Anforderungen basierende Umsetzung in Oryx.

7. Verwandte Arbeiten

Schumm [Scho8] **visualisiert** die Konstrukte der **WS-BPEL V2.0** in der **BPMN V1.1**. Nach einer Einführung in die WS-BPEL V2.0 und die BPMN V1.1, einer Gegenüberstellung der Sprachkonzeptionen sowie einer Darstellung der Elemente werden Ansätze zu deren Verbindung vorgestellt. Es folgt eine Beschreibung und Gegenüberstellung von grafischen Editoren sowie die Definition eines geeigneten Konzepts zur Verbindung von BPMN V1.1 und WS-BPEL V2.0. Dieses Konzept ist das Ergebnis einer kritischen Analyse der vorgestellten Ansätze und führt zu einer alternativen Herangehensweise. Es findet keine Modelltransformation statt. „*Die gegebenen BPEL-Konstrukte werden lediglich in der BPMN-Notation visualisiert.*“ [Scho8, S. 98] Schumm et al. [SKLN09] haben die Ergebnisse veröffentlicht.

Schleicher et al. [SWLS10] definieren einen Bereich, in welchem zuvor festgelegte Regeln eingehalten werden müssen. Zu diesem Zweck wird das **Metamodel** der **BPMN V2.0** durch ein neues Element, den **Compliance Scope**, **erweitert**. Der Compliance Scope enthält ein Attribut mit einer Liste der in *Linear Temporal Logic* (LTL) definierten Regeln. Grohe [Gro10] setzt die Definition des Compliance Scopes in Oryx um.

8. Zusammenfassung und Ausblick

Das Konzept der Erweiterung BPMN4Manu (Kapitel 4) ist entwickelt und im Modellierungswerkzeug Oryx umgesetzt (Kapitel 5); Es existiert *ein Modellierungswerkzeug für Produktionsprozesse auf Basis einer BPMN-Erweiterung*.

Aus der Motivation geht hervor, dass auch Produktionsunternehmen in einer SOA kollabrieren sollten und dass hierzu die Business Manufacturing Gap geschlossen werden muss. Einer der Schritte zur Schließung der Business Manufacturing Gap ist die Umsetzung der Phase Modellierung für Produktionsprozesse.

Die BPMN zeichnet sich durch den Aufbau der Spezifikation sowie durch die konsequente Trennung von Notation und Metamodel aus. Der BPMN Core ist grundlegend, präzise und robust. Die Layer Structure macht die BPMN flexibel, transparent und (abwärts)kompatibel. Durch die konsequente Trennung von Notation und Metamodel ist die Qualität der Notation erhöht. Das Metamodel der BPMN ist erweiterbar, austauschbar und bestimmt eine eindeutige sowie verständliche(re) Ausführungssemantik.

Das Konzept der Erweiterung BPMN4Manu erfüllt die an eine Erweiterung der BPMN gestellten, formalen Bedingungen. Es werden die neuen Symbole eingeführt und die Erweiterung des Metamodells als UML-Klassendiagramme dargestellt. Die statische Struktur des zusätzlichen Metamodells ist als Spezialisierung der Klasse ExtensionDefinition aus dem Package Foundation des BPMN Core konzipiert und nach der Kategorie Connecting Objects unterteilt. Die direkten und die indirekten Beziehungen sind in Form von Matrizen aufgeführt.

Im Modellierungswerkzeug Oryx sind Stencil Sets eine Menge von Elementen und Beziehungen. Neue Symbole werden für den View als SVG realisiert und für das Icon als PNG bereitgestellt. Neue Elemente und deren (in)direkten Beziehungen werden als JSON definiert. Die direkten Beziehungen sind die roles der Stencils und die indirekten Beziehungen sind die rules des Stencil Sets.

Zur Schließung der Business Manufacturing Gap sollte ein Produktionsprozessmodell alle fachlichen Details sowie die für ein BPMS zusätzlich notwendigen, technischen Details exakt abbilden können. Schließlich sollte die Transformation einschließlich der Erweiterung BPMN4Manu in ein ausführbares Workflow Model in BPEL umgesetzt werden. Es muss untersucht werden, ob eines der Elemente der Erweiterung BPMN4Manu die Ausführungssemantik existierender Elemente der BPMN verletzt. Des Weiteren muss untersucht werden, ob und inwieweit ein Produktionsprozessmodell in BPMN4Manu vor der Transformation in ein Workflow Model in BPEL verfeinert werden muss. Ist eine Verfeinerung nötig, dann dürfen auch die Elemente des verfeinerten Produktionsprozessmodells die Ausführungssemantik existierender Elemente der BPMN nicht verletzen.

Literaturverzeichnis

- [BBD⁺08] M. Behmel, U. Berger, M. Dambacher, W. Günter, B. Heine, F. Holzwarth, H. Kaiser, E. Kalhöfer, M. Kaufmann, F. Klein, G. Rohde, R. R. K. Schekulin, G. Schellenberg, D. Schmid, P. Strobel. *Industrielle Fertigung – Fertigungsverfahren*. Europa Lehrmittel, 2008. (Zitiert auf Seite 10)
- [BDH05] A. P. Barros, M. Dumas, A. H. M. ter Hofstede. Service Interaction Patterns. In *Proceedings of the International Conference on Business Process Management (BPM)*. 2005. (Zitiert auf Seite 39)
- [Bod93] J. Bode. *Betriebliche Produktion von Information*. Deutscher Universitätsverlag, 1993. (Zitiert auf Seite 10)
- [Czu07] M. Czuchra. Oryx – Embedding Business Process Data into the Web, 2007. (Zitiert auf den Seiten 31, 39 und 57)
- [DBo07] G. Decker, A. P. Barros. Interaction Modeling using BPMN. In *Proceedings of the International Workshop on Collaborative Business Processes (CBP)*. 2007. (Zitiert auf Seite 39)
- [Deb] Debian Project. Debian. <http://www.debian.org/>. Abgerufen am 29.02.2012. (Zitiert auf Seite 58)
- [Deu] Deutsche MTM-Vereinigung e.V. Methods Time Measurement (MTM) - KOSTEN VERMEIDEN - statt kosten senken. <https://www.dmtm.com/>. Abgerufen am 29.02.2012. (Zitiert auf Seite 13)
- [DKL⁺08] G. Decker, O. Kopp, F. Leymann, K. Pfitzner, M. Weske. Modeling Service Choreographies using BPMN and BPEL4Chor. In *Proceedings of the International Conference on Advanced Information Systems Engineering (CAiSE)*. 2008. (Zitiert auf Seite 39)
- [DKLW07] G. Decker, O. Kopp, F. Leymann, M. Weske. BPEL4Chor: Extending BPEL for Modeling Choreographies. In *Proceedings of the International Conference on Web Services (ICWS)*. 2007. (Zitiert auf Seite 39)
- [DPo07] G. Decker, F. Puhlmann. Extending BPMN for Modeling Complex Choreographies. In *Proceedings of the International Conference on On the Move to Meaningful Internet Systems (OTM)*. 2007. (Zitiert auf Seite 39)

- [EHLB95] D. J. Elzinga, T. Horak, C.-Y. Lee, C. Bruner. Business Process Management: Survey and Methodology. *IEEE Transactions on Engineering Management (TSE)*, 42(2):119–128, 1995. (Zitiert auf Seite 9)
- [FR10] J. Freund, B. Rücker. *Praxishandbuch BPMN 2.0*. Hanser, 2010. (Zitiert auf Seite 17)
- [Gifo07] C. Gifford. *The Hitchhiker's Guide to Manufacturing Operations Management: ISA-95 Best Practices Book 1.0*. ISA, 2007. (Zitiert auf Seite 10)
- [Gro10] S. Grohe. Visualisierung und Implementierung von Compliance Scopes, 2010. (Zitiert auf Seite 40)
- [GT12] H.-O. Günther, H. Tempelmeier. *Produktion und Logistik*. Springer, 2012. (Zitiert auf Seite 10)
- [Gut83] E. Gutenberg. *Grundlagen der Betriebswirtschaftslehre, Erster Band: Die Produktion*. Springer, 1983. (Zitiert auf Seite 10)
- [Has] Hasso-Plattner-Institute. Oryx: Research. <https://bpt.hpi.uni-potsdam.de/Oryx/Research>. Abgerufen am 29.02.2012. (Zitiert auf den Seiten 31, 39 und 57)
- [HC93] M. Hammer, J. Champy. *Reengineering the Corporation: A Manifesto for Business Revolution*. HarperBusiness, 1993. (Zitiert auf Seite 9)
- [ICCo8] M. International, I. Corporation, Capgemini. SOA in Manufacturing Guidebook, 2008. (Zitiert auf Seite 10)
- [Int11] E. International. ECMAScript Language Specification (ECMA-262), 2011. (Zitiert auf den Seiten 32 und 57)
- [KGTo6] A. Knöpfel, B. Gröne, P. Tabeling. *Fundamental Modeling Concepts: Effective Communication of IT Systems*. Wiley, 2006. (Zitiert auf Seite 31)
- [Kop] O. Kopp. SetupDevelopmentEnvironment. <http://code.google.com/p/oryx-editor/wiki/SetupDevelopmentEnvironment>. Abgerufen am 29.02.2012. (Zitiert auf Seite 58)
- [Kun] M. Kunze. InstallUpdateConfigureProductiveSystems. <http://code.google.com/p/oryx-editor/wiki/InstallUpdateConfigureProductiveSystems>. Abgerufen am 29.02.2012. (Zitiert auf Seite 58)
- [Ley10] F. Leymann. Workflow Management. Lecture, 2010. (Zitiert auf den Seiten 9, 10 und 17)
- [LRoo] F. Leymann, D. Roller. *Production Work Flow: Concepts and Techniques*. Prentice Hall PTR, 2000. (Zitiert auf Seite 10)
- [Mey09] A. Meyer. Resource Perspective in BPMN, 2009. (Zitiert auf Seite 39)
- [OASo2] OASIS. Business Process Execution Language for Web Services (BPEL4WS) Version 1.0, 2002. (Zitiert auf Seite 29)

- [OASo3] OASIS. Business Process Execution Language for Web Services (BPEL4WS) Version 1.1, 2003. (Zitiert auf Seite 29)
- [OASo7] OASIS. Web Services Business Process Execution Language (WS-BPEL) Version 2.0, 2007. (Zitiert auf Seite 29)
- [OMG03] OMG. MDA Guide Version 1.0.1, 2003. (Zitiert auf den Seiten 11 und 39)
- [OMG06] OMG. Business Process Modeling Notation (BPMN) Version 1.0, 2006. (Zitiert auf den Seiten 10, 16 und 39)
- [OMG08a] OMG. Business Process Definition MetaModel (BPDM) Version 1.0 – Volume I: Common Infrastructure, 2008. (Zitiert auf Seite 16)
- [OMG08b] OMG. Business Process Definition MetaModel (BPDM) Version 1.0 – Volume II: Process Definitions, 2008. (Zitiert auf Seite 16)
- [OMG08c] OMG. Business Process Modeling Notation (BPMN) Version 1.1, 2008. (Zitiert auf den Seiten 10, 16 und 39)
- [OMG09] OMG. Business Process Model and Notation (BPMN) Version 1.2, 2009. (Zitiert auf den Seiten 10, 16 und 39)
- [OMG11a] OMG. Business Process Model and Notation (BPMN) Version 2.0, 2011. (Zitiert auf den Seiten 10, 15, 16, 18, 19 und 39)
- [OMG11b] OMG. Meta Object Facility (MOF) Core Specification, 2011. (Zitiert auf Seite 17)
- [OMG11c] OMG. OMG Unified Modeling Language (OMG UML), Superstructure Version 2.4.1, 2011. (Zitiert auf Seite 19)
- [Ope] OpenID Foundation. OpenID Foundation website. <http://openid.net/>. Abgerufen am 29.02.2012. (Zitiert auf Seite 67)
- [PDKL07] K. Pfitzner, G. Decker, O. Kopp, F. Leymann. Web Service Choreography Configurations for BPMN. In *Proceedings of the International Workshop on Engineering Service-oriented Applications: Analysis, Design and Composition (WESOA)*. 2007. (Zitiert auf Seite 39)
- [Pet07] N. Peters. Oryx – Stencil Set Specification, 2007. (Zitiert auf den Seiten 31, 39 und 57)
- [Pfio07] K. Pfitzner. Choreography Configuration for BPMN, 2007. (Zitiert auf Seite 39)
- [Polo07] D. Polak. Oryx – Stencil Set Implementation, 2007. (Zitiert auf den Seiten 31, 39 und 57)
- [LATEX Project] LATEX Project. LATEX – A document preparation system. <http://www.latex-project.org/>. Abgerufen am 29.02.2012. (Zitiert auf Seite 57)
- [Ren10] B. Renz. Fundamental Modeling Concepts – Ein mentaler Rahmen für Softwarearchitektur - Slides, 2010. (Zitiert auf Seite 31)

- [RFMPo7] A. Rodríguez, E. Fernández-Medina, M. Piattini. A BPMN Extension for the Modeling of Security Requirements in Business Processes. *IEICE Transactions on Information and Systems*, E90-D(4):745–752, 2007. (Zitiert auf Seite 39)
- [RHEAo4] N. Russell, A. H. M. ter Hofstede, D. Edmond, W. M. P. van der Aalst. WORKFLOW RESOURCE PATTERNS. Technical report, Eindhoven University of Technology, 2004. (Zitiert auf Seite 39)
- [RS99] M. Rother, J. Shook. *Learning to See: Value Stream Mapping to Add Value and Eliminate MUDA*. Lean Enterprise Institute, 1999. (Zitiert auf den Seiten 10 und 13)
- [Scho08] D. Schumm. Graphische Modellierung von BPEL Prozessen unter Verwendung der BPMN Notation, 2008. (Zitiert auf Seite 40)
- [SGNWPo8] G. Schuh, S. Gottschalk, J. Nöcker, C. Wesch-Potente. Flexible Vernetzung statt starrer Integration – die Zukunft der digitalen Fabrik. *wt Werkstattstechnik online*, 3:127–131, 2008. (Zitiert auf Seite 10)
- [SHo4] P. Stahlknecht, U. Hasenkamp. *Einführung in die Wirtschaftsinformatik*. Springer, 2004. (Zitiert auf Seite 10)
- [SKLN09] D. Schumm, D. Karastoyanova, F. Leymann, J. Nitzsche. On Visualizing and Modelling BPEL with BPMN. In *Proceedings of the International Workshop on Workflow Management (IWWM)*. 2009. (Zitiert auf Seite 40)
- [SWLS10] D. Schleicher, M. Weidmann, F. Leymann, D. Schumm. Compliance Scopes: Extending the BPMN 2.0 Meta Model to Specify Compliance Requirements. In *Proceedings of the International Conference on Service-Oriented Computing and Applications (SOCA)*. 2010. (Zitiert auf Seite 40)
- [Töpo7] A. Töpfer. *Betriebswirtschaftslehre: Anwendungs- und prozessorientierte Grundlagen*. Springer, 2007. (Zitiert auf Seite 9)
- [Tsc07] W. Tscheschner. Oryx – Dokumentation, 2007. (Zitiert auf den Seiten 31, 39 und 57)
- [VWB⁺09] S. Vajna, C. Weber, H. Bley, K. Zeman, P. Hehenberger. *CAx für Ingenieure: Eine praxisbezogene Einführung*. Springer, 2009. (Zitiert auf Seite 10)
- [Wag] B. Wagner. DeployOryxOnWindowsOrDebian. <http://code.google.com/p/oryx-editor/wiki/DeployOryxOnWindowsOrDebian>. Abgerufen am 29.02.2012. (Zitiert auf Seite 58)
- [WCL⁺08] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. F. Ferguson. *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More*. Prentice Hall PTR, 2008. (Zitiert auf Seite 10)
- [Wee] D. Weerasiri. How to build ORYX editor in Ubuntu. <http://ddweerasiri.blogspot.com/2009/04/how-to-build-oryx-editor-in-ubuntu.html>. Abgerufen am 29.02.2012. (Zitiert auf Seite 58)

- [WFM95] WFMC. The Workflow Reference Model, 1995. (Zitiert auf Seite 10)
- [WFM05] WFMC. XML Process Definition Language (XPDL) Version 2.0, 2005. (Zitiert auf Seite 39)
- [Wora] World Wide Web Consortium. Extensible Markup Language (XML). <http://www.w3.org/XML/>. Abgerufen am 29.02.2012. (Zitiert auf Seite 16)
- [Worb] World Wide Web Consortium. Hypertext Markup Language (HTML). <http://www.w3.org/MarkUp/>. Abgerufen am 29.02.2012. (Zitiert auf Seite 57)
- [Worc] World Wide Web Consortium. Portable Network Graphics (PNG). <http://www.w3.org/png/>. Abgerufen am 29.02.2012. (Zitiert auf Seite 32)
- [Word] World Wide Web Consortium. Scalable Vector Graphics (SVG). <http://www.w3.org/SVG/>. Abgerufen am 29.02.2012. (Zitiert auf Seite 32)
- [WZo9] E. Westkämper, E. Zahn. *Wandlungsfähige Produktionsunternehmen: Das Stuttgarter Unternehmensmodell*. Springer, 2009. (Zitiert auf Seite 10)
- [ZLS11] S. Zor, F. Leymann, D. Schumm. A Proposal of BPMN Extensions for the Manufacturing Domain. In *Proceedings of the CIRP International Conference on Manufacturing Systems (ICMS)*. 2011. (Zitiert auf den Seiten 11, 13, 14, 21, 38 und 49)

A. Anhang

A.1. README.txt

Die Datei README.txt ist Bestandteil der Implementierung. Sie enthält die zusammengefassten Inhalte des Vorschlags von Zor et al. [ZLS11] der Erweiterung BPMN4Manu.

Sie enthält außerdem die Attribute der neuen Elemente als Properties und dokumentiert die während der Umsetzung im Modellierungswerkzeug Oryx (Kapitel 5) modifizierten und hinzugefügten Dateien. Die Dateien stellen die Implementierung dar und sind Bestandteil dieser Arbeit auf CD-ROM.

README.txt

- = BPMN4Manu
 - Activity Type for Manufacturing Processes
 - * Activities / Manufacturing Task
 - Resource Containers
 - * Resource Containers / Machines & Tools
 - * Resource Containers / Parts
 - Connector Types for the Movement of Material
 - * Connecting Objects / Machines & Tools Flow
 - * Connecting Objects / Parts Flow
 - Material Gateways
 - * Gateways / Material Route
 - * Gateways / Material Select
 - * Gateways / Material Split
 - * Gateways / Material Join
- = Modified and Added Files
 - Modified
 - Added

A. Anhang

=====

= BPMN4Manu

- Activity Type for Manufacturing Processes

* Activities / Manufacturing Task
"Activities" / "ManuTask"

Meaning: Manufacturing processes use machines, tools and labor to produce goods by transforming raw materials into finished goods.

Description: To represent manufacturing processes we introduce a new activity type Manufacturing Task. A Manufacturing Task is an atomic activity within a production process flow. This task is used when the work in the production process cannot be broken down to a finer level of detail. A Manufacturing Task object is a rectangle that has rounded corners. The marker for the Manufacturing Task is a little factory shape located within the rectangle (upper left).

Properties:

- cycle time
- set-up time
- fit tolerance
- machine reliability
- role of involved persons
- number of involved persons

- Resource Containers

* Resource Containers / Machines & Tools
"Resource Containers" / "MachinesNTools"

Meaning: Machines and tools are fixed components of each production facility. These devices are used within manufacturing processes.

Description: For this, we introduce a new element Machines & Tools Container. This construct groups all material objects flowing from or to the related Manufacturing Task. It is represented by a rectangle which can be collapsed to see the involved machines and tools. The container must be connected to a Manufacturing Task by a

Machines & Tools Flow Connector.

Properties:

- identifier
- quantity
- moveable | non-moveable
- safety information
- role of involved persons
- number of involved persons

* Resource Containers / Parts
"Resource Containers" / "Parts"

Meaning: The most important material that flows in a production is the product (or its parts) itself. Depending on the developmental stage of the product, it can be raw material, unfinished parts or goods.

Description: The Parts Container is designed as elliptical shape to represent the part. It can be collapsed to see its concrete and process-specific elements. The Parts Container can either be connected to a Manufacturing Task or a Material Gateway by using Parts Flow Connectors.

Properties:

- identifier
- quantity
- processing status
- product shape information

- Connector Types for the Movement of Material

* Connecting Objects / Machines & Tools Flow
"Connecting Objects" / "MachinesNToolsFlow"

Meaning: As already mentioned, in each production facility we have fixed and moveable machines and tools.

Description: In order to model the movement of these machines and tools, we introduce a Machines & Tools Flow Connector. Each Machines & Tools Flow Connector has exactly one source and exactly one target. The source and target must be of the type activity (task or sub-process) or gateway. A Machines & Tools Flow Connector is a broken line with a solid arrowhead.

A. Anhang

Properties:

- transportation type
 - human
 - machine
- role of involved persons
- number of involved persons
- type of machines and tools
- number of machines and tools

* Connecting Objects / Parts Flow
"Connecting Objects" / "PartsFlow"

Meaning: We need a construct which indicates a transfer of a product material from one source to one target. It represents raw material, parts or finished goods. The material can be transferred by human, via machine, or via using an assembly line.

Description: In order to indicate the "parts flow", we propose a Parts Flow Connector which connects Manufacturing Tasks, Parts or Material Gateways with each other, also across sub-process borders. The Parts Flow Connector is a broken line with an unfilled arrowhead.

Properties:

- material access mechanism
 - push
 - pull (kanban)
 - first-in-first-out
 - last-in-first-out
- transportation type
 - human
 - machine
- role of involved persons
- number of involved persons
- type of machines and tools
- number of machines and tools

- Material Gateways

* Gateways / Material Route
"Gateways" / "MaterialRoute"

Meaning: Basically, this construct is a diversion point in the material flow.

Description: A diverging Material Route Gateway is used when alternative paths within a process could be taken. With this gateway we can model a decision which can be thought of as a question that is asked at a particular point in the production process. A set of answers has to be defined for this question.

Properties:

- routing evaluation mechanism
 - human
 - machine
- role of involved persons
- number of involved persons
- type of machines and tools
- number of machines and tools

* Gateways / Material Select
"Gateways" / "MaterialSelect"

Meaning: In shop floor environments we have the scenario to select one out of multiple materials which has to be routed through the production process.

Description: The Material Select Gateway has multiple incoming flows and one outgoing flow. If there is at least one incoming flow the gateway reacts by giving through this flow.

Properties:

- selecting evaluation mechanism
 - human
 - machine
- role of involved persons
- number of involved persons
- type of machines and tools
- number of machines and tools

* Gateways / Material Split
"Gateways" / "MaterialSplit"

Meaning: We need a construct, which indicates the splitting of material.

Description: The Material Split Gateway creates parallel flows by dividing material into multiple constituent parts. The condition

A. Anhang

specifies how the material has to be split up.

Properties:

- splitting evaluation mechanism
 - human
 - machine
- role of involved persons
- number of involved persons
- type of machines and tools
- number of machines and tools

* Gateways / Material Join
"Gateways" / "MaterialJoin"

Meaning: Joining resources is an indispensable scenario in production environments.

Description: The Material Join Gateway controls incoming resources. It waits for the completion of all incoming flows before triggering the flow through its outgoing flow.

Properties:

- joining evaluation mechanism
 - human
 - machine
- role of involved persons
- number of involved persons
- type of machines and tools
- number of machines and tools

=====

= Modified and Added Files

wsd - "path to workspace directory"

woed=\$wsd/oryx/editor/data
woec=\$wsd/oryx/editor/client
woes=\$wsd/oryx/editor/server
woee=\$wsd/oryx/editor/etc

woedse=\$woed/stencilsets/extensions

- Modified

\$woedse/extensions.json

\$woed/i18n/translation_de.js
\$woed/i18n/translation_en_us.js

\$woec/scripts/config.js
\$woec/scripts/Plugins/profiles.xml
\$woec/scripts/Plugins/plugins.xml

\$woee/web.xml

- Added

\$woedse/bpmn4manu

\$woedse/bpmn4manu/bpmn4manu.json

\$woedse/bpmn4manu/icons
\$woedse/bpmn4manu/icons/activity
\$woedse/bpmn4manu/icons/activity/manuTask.png
\$woedse/bpmn4manu/icons/activity/list
\$woedse/bpmn4manu/icons/activity/list/looptypeLoop.png
\$woedse/bpmn4manu/icons/activity/list/looptypeParallel.png
\$woedse/bpmn4manu/icons/activity/list/looptypeSequential.png
\$woedse/bpmn4manu/icons/artifact
\$woedse/bpmn4manu/icons/artifact/ignitionCoil.png
\$woedse/bpmn4manu/icons/artifact/ignitionPlug.png
\$woedse/bpmn4manu/icons/artifact/wrench.png
\$woedse/bpmn4manu/icons/connectingobject
\$woedse/bpmn4manu/icons/connectingobject/machinesNToolsFlow.png
\$woedse/bpmn4manu/icons/connectingobject/partsFlow.png
\$woedse/bpmn4manu/icons/gateway
\$woedse/bpmn4manu/icons/gateway/materialJoin.png
\$woedse/bpmn4manu/icons/gateway/materialRoute.png
\$woedse/bpmn4manu/icons/gateway/materialSelect.png
\$woedse/bpmn4manu/icons/gateway/materialSplit.png
\$woedse/bpmn4manu/icons/resourcecontainer
\$woedse/bpmn4manu/icons/resourcecontainer/machinesNTools.png
\$woedse/bpmn4manu/icons/resourcecontainer/parts.png

\$woedse/bpmn4manu/view
\$woedse/bpmn4manu/view/activity
\$woedse/bpmn4manu/view/activity/manuTask.svg
\$woedse/bpmn4manu/view/artifact

A. Anhang

```
$woedse/bpmn4manu/view/artifact/ignitionCoil.svg  
$woedse/bpmn4manu/view/artifact/ignitionPlug.svg  
$woedse/bpmn4manu/view/artifact/wrench.svg  
$woedse/bpmn4manu/view/connectingobject  
$woedse/bpmn4manu/view/connectingobject/machinesNToolsFlow.svg  
$woedse/bpmn4manu/view/connectingobject/partsFlow.svg  
$woedse/bpmn4manu/view/gateway  
$woedse/bpmn4manu/view/gateway/materialJoin.svg  
$woedse/bpmn4manu/view/gateway/materialRoute.svg  
$woedse/bpmn4manu/view/gateway/materialSelect.svg  
$woedse/bpmn4manu/view/gateway/materialSplit.svg  
$woedse/bpmn4manu/view/resourcecontainer  
$woedse/bpmn4manu/view/resourcecontainer/machinesNTools.svg  
$woedse/bpmn4manu/view/resourcecontainer/parts.svg  
  
$woec/images/toAbstrBpelXmlIcon.png  
$woec/images/toBpmn2_0DiXmlIcon.png  
$woec/images/toBpmn2_0XmlIcon.png  
$woec/images/toExecBpelXmlIcon.png  
  
$woec/scripts/Plugins/bpmn4manu.js  
  
$woes/src/de/unistuttgart/iaas/bpmn4manu/  
  
*****
```

A.2. Oryx Setup Guide

Der *Oryx Setup Guide* ist eine in HTML [Worb] verfasste und durch Verlinkungen innerhalb des Dokuments navigierbare Installationbeschreibung des Modellierungswerkzeugs Oryx [Tsc07, Czu07, Polo7, Pet07, Has]. Der Einsatz von JavaScript [Int11] ermöglicht das Ein- und Ausblenden von (Unter-)Abschnitten. Zur Verwendung in dieser Arbeit wurde der Oryx Setup Guide nach L^AT_EX [L^AT_EX Project] transformiert.

Content

1. Other Setup Guides

2. Platform

3. Definitions

4. Development

4.1 Tomcat

4.2 Application

4.3 Database

4.4 Build

4.4.1 Eclipse

4.4.2 Ant

5. Production

5.1 Tomcat

5.2 Application

5.3 Database

5.4 Build/Deploy

6. Test

6.1 Backend

6.2 Frontend

1. Other Setup Guides

- SetupDevelopmentEnvironment [Kop]
- How to build ORYX editor in Ubuntu [Wee]
- DeployOryxOnWindowsOrDebian [Wag]
- InstallUpdateConfigureProductiveSystems [Kun]

↑ Content ↑

2. Platform

- Debian [Deb]
- Switch user with
 - su
- In
 - /etc/apt/sources.list
 - add line
 - deb http://ftp-stud.fht-esslingen.de/debian/ squeeze contrib non-free
- apt-get update
- apt-get install ant graphviz postgresql-plpython-8.4 sun-java6-jdk tomcat6-user
- With
 - update-alternatives --config java
 - choose
 - /usr/lib/jvm/java-6-sun/jre/bin/java
- exit

↑ Content ↑

3. Definitions

- wd - working directory
- wsd - workspace directory
- poem-pwd - password of user/role poem
- oryx-pwd - password of user/role oryx

↑ Content ↑

4. Development

4.1 Tomcat

- In

```
/usr/share/tomcat6/bin/startup.sh  
change last line from  
exec "$PRGDIR"/"$EXECUTABLE" start "$@"  
to  
exec "$PRGDIR"/"$EXECUTABLE" jpda start
```

- Change directory with

```
cd $wd
```

- Create instance with

```
tomcat6-instance-create tomcat4oryx_dev
```

- In

```
$wd/tomcat4oryx_dev/conf/server.xml
```

add

```
<Context path="/backend" docBase="$wsd/oryx/dist/backend"  
debug="0" reloadable="true" />
```

and

```
<Context path="/oryx" docBase="$wsd/oryx/dist/client"  
debug="0" reloadable="true" />
```

as child tags of

```
<Host>
```

A. Anhang

- In

```
$wd/tomcat4oryx_dev/bin/startup.sh
```

add line

```
export JPDA_ADDRESS=8001
```

- Start instance with

```
$wd/tomcat4oryx_dev/bin/startup.sh
```

- Later, stop instance with

```
$wd/tomcat4oryx_dev/bin/shutdown.sh
```

↑ Content ↑

4.2 Application

- Change directory with

```
cd $wsd
```

- Checkout Oryx from <http://code.google.com/p/oryx-editor/source/checkout>

- In

```
$wsd/oryx/build.properties
```

change value of

deploymentdir

to

```
$wd/tomcat4oryx_dev/webapps
```

and of

```
postgresql-bin-dir
```

to

```
/usr/lib/postgresql/8.4/bin/
```

- In

```
$wsd/oryx/test.properties
```

change value of

```
localWorkingDirectoryUrl
```

to

```
$wsd/oryx
```

and of

```
tomcat.home  
to  
/usr/share/tomcat6
```

- In

```
$wsd/oryx/poem-jvm/etc/backend.properties  
change value of  
bpmnq.database.password  
to  
$poem-pwd
```

- In

```
$wsd/oryx/poem-jvm/shared/database.yml  
change value of  
password  
to  
$poem-pwd
```

- In

```
$wsd/oryx/editor/etc/database.properties  
change value of  
db.password  
to  
$oryx-pwd  
and of  
db.url  
to  
jdbc:postgresql://localhost/poem  
and of  
db.connector  
to  
org.postgresql.Driver
```

A. Anhang

↑ Content ↑

4.3 Database

- Switch user with

```
su
```

- Switch user with

```
su postgres
```

- Create new user/role with

```
createuser --username=postgres --echo --pwprompt --encrypted poem
```

```
Enter password for new role: $poem-pwd
```

```
Enter it again: $poem-pwd
```

```
Shall the new role be a superuser? (y/n) n
```

```
Shall the new role be allowed to create databases? (y/n) y
```

```
Shall the new role be allowed to create more new roles? (y/n) n
```

- Create new database with

```
createdb --username=postgres --echo --encoding=utf8 --owner=poem poem
```

- Configure database with

```
psql poem < $wsd/oryx/poem-jvm/data/database/db_schema.sql postgres
```

- Create new user/role with

```
createuser --username=postgres --echo --pwprompt --encrypted oryx
```

```
Enter password for new role: $oryx-pwd
```

```
Enter it again: $oryx-pwd
```

```
Shall the new role be a superuser? (y/n) n
```

```
Shall the new role be allowed to create databases? (y/n) n
```

```
Shall the new role be allowed to create more new roles? (y/n) n
```

- exit

- exit

↑ Content ↑

4.4 Build

4.4.1 Eclipse

- Right-Click \$wsd/oryx/poem-jvm/build.xml and then “Run As” → “External Tools Configurations...”
- Double-Click “Ant Build”
 - Name: backend
 - Tab “Main”
 - * Base Directory: \${workspace_loc:/oryx}
 - Tab “Targets”
 - * Check “dist”
 - Click “Run”
- Right-Click \$wsd/oryx/editor/build.xml and then “Run As” → “External Tools Configurations...”
- Double-Click “Ant Build”
 - Name: frontend
 - Tab “Main”
 - * Base Directory: \${workspace_loc:/oryx}
 - Tab “Targets”
 - * Check (in that order) “build”, “examples”, “compress”, and “dist”
 - Click “Run”

↑ Content ↑

4.4.2 Ant

- Change directory with

```
cd $wsd/oryx/poem-jvm
```
- Build with

```
ant dist
```
- Change directory with

```
cd $wsd/oryx/editor
```
- Build with

A. Anhang

```
ant build examples compress dist  
↑ Content ↑
```

5. Production

5.1 Tomcat

- Change directory with

```
cd $wd
```

- Create instance with

```
tomcat6-instance-create tomcat4oryx
```

- Start instance with

```
$wd/tomcat4oryx/bin/startup.sh
```

- Later, stop instance with

```
$wd/tomcat4oryx/bin/shutdown.sh
```

↑ Content ↑

5.2 Application

- Change directory with

```
cd $wsd
```

- Checkout Oryx from <http://code.google.com/p/oryx-editor/source/checkout>

- In

```
$wsd/oryx/build.properties
```

change value of

deploymentdir

to

```
$wd/tomcat4oryx/webapps
```

and of

postgresql-bin-dir

to

```
/usr/lib/postgresql/8.4/bin/
```

- In

\$wsd/oryx/test.properties

change value of

localWorkingDirectoryUrl

to

\$wsd/oryx

and of

tomcat.home

to

/usr/share/tomcat6

- In

\$wsd/oryx/poem-jvm/etc/backend.properties

change value of

bpmnq.database.password

to

\$poem-pwd

- In

\$wsd/oryx/poem-jvm/shared/database.yml

change value of

password

to

\$poem-pwd

- In

\$wsd/oryx/editor/etc/database.properties

change value of

db.password

to

\$oryx-pwd

and of

db.url

A. Anhang

```
to  
jdbc:postgresql://localhost/poem  
and of  
db.connector  
to  
org.postgresql.Driver
```

↑ Content ↑

5.3 Database

- Switch user with

```
su
```

- Switch user with

```
su postgres
```

- Create new user/role with

```
createuser --username=postgres --echo --pwprompt --encrypted poem
```

```
Enter password for new role: $poem-pwd
```

```
Enter it again: $poem-pwd
```

```
Shall the new role be a superuser? (y/n) n
```

```
Shall the new role be allowed to create databases? (y/n) y
```

```
Shall the new role be allowed to create more new roles? (y/n) n
```

- Create new database with

```
createdb --username=postgres --echo --encoding=utf8 --owner=poem poem
```

- Configure database with

```
psql poem < $wsd/oryx/poem-jvm/data/database/db_schema.sql postgres
```

- Create new user/role with

```
createuser --username=postgres --echo --pwprompt --encrypted oryx
```

```
Enter password for new role: $oryx-pwd
```

```
Enter it again: $oryx-pwd
```

```
Shall the new role be a superuser? (y/n) n  
Shall the new role be allowed to create databases? (y/n) n  
Shall the new role be allowed to create more new roles? (y/n) n
```

- exit
- exit

↑ Content ↑

5.4 Build/Deploy

- Change directory with
cd \$wsd/oryx
- Build and Deploy with
ant build-all deploy-all

↑ Content ↑

6. Test

6.1 Backend

- Start Browser and go to <http://localhost:8080/backend/poem/repository>
- Create/Browse/Manage models
- Double-click opens on a model in the frontend

↑ Content ↑

6.2 Frontend

- Start Browser and go to <http://localhost:8080/oryx/editor>
- Modify models
- To store a model you need an OpenID [Ope]

↑ Content ↑

A.3. Tests

Die Abbildungen A.1, A.2 und A.3 zeigen die Ergebnisse von Tests der Regeln aus dem Listing 5.4.

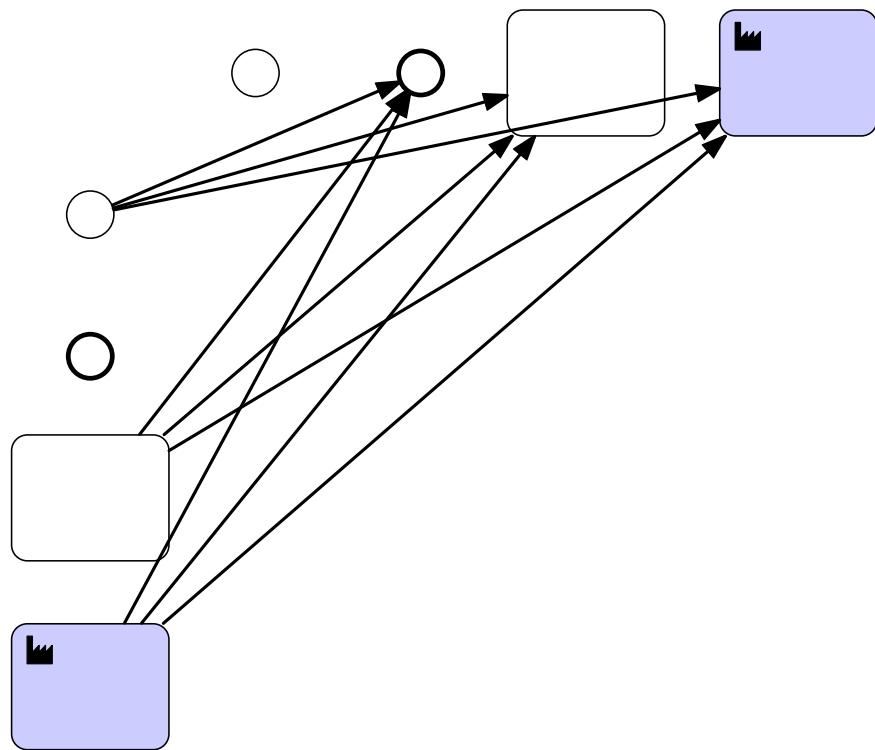


Abbildung A.1.: Test der Sequence Flow connectionRules

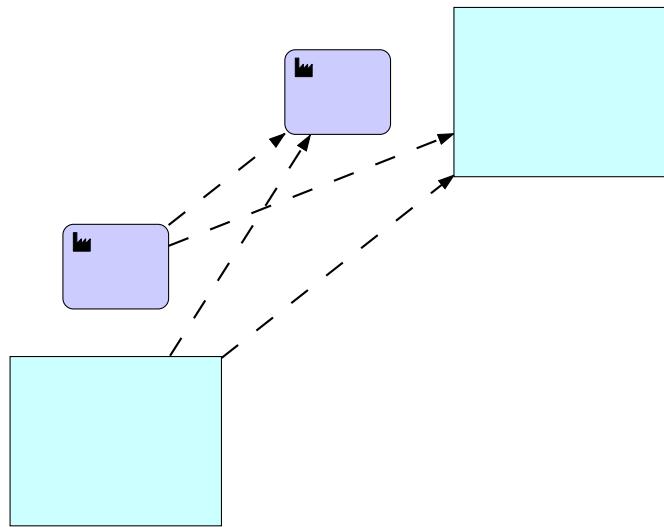


Abbildung A.2.: Test der Machines & Tools Flow connectionRules

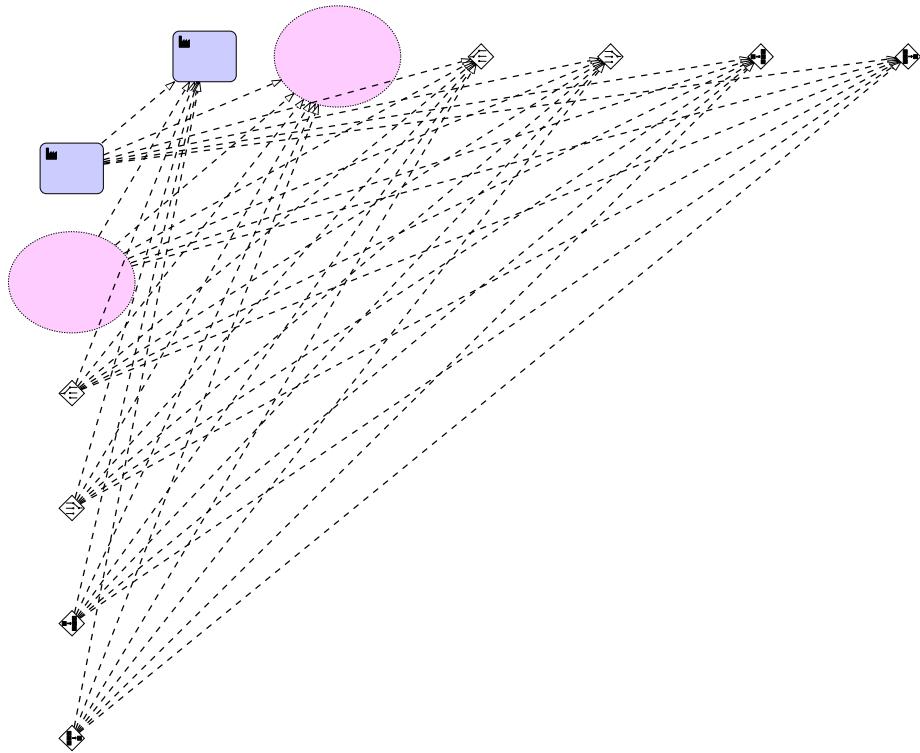


Abbildung A.3.: Test der Parts Flow connectionRules

Erklärung

Hiermit versichere ich, diese Arbeit selbständig verfasst und nur die angegebenen Quellen benutzt zu haben.

(Monir Zouaoui)