

Fachbereich Wirtschaft, Informatik, Recht  
Studiengang Wirtschaftsinformatik

Bachelorarbeit

**Thema (deutsch):** Generierung von Geschäftsprozesssimulationsmodellen  
aus einem BPMN Modellierungswerkzeug  
– Eine Machbarkeitsstudie

**Thema (englisch):** Generating of business process simulation models from  
an BPMN modelling tool  
– A feasibility study

zur Erlangung des akademischen Grades

Bachelor of Science  
(B. Sc.)

**Author:** Sebastian Nemark  
**Matrikelnr.:** 0112770509  
**Seminargruppe:** I1/11

**1. Gutachter** Prof. Dr. Christian Müller  
**2. Gutachter** Prof. Dr. Klaus D. Bösing

**Reg.-Nr.:** I 151/14  
**Eingereicht am:** 14.07.2014



## **Zusammenfassung**

Diese Bachelorarbeit beschäftigt sich mit der Frage, wie sich mit Hilfe von quelloffenen Business Process Model and Notation (BPMN) Modellierungswerkzeugen Simulationsparameter in Geschäftsprozessmodelle integrieren lassen.

Diese Fragestellung ist heutzutage für prozessorientierte Unternehmen und Organisationen interessant. Beim initial Design, sowie häufigen Anpassungen von Geschäftsprozessen, sind die Ergebnisse kaum vorhersehbar. Fehler im produktiven Betrieb können somit hohe sichtbare und unsichtbare Kosten hervorrufen. Die Generierung von Geschäftsprozesssimulationsmodellen ist einer der ersten Schritte zur Simulation, wodurch Risiken minimiert werden können.

Das Ziel dieser Arbeit ist eine Machbarkeitsstudie. In dieser wird herausgefunden, welche quelloffenen BPMN Modellierungswerkzeuge bereits die Generierung von Geschäftsprozesssimulationsmodellen unterstützen. Ist dies nicht der Fall, so werden Möglichkeiten aufgezeigt, diese Simulationsparameter in die Geschäftsprozessmodelle zu integrieren.

## **Abstract**

This bachelor thesis concerned the question, how simulation parameter in business process model with open source Business Process Model and Notation (BPMN) modeling tools are defined.

This question is interesting for process oriented companies and organizations. The initial design and frequent changes of business process models has sometimes unexpected results. This errors are in many cases expensive. One of the first steps to simulate business process models are the model extending with simulation parameters.

The aim of this study is a feasibility study. This figure out which open source BPMN tool support the extending of simulation parameters. If this is not the case are shown how they can integrate simulation parameters.



# Inhaltsverzeichnis

<b>Zusammenfassung</b>	<b>III</b>
<b>Abbildungsverzeichnis</b>	<b>VII</b>
<b>Tabellenverzeichnis</b>	<b>IX</b>
<b>Quellcodeverzeichnis</b>	<b>XI</b>
<b>Abkürzungsverzeichnis</b>	<b>XIII</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Zielsetzung . . . . .	2
1.2 Aufbau . . . . .	2
1.3 Abgrenzung . . . . .	3
<b>2 Grundlagen</b>	<b>5</b>
2.1 Geschäftsprozess . . . . .	5
2.2 Workflow . . . . .	6
2.3 Business Process Management . . . . .	7
2.3.1 ereignisgesteuerte Prozesskette . . . . .	9
2.3.2 Business Process Model and Notation . . . . .	12
2.4 Business Process Simulation . . . . .	13
2.4.1 BPSim . . . . .	14
2.4.2 EPC-Simulator . . . . .	16
<b>3 Auswahl der Modellierungswerkzeuge</b>	<b>19</b>
3.1 Recherche . . . . .	19
3.2 Auswahl . . . . .	22
<b>4 Modellierungswerkzeuge</b>	<b>25</b>
4.1 Voraussetzungen . . . . .	25
4.2 jBPM . . . . .	26
4.2.1 Eclipse Developer Tools . . . . .	27
4.2.2 Process Designer . . . . .	37
4.3 Activiti . . . . .	42
4.3.1 Activiti Designer . . . . .	43
4.3.2 Activiti Designer Erweiterung . . . . .	45
4.3.3 Crystalball . . . . .	50
4.4 Camunda . . . . .	53
4.4.1 Camunda Modeler . . . . .	54
4.4.2 BPMN Model API . . . . .	57
4.5 Alternativen . . . . .	59
4.5.1 Yaoqiang . . . . .	59

4.5.2	BIMP . . . . .	61
<b>5</b>	<b>Fazit</b>	<b>67</b>
5.1	Bewertung . . . . .	69
5.2	Offene Punkte . . . . .	70
	<b>Literaturverzeichnis</b>	<b>XIII</b>
<b>A</b>	<b>Anhang</b>	<b>XVII</b>
A.1	Inhalt der CD . . . . .	XVII

# Abbildungsverzeichnis

1.1	Geschäftsziele mittelständischer Unternehmen 2011 (Statista 2014)	1
2.1	BPM Lebenszyklus (Dumas, Rosa, Mendling & Reijers 2013, S. 21)	8
2.2	Historische Entwicklung wichtiger BPM Modellierungsstandards (Bartonitz 2009)	10
2.3	Beispiel Dienstreiseantrag (eEPK)	11
2.4	Beispiel Dienstreiseantrag (BPMN)	13
2.5	BPSim Scope (WfMC 2012, S.5)	15
2.6	EPC-Simulator Architektur und Logo (Müller 2013, S. 1)	17
4.1	jBPM Übersicht (jBPM team 2014a)	27
4.2	jBPM Eclipse Plugins installieren	28
4.3	jBPM Eclipse Plugin Auswahl	29
4.4	jBPM Eclipse Einstellungen	30
4.5	jBPM neues Projekt anlegen	30
4.6	jBPM Perspektive	31
4.7	jBPM / Eclipse BPMN2 Modeler Diagrammtypen	31
4.8	jBPM / Eclipse BPMN2 Modeler Palette (links v1.0.2, rechts v1.1.0)	32
4.9	jBPM / Eclipse BPMN2 Modeler Kontextmenü	33
4.10	jBPM / Eclipse BPMN2 Modeler Simulationsparameter	33
4.11	jBPM / Eclipse BPMN2 Modeler Fehlermeldung beim Öffnen der externen Datei	33
4.12	jBPM / Eclipse BPMN2 Modeler Anzeige externer Datei	34
4.13	jBPM / Eclipse BPMN2 Modeler Extension Aktionsaufruf	35
4.14	jBPM / Eclipse BPMN2 Modeler Extension Einstellungen	36
4.15	jBPM installer Start	38
4.16	jBPM Process Designer Anmeldung	38
4.17	jBPM Process Designer Prozessmodell	39
4.18	jBPM Process Designer Export	40
4.19	jBPM Process Designer Simulation starten	40
4.20	jBPM Process Designer Simulation - Laufzeitparameter	41
4.21	jBPM Process Designer Simulationsergebnisse	41
4.22	Activiti Komponentenübersicht (Activiti n.d.)	42
4.23	Activiti Designer Repository Plugin Installation	44
4.24	Activiti Designer Beispielprojekt	44
4.25	Activiti Designer Anzeige externer Datei	45
4.26	Activiti Designer Integration der Palettenerweiterung	48
4.27	Activiti Designer Extension Palette Customizing	48
4.28	Activiti Designer Extension Validierung und Export	49
4.29	Activiti Crystalball Testabdeckung	51

4.30	Activiti Crystalball Eskalationswahrscheinlichkeiten . . . . .	52
4.31	Camunda Architekturübersicht (Camunda 2014) . . . . .	53
4.32	Camunda Modeler Ansicht . . . . .	55
4.33	Camunda Modeler Fehler beim öffnen der externer Datei . . . . .	56
4.34	Camunda Modeler externe Datei geöffnet . . . . .	56
4.35	Yaoqiang Anwendung Version 2.2.18 . . . . .	60
4.36	Yaoqiang Fehlermeldung beim Öffnen . . . . .	60
4.37	BIMP Simulator Prozessmodell hochladen . . . . .	62
4.38	BIMP Simulator Kategorien . . . . .	63
4.39	BIMP Simulator Eingaben . . . . .	64
4.40	BIMP Simulator Ergebnisse . . . . .	64
4.41	BIMP Simulator auf eigenem Computer . . . . .	66



# Tabellenverzeichnis

2.1	erweiterte / EPK Elemente . . . . .	11
2.2	BPMN Basiselemente . . . . .	13
3.1	Internetecherche nach Modellierungswerkzeuge . . . . .	21
4.1	Prototypische Eclipse BPMN2 Modeler und jBPM Projekte . .	35
5.1	Übersicht der Modellierungswerkzeugbewertung . . . . .	70



# Quellcodeverzeichnis

2.1	Evaluation.v1.bpmn2 . . . . .	15
4.1	ProcessMain.java . . . . .	30
4.2	Ausgabe der Sample.bpmn2 Elemente . . . . .	35
4.3	Ausschnitte der Ausgabe der jBPM Simulation . . . . .	36
4.4	jBPM installer Kommandos . . . . .	37
4.5	Activiti Engine JUnit test case - ProcessTestActivitiReview.java	44
4.6	Activiti Designer Palette Erweiterung - pom.xml . . . . .	46
4.7	Activiti Designer Palette Erweiterung - PaletteCustomizer.java .	46
4.8	Activiti Designer Palette Erweiterung - ThWildauTask.java . . .	47
4.9	Activiti Designer VE Erweiterung - ThProcessValidator.java . .	49
4.10	Activiti Crystalball Kommando mvn clean install unter Mac OS	50
4.11	Camunda Model API - FluentApi.java . . . . .	57
4.12	fluentApiProcess.bpmn . . . . .	57
4.13	Auszug aus ModelApiExtElement.java . . . . .	58
4.14	Erweitertes Prozessmodell - modelApiExtElementProcess.bpmn	58
4.15	Yaoqiang Fehlermeldung der Simulation . . . . .	60
4.16	BIMP Simulator Prozessmodell-Datei . . . . .	64



# Abkürzungsverzeichnis

<b>ARIS</b>	Architektur Integriertes Informationssystem
<b>AGPL</b>	GNU Affero General Public License
<b>API</b>	Application Programming Interface
<b>AS</b>	Application Server
<b>BPEL</b>	Business Process Execution Language
<b>BPD</b>	Business Process Diagram
<b>BPM</b>	Business Process Management
<b>BPR</b>	Business Process Reengineering
<b>BPS</b>	Business Process Simulation
<b>BPE</b>	Business Process Engine
<b>BPMN</b>	Business Process Model and Notation
<b>BPMS</b>	Business Process Management System
<b>BPR</b>	Business Process Reengineering
<b>BPRD</b>	Business Process Re-Design
<b>BPMN MIWG</b>	BPMN Model Interchange Working Group
<b>CPI</b>	Continuous Process Improvement
<b>CDI</b>	Contexts and Dependency Injection
<b>CR</b>	Candidate for Release
<b>DESMO-J</b>	Discrete-Event Simulation Modelling in Java
<b>EPL</b>	Eclipse Public License
<b>EPK</b>	ereignisgesteuerte Prozesskette
<b>EPC</b>	Event-driven Process Chain
<b>GPL</b>	GNU General Public License
<b>LGPL</b>	GNU Lesser General Public License
<b>IT</b>	Informationstechnologie

<b>Java EE</b>	Java Enterprise Edition
<b>KPI</b>	Key Performance Indicator
<b>JPPF</b>	Java Parallelize Processing Framework
<b>jBPM</b>	JBoss Business Process Management
<b>JSON</b>	JavaScript Object Notation
<b>OMG</b>	Object Management Group
<b>MIT</b>	Massachusetts Institute of Technology
<b>PDCA</b>	plan-do-check-act
<b>REST</b>	Representational state transfer
<b>SLA</b>	Service-level agreement
<b>SOA</b>	Service-Oriented Architecture
<b>UML</b>	Unified Modeling Language
<b>URL</b>	Uniform Resource Locator
<b>WfMC</b>	Workflow Management Coalition
<b>WS</b>	Web Service
<b>XML</b>	Extensible Markup Language
<b>XPDL</b>	XML Process Definition Language
<b>XSD</b>	XML Schema Definition

# 1 Einleitung

In der heutigen Gesellschaft des 21. Jahrhunderts arbeiten im Hintergrund eine Vielzahl von IT-Systemen. Hierbei werden die wirtschaftlichen Aktionen häufig durch Prozesse abgebildet. Die verstärkte Prozessorientierung ist sowohl in Unternehmen als auch in der Verwaltung zu finden (Bundesverwaltungsamt n.d.). Es entstehen neue Prozesse und bestehende werden stetig optimiert, um effizienter zu wirtschaften und den steigenden Anforderungen an Flexibilität gerecht zu werden. Zudem verstärkt sich dieser Trend aufgrund sich schnell ändernder Marktsituationen und kürzer werdender Produktlebenszyklen. Für die Erzeugung der Produkte und Erbringung der Dienstleistungen verfolgt das Unternehmen meist klassische Ziele wie Kostenminimierung, Qualitätssicherung oder hohe Lieferbereitschaft. Es wird hier angenommen, dass diese Ziele für alle Unternehmensgrößen gelten. Die Abbildung 1.1 verdeutlicht dies.

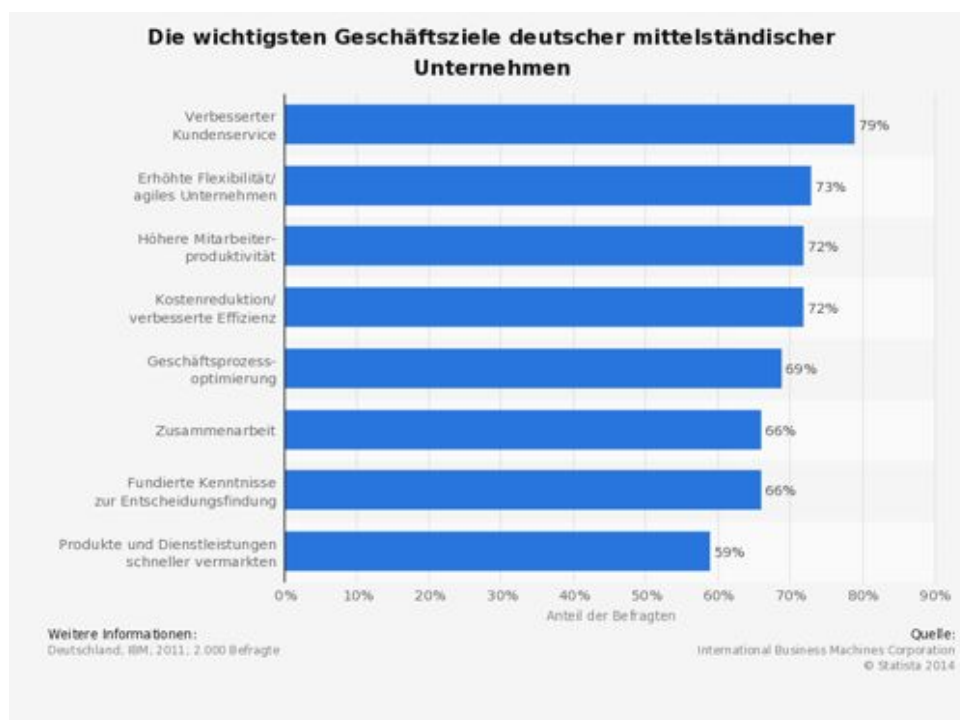


Abbildung 1.1: Geschäftsziele mittelständischer Unternehmen 2011 (Statista 2014)

Um diese Ziele zu erreichen, werden stetig mehr Beteiligte in den Prozess integriert. Ein Beispiel hierfür ist die immer stärker werdende Einbindung der Liefere

ranten. Kundenwünsche und -forderungen können dadurch schneller umgesetzt werden. Gegebenenfalls erhält der Kunde das Produkt oder die Dienstleistung zu günstigeren Konditionen. Durch die Weitergabe der Kostenreduktionen, wird ein höherer Marktanteil angestrebt.

Diese komplexen Prozesse werden durch Business Process Management (BPM) verwaltet. Mit Hilfe des Business Process Management Systems (BPMS) werden die Prozesse modelliert, verwaltet, ausgeführt und überwacht. Die Modellierung erfolgt viele Jahre mit dem defakto Standard der ereignisgesteuerten Prozesskette (EPK). Dieser Trend ändert sich seit einigen Jahren hin zur Verwendung von Business Process Model and Notation (BPMN). Diese Modellierungsmethode ist international standardisiert und soll die Lücke zwischen der fachlichen Prozessdokumentation und dem möglichst vollständig automatisierten Geschäftsprozess schließen.

## 1.1 Zielsetzung

Diese Arbeit entstand in Zusammenarbeit mit Prof. Dr. Müller. Er und seine Mitarbeiter entwickelten den EPC-Simulator<sup>1</sup>, welcher es ermöglicht ereignisgesteuerte Prozessketten mit zusätzlichen Daten für die Simulation anzureichern und im Anschluss zu simulieren.

Aufgrund der wachsenden Popularität von Business Process Model and Notation (BPMN) (vgl. Freund & Rücker 2012, S. XIII) in Unternehmen und Verwaltungen werden folgende drei Punkte analysiert.

Welche Quelloffenen BPMN Modellierungswerkzeuge am Markt existieren, welches deren Simulationsunterstützung ist und ob bzw. wie sich die Geschäftsprozessmodelle mit Simulationsparametern erweitern lassen. Mit prototypischen Entwicklungen werden Möglichkeiten gezeigt wie Geschäftsprozesssimulationsmodelle generiert werden könnten. Ein mögliches Vorgehen ist das Konzept des EPC-Simulators auf ein BPMN Modellierungswerkzeug zu übertragen.

Durch diese Arbeit erhält der Leser sowohl einen Überblick, welche freien Business Process Model and Notation Modellierungswerkzeuge existieren, ob sie eine Simulationsfunktion besitzen und wie sich mit diesen Modellierungswerkzeugen Simulationsparameter in Geschäftsprozessmodelle integrieren lassen.

---

<sup>1</sup><http://www.th-wildau.de/cmueeller/EpcSimulator/>



## 1.2 Aufbau

Nach dieser Einleitung erfolgt im Kapitel 2 „Grundlagen“ die Einführung in die Thematik der Geschäftsprozesse. Dies umfasst den Geschäftsprozess und Workflow, das Business Process Management und die Business Process Simulation. Das Kapitel Business Process Management beinhaltet die zwei grafischen Modellierungsmethoden Ereignisgesteuerte Prozesketten und Business Process Model and Notation. Das Kapitel Business Process Simulation beschreibt zudem den BPSim Standard und den EPC-Simulator.

Kapitel 3 beschreibt die Auswahl der Modellierungswerkzeuge.

Die ausgewählten Modellierungswerkzeuge werden in Kapitel 4 erläutert und anhand der Vergleichskriterien untersucht. Dies umfasst u.a. in wie weit es möglich ist Prozessmodelle mit zusätzlichen Daten für die Simulation anzureichern und ggf. eine Simulation durch zu führen. Weiterhin wird betrachtet wie sich Prozessmodelle mit Simulationsparametern anreichen lassen.

Das abschließende Kapitel 5 fasst die erarbeiteten Ergebnisse zusammen. Darüber hinaus werden Kritikpunkte und mögliche zukünftige Anknüpfungspunkte dargelegt.

## 1.3 Abgrenzung

Neben den theoretischen Grundlagen werden die beiden Modellierungsmethoden EPK und BPMN erläutert. In der Beschreibung der EPK wird ausschließlich auf die Notation eingegangen. Auf andere Modellierungsmethoden, wie z.B. Unified Modeling Language (UML), Business Process Execution Language (BPEL) und XML Process Definition Language (XPDL) wird nicht eingegangen. Des weiteren werden keine weiteren Optimierungsmethoden von Geschäftsprozessen erläutert.

In dieser Arbeit werden open source (deutsch quelloffen) lizenzierte Modellierungswerkzeuge betrachtet und hinsichtlich ihrer Funktionalität Geschäftsprozessmodelle mit Simulationsparametern anzureichern, untersucht. Unter der open source Lizenzierung wird hier die kostenfreie Verfügbarkeit, Nutzung und Anpassung des Quellcodes verstanden. Eine solche Lizenz ist beispielsweise die GPL<sup>2</sup> oder Apache Lizenz<sup>3</sup>. Durch die Verfügbarkeit des Quellcodes soll vorzugsweise die Möglichkeit gegeben sein eigene Erweiterung zu integrieren. Insbesondere die mögliche Unabhängigkeit des Geschäftsprozessmodelles und der

<sup>2</sup><https://www.gnu.org/licenses/licenses.html>

<sup>3</sup><http://www.apache.org/licenses/>

Modellierungswerkzeuge werden betrachtet. Die möglichen Implementierungen sind rein prototypisch. Das darauffolgende Ausführen der Simulation ist nicht Bestandteil dieser Arbeit.

In dieser Arbeit wird der Begriff Prozessmodell als synonym für das Geschäftsprozessmodell verwendet.

## 2 Grundlagen

In diesem Kapitel werden die Grundlagen des Geschäftsprozesses und dessen Management erläutert. Zunächst werden die grundlegenden Begriffe des Geschäftsprozesses und Workflows definiert und deren Abgrenzung aufgezeigt. Im Anschluss werden das Geschäftsprozess- und Workflowmanagement, sowie die beiden Modellierungsmethoden ereignisgesteuerte Prozesskette (EPK) und BPMN erklärt. Im letzten Teil der Grundlagen wird die Geschäftsprozesssimulation erläutert.

### 2.1 Geschäftsprozess

Das Wort Prozess stammt von dem lateinischen Wort „processus“ und kann mit „das Fortschreiten“, „Fortgang“ oder „Verlauf“ übersetzt werden (vgl. Dudenverlag 2013a). Der Prozess als solches ist eine inhaltlich abgeschlossene, zeitlich und sachlogische Folge von Aktivitäten. In der Literatur existieren eine Vielzahl von Definitionen für den Geschäftsprozess. (vgl. Gadatsch 2012, S. 41)

Aus der Sicht des Business Process Reengineering (BPR) definiert Hammer und Champy „...den Unternehmensprozess als eine Menge von Aktivitäten, für die ein oder mehrere unterschiedliche Inputs benötigt werden und die für den Kunden ein Ergebnis von Wert erzeugen ...“ (Gadatsch 2012, S. 40). Der Geschäftsprozess ist auf die Unternehmensziele ausgerichtet. Er beschreibt die strategische bzw. fachlich-konzeptionelle Ebene und damit die nicht technischen Implementierungen der Wertschöpfungskette. In dieser Arbeit wird folgende Definition des Geschäftsprozesses zugrunde gelegt.

„Ein Geschäftsprozess ist eine zielgerichtete, zeitlich-logische Abfolge von Aufgaben, die arbeitsteilig von mehreren Organisationen oder Organisationseinheiten unter Nutzung von Informations- und Kommunikationstechnologien ausgeführt werden können. Er dient der Erstellung von Leistungen entsprechend den vorgegebenen, aus der Unternehmensstrategie abgeleiteten Prozesszielen. Ein Geschäftsprozess kann formal auf unterschiedlichen Detaillierungsebenen und

aus mehreren Sichten beschrieben werden. Ein maximaler Detaillierungsgrad der Beschreibung ist dann erreicht, wenn die ausgewiesenen Aufgaben je in einem Zug von einem Mitarbeiter ohne Wechsel des Arbeitsplatzes ausgeführt werden können.“ (Gehring & Gadatsch 1999, S. 3)

## 2.2 Workflow

Ebenso wie der Begriff des Geschäftsprozesses wird der Workflow (englisch für Arbeitsablauf oder -fluss) unterschiedlich definiert. Von der Workflow Management Coalition (WfMC) sowie von den meisten anderen Begriffsbeschreibungen lässt sich ableiten, dass der Workflow einen Geschäftsprozess zumindest teilweise technisch unterstützt und dessen Ausführung genauer spezifiziert. Diese Spezifikation umfasst die Fachlichkeit, die Zeit und die verwendeten Ressourcen, um die einzelnen Arbeitsschritte zu definieren und automatische Verarbeitung werden können (vgl. Gadatsch 2012, S. 46 f.). Dieser Arbeit liegt folgende Definition des Workflows zugrunde.

„Ein Workflow ist 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. Die hierbei anzustoßenden Arbeitsschritte sind zur Ausführung durch Mitarbeiter oder durch Anwendungsprogramme vorgesehen. Von dem Workflow als Typ oder Schema eines (teil-) automatisierten Arbeitsablaufes zu unterscheiden ist eine Workflow-Instanz, die eine konkrete Ausführung des Workflows bezeichnet.“ (Gehring & Gadatsch 1999, Seite 4)

Das Workflow-Modell beschreibt die Abfolge der Workflow-Schritte. Hingegen wird die konkrete Ausführung eines Workflows als Workflow-Instanz bezeichnet. Zudem lässt sich ein Workflow in die 4 Typen Ad-hoc-, Collaborative-, Administrative- und Production-Workflow unterteilen.

Der Ad-hoc-Workflow beschreibt einmalige und stark variierende Prozesse mit Projektcharakter. Der Collaborative-Workflow unterstützt die gemeinsame Erstellung eines Ergebnisses, bei dem die Vernetzung der Mitarbeiter im Vordergrund steht. Der Administrative-Workflow beinhaltet klar strukturierte Routineprozesse, die meist nicht zeitkritisch und preiswert sind. Der Production-Workflow beschreibt fest strukturierte und definierte Vorgänge, die zeitkritisch

und von strategischer Bedeutung sind. Mit dieser Einteilung lässt sich ein Workflow auf seine Eignung für ein Workflow-Management-System überprüfen. Denn je vorhersehbarer und standardisierter der Geschäftsprozess ist, desto eher kann er vom Workflow-Management-System unterstützt werden. Die Eignung ist jedoch auch vom nötigen Implementierungsaufwand abhängig und unterscheidet sich somit von Fall zu Fall. Ein Workflow muss nicht immer vollständig automatisiert sein. (vgl. Gadatsch 2012, S. 49 f.)

Mit dem Geschäftsprozess vergleichend lassen sich folgende Aussagen treffen. Der Geschäftsprozess beschreibt „was“ zu tun ist, um die Geschäftsstrategie umzusetzen. Der Workflow beschreibt „wie“ dies umgesetzt werden soll. Des weiteren ist der Geschäftsprozess der fachlich-konzeptionellen Ebene, der Workflow hingegen der operativen Ebene zuzuordnen. Welcher Modellierungsgrad erreicht ist, lässt sich zwischen der Ausführbarkeit der Schritte beschreiben. Bei dem Geschäftsprozess ist dies der Mitarbeiter und beim Workflow eine Software. (vgl. Gadatsch 2012, S. 52 f.)

## **2.3 Business Process Management**

Beim Business Process Management (BPM, deutsch Geschäftsprozessmanagement) handelt es sich um eine Methodik zum Management der Geschäftsprozesse in einer Organisation. Es beruht auf der Entwicklung, Führung, Organisation und Kontrolle der zielgerichteten Steuerung von Geschäftsprozessen. Das BPM umfasst grundsätzlich Entscheidungen und Sachverhalte. Hierzu zählen beispielsweise die Unternehmenskultur und -ziele, das Geschäftsverständnis, sowie die Personalführung und -bildung. Der Geschäftsprozess wird heutzutage ganzheitlich betrachtet bzw. als Managementdisziplin verstanden. Wesentliche Ziele des Geschäftsprozessmanagements sind das Erreichen der Unternehmensziele, Steigerung der Kundenzufriedenheit, Mitarbeitermotivation, Erhöhung der Produktivität, Steigerung der Qualität, Reduzierung der Zeit und Senkung der Kosten. (vgl. Gadatsch 2012, S. 53 ff.)

Für das Erreichen dieser Ziele werden Business Process Management Systems (BPMS) eingesetzt. Das BPMS führt Geschäftsprozesse in einer Business Process Engine (BPE) aus und unterstützt den gesamten Lebenszyklus des Geschäftsprozesses. Zu diesem Lebenszyklus gehört die Modellierung, Ausführung und Analyse. Die Betrachtung der Geschäftsprozesse erfolgt im BPMS übergreifend. Bei der Ausführung der Geschäftsprozesse werden häufig wichtige Informationen erfasst. Diese Informationen sind u.a. für die strategische Führung und die Simulation interessant.

Das Workflow-Management ist der operative Bestandteil des BPM. Es handelt sich dabei um anwendungsunabhängige Middlewaresysteme das heißt, dass häufig verschiedene Hard- und Software eingebunden werden. Das Workflow-Management stellt zudem Methoden und Werkzeuge bereit, um mit Hilfe von Informationstechnologie (IT) die Arbeitsschritte zu planen, zu analysieren, zu gestalten, zu steuern und zu überwachen. Modellierungsmethoden wie beispielsweise EPK und BPMN definieren Prozesse. Ein Prozess beschreibt die einzelnen Arbeitsschritte. Jeder Prozess hat einen iterativen Lebenszyklus. Der Lebenszyklus basiert auf dem plan-do-check-act (PDCA) Kreislauf von Deming und wird von den verschiedenen Business Process Management Systems im unterschiedlichem Detailgrad umgesetzt. Die Abbildung 2.1 verdeutlicht einen Lebenszyklus.

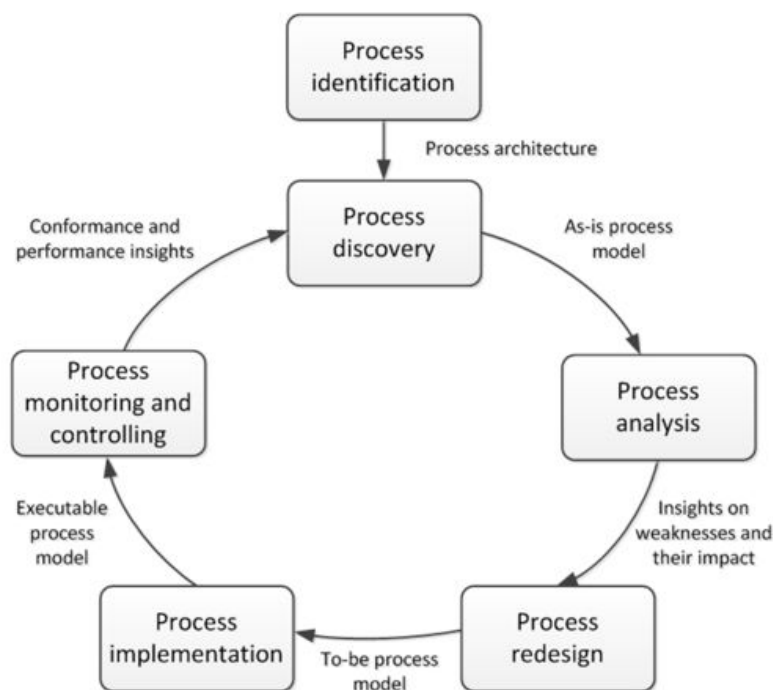


Abbildung 2.1: BPM Lebenszyklus (Dumas et al. 2013, S. 21)

In der ersten Phase wird der Geschäftsprozess analysiert, um Schwachstellen zu identifizieren. Das Ergebnis gibt Aufschlüsse über mögliche Prozessmodelländerungen und liefert Messwerte für die Simulation. Wird ein Prozessmodell neu entworfen, entsteht dieses in der design Phase. Bei existierenden Prozessen wird durch Änderungen versucht die analysierten Schwachstellen zu eliminieren. Die Analyse der Schwachstellen kann u.a mit Hilfe gewonnener Prozessdaten und verschiedenen Optimierungsverfahren erfolgen. In der Umsetzungsphase wird das Prozessmodell mit Hilfe der IT erweitert. Diese Erweiterung dient der Automatisierung und der Anbindung anderer Hard- und Softwaresysteme. In der Ausführungsphase wird der Prozess veröffentlicht und kann auf einem BPMS

bzw. einer BPE produktiv ausgeführt werden. Während der Überwachung laufender Prozesse können u.a. Service-level agreements (SLA) überprüft und Key Performance Indicators (KPI) erfasst werden. Die erfassten Daten können für spätere Simulationen genutzt werden. SLAs (deutsch Dienstleistungsvereinbarungen) sind Vereinbarungen zwischen Auftraggeber und Auftragnehmer. Ziel dieser Vereinbarung ist es, Leistungseigenschaften, wie beispielsweise Reaktionszeiten oder Leistungsumfang, transparent und kontrollierbar zu gestalten. KPIs (deutsch Leistungskennzahlen) sind Kennzahlen an denen Zielsetzungen gemessen werden können. Typische Kennzahlen sind beispielsweise Durchlaufzeiten, Fehlerraten, Umsatz pro Kunde und Termintreue. An dem BPM Lebenszyklus sind folgende Personengruppen Prozessanalyst, Prozessarchitekt, Prozessentwickler und Prozessinhaber beteiligt.

Für die Beschreibung bzw. Modellierung von Geschäftsprozessen existiert eine Vielzahl verschiedener Methoden. Die Abbildung 2.2 verdeutlicht wichtige Modellierungsstandards im Business Process Management (BPM). Die zwei grafischen und Kontrollfluss-orientierten Modellierungsmethoden ereignisgesteuerte Prozesskette (EPK) von IDS Scheer<sup>1</sup> und Business Process Model and Notation (BPMN) von der Object Management Group (OMG)<sup>2</sup> werden in den Kapiteln 2.3.1 und 2.3.2 erläutert.

### 2.3.1 ereignisgesteuerte Prozesskette

Die ereignisgesteuerte Prozesskette (EPK, englisch Event-driven Process Chain (EPC)) ist ein grafischer Modellierungsansatz. Er wurde 1991 an der saarländischen Universität von Keller, Nüttgens und Scheer entwickelt. Ziel war die Entwicklung einer Geschäftsprozessdokumentation. Dies wurde im Bezug zum Architektur Integriertes Informationssystem (ARIS)-Konzept umgesetzt. Das ARIS-Konzept ist ein Rahmenwerk zur Beschreibung von Unternehmen und betriebswirtschaftlichen Anwendungssystemen.

Die EPK Grundelemente sind Ereignisse, Funktionen und Operatoren. Diese Elemente werden mit gerichteten Verbindungslinien und -pfeilen verbunden. Die erweiterte Form der Modellierungsmethode stellt die erweiterte ereignisgesteuerte Prozesskette (eEPK) dar. Diese ergänzt die Elemente Organisationseinheit, Informationsobjekt, Anwendungssystem und Prozessschnittstelle. Die Tabelle 2.1 stellt die grafischen Elemente dar.

---

<sup>1</sup>IDS Scheer Consulting GmbH, <http://www.ids-scheer-consulting.de/de>

<sup>2</sup><http://www.omg.org/>

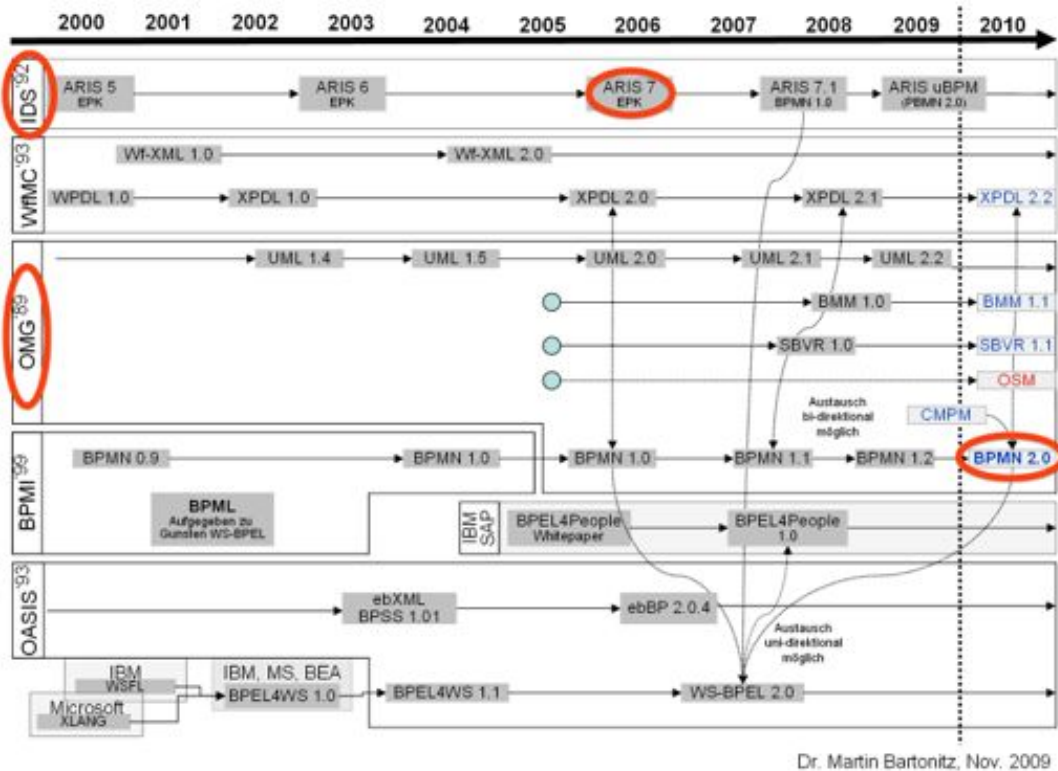


Abbildung 2.2: Historische Entwicklung wichtiger BPM Modellierungsstandards (Bartonitz 2009)

Notation	Element	Bedeutung
	Ereignis	Beschreibung eines Zustandes
	Funktion	Beschreibung einer Transformation von Input zu einem Output
	Logische Operatoren (und, oder, exklusives oder)	Beschreibung logische Verknüpfungen von Ereignissen und Funktionen
	Kontrollfluss	Beschreibung den Ablauf
erweiterte EPK		
	Organisation	Beschreibung der Struktur des Unternehmens
	Information	Beschreibung eines Informationstragendes Objektes
	Anwendungssystem	Beschreibung eines Prozessunterstützendensystemes
	Datenfluss	Beschreibung ob Daten gelesen oder geschrieben werden





Notation	Element	Bedeutung
	Zuordnung	Beschreibung einer Ressourcen Zuordnung
	Prozesswegweiser	Beschreibung einer horizontalen Prozessverknüpfung

Tabelle 2.1: erweiterte / EPK Elemente

Ein Beispiel für ein eEPK Prozessmodell ist in Abbildung 2.3 dargestellt. Es handelt sich um einen Dienstreiseantrag. Dieser wird von einem Mitarbeiter gestellt und von einer Managementposition bearbeitet.

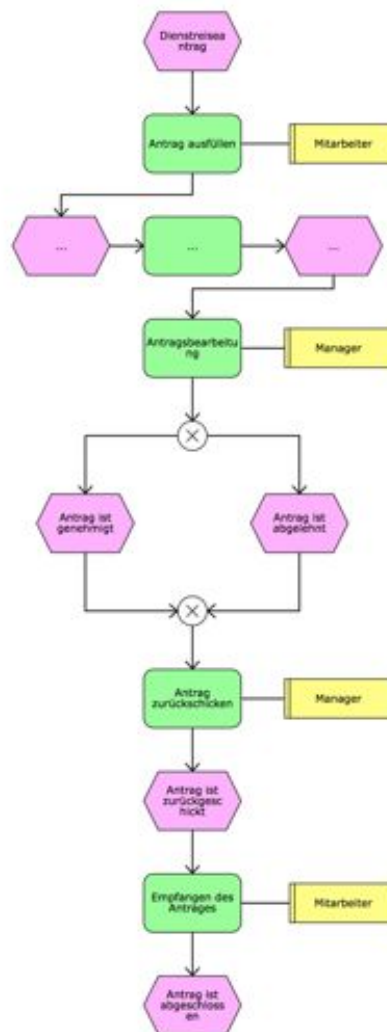


Abbildung 2.3: Beispiel Dienstreiseantrag (eEPK)

### 2.3.2 Business Process Model and Notation

Business Process Model and Notation (BPMN) wurde von Stephen A. White (IBM Mitarbeiter) zur graphischen Darstellung von Geschäftsprozessen entwickelt. Es handelt sich hierbei um eine international standardisierte grafische Methode zur Prozessmodellierung. Sie wird heute von der Object Management Group (OMG) verwaltet. Die OMG ist ein non-profit Konsortium aus IT-Unternehmen wie zum Beispiel IBM, Oracle, SAP AG, Software AG oder IDS Scheer. Die OMG hat das Ziel herstellerunabhängige und international anerkannte Standards für die Entwicklung und den Betrieb von Informationssystemen, wie zum Beispiel UML, zu entwickeln. Der Hintergrund der Entwicklung von BPMN war die Vereinheitlichung unterschiedlicher Sprachen zur Geschäftsprozessmodellierung. Ein weiteres Ziel war die Unterstützung der automatisierten Transformationen von Prozessmodellen in ablauffähige Workflows. Ziel ist es, dass organisatorische und technische BPM zu verbinden und die Lücke zwischen Prozessanalyst und Prozessentwickler zu schließen.

Die aktuell Spezifikation von BPMN liegt in der Version 2.0.2<sup>3</sup> vor. Es definiert alle Symbole und Regeln, nach denen sie kombiniert werden dürfen. Sie regelt damit Syntax und Semantik. Die Syntax ist das System an Regeln, welches die Kombination der Symbole organisiert. Die Semantik legt die Bedeutung von Symbolen und ihren Beziehungen fest. In der Spezifikation werden mehrere Diagrammtypen definiert. Der am meisten verwendete Typ ist das Business Process Diagram (BPD). Neben diesem gibt es noch das Kollaborationsdiagramm und seit der Version 2.0 das Choreographie- und Konversationsdiagramm. (vgl. Allweyer 2009, S. 11) Auf diese weiteren Diagrammtypen wird nicht weiter eingegangen. Ähnlich wie bei der EPK erfolgt die Modellierung in einem Ebenenkonzept. Dabei werden jedoch nicht immer die gleichen Begrifflichkeiten verwendet (vgl. Freund & Rücker 2012, S. 15 f.).

BPMN ist XML-basiert. Die mehr als 100 Elemente lassen sich in die fünf Kategorien flow objects, data, connecting objects, swimlanes und artifacts einteilen. (vgl. Object Management Group & andere 2014, S. 25) In der Praxis wird jedoch meist nur ein Bruchteil der Elemente verwendet. Neben der Spezifikation existieren verschiedene *guidelines* und *best practis* Vorschläge<sup>4</sup>. Die Basiselemente sind in der Tabelle 2.2 dargestellt. Für genauere Erläuterungen ist die Spezifikation oder einschlägige Literatur zu verwenden.

<sup>3</sup><http://www.omg.org/spec/BPMN/2.0.2/>

<sup>4</sup>z.B. <http://www.modeling-guidelines.org/de>

Kategorie	Notation	Element
Flow objects	○ ○ ○	Event (Start, Zwischen, End)
	□ □	Aktivität, Sub-Aktivität
	◇	Gateway
Data	📄	Datenobjekt
Connecting objects	→	Sequenzfluss
	⌋ - - - - - ⌋	Nachrichtenfluss
	⋯ - - - - - ⌋	Association
Swimlanes	<div style="border: 1px solid black; padding: 2px;"> <div style="border-right: 1px solid black; padding-right: 5px;">Name</div> <div style="padding: 2px;">Name</div> </div>	Pool mit Lane
Artifacts	□	Gruppe
	<div style="border: 1px solid black; padding: 2px;"> <div style="border-right: 1px solid black; padding-right: 5px;">Name</div> <div style="padding: 2px;">Name</div> </div>	Anmerkung

Tabelle 2.2: BPMN Basiselemente

Ein Beispiel für ein BPMN Prozessmodell ist in Abbildung 2.4 dargestellt. Es handelt sich, wie bei dem eEPK Beispiel in Kapitel 2.3.1, um einen Dienstreiseantrag.

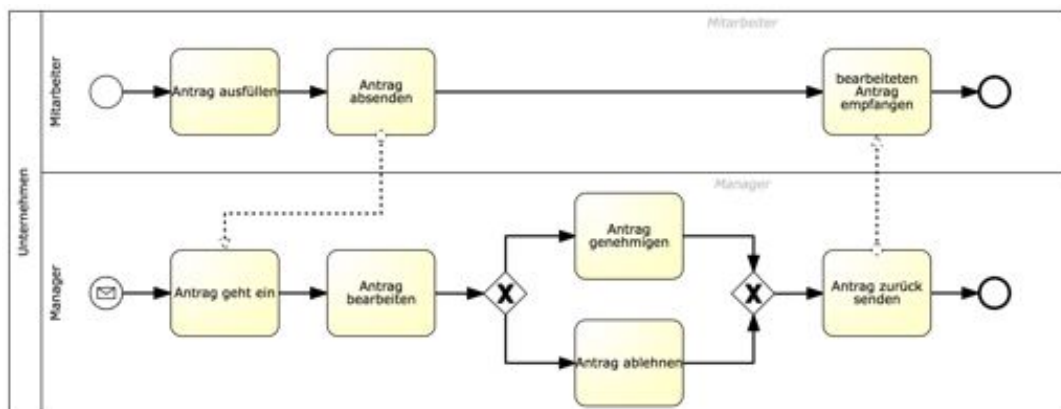


Abbildung 2.4: Beispiel Dienstreiseantrag (BPMN)

## 2.4 Business Process Simulation

Die Business Process Simulation (BPS) beschäftigt sich mit der Simulation von Geschäftsprozessen. Der Begriff Simulation stammt aus dem lateinischen und

bedeutet soviel wie „nachbilden“ oder „nachahmen“ (vgl. Dudenverlag 2013b). Eine Simulation kann als eine Ausführung verstanden werden, die in Wirklichkeit nicht umgesetzt wurde. Es ist ein quantitatives Analyseverfahren und im BPM Lebenszyklus der Analysephase zugeordnet. Sie kann jedoch auch in der Design-, Ausführungs- und Kontrollphase verwendet werden.

Jeder Prozess kostet Geld, Ressourcen und Zeit. Um dies zu optimieren, gibt es verschiedene Techniken. Drei bekannten Techniken sind das Continuous Process Improvement (CPI), Business Process Reengineering (BPR) und Business Process Re-Design (BPRD). Mit der Simulation kann herausgefunden werden wie sich bestimmte Sachverhalte, z.B. Ausfall von Mitarbeitern oder sessionalen Nachfragen, auf das Prozessmodell und den gesamten Prozess auswirken. Diese Prozessschwachstellen können identifiziert und optimiert werden. Dadurch minimiert sich ebenfalls einhergehend das Risiko. Zudem können Geschäftsprozessparameter wie KPI und SLA vorausberechnet werden. Weitere Ziele neben der Optimierung der Prozessschritte sind die Validierung der Realitätstreue der Prozessmodelle und die Überprüfung deren Konsistenz.

Der Simulationsprozess gliedert sich in die fünf Schritte: Prozessmodellierung, die Erweiterung des Prozesses mit Simulationsinformationen (Simulationsmodel), Ausführen der Simulation, Analyse der Simulationsergebnisse und das Wiederholen des Prozesses mit anderen Szenarien bzw. Inputs. (vgl. Dumas 2014, S. 20)

Für die Simulation wird ein Prozessmodell benötigt, welches die Realität abbildet. Dieses Prozessmodell muss mit zusätzlichen Parameter angereichert werden, damit die Leistungsfähigkeit des Modells ermittelt werden kann. Die hinterlegten Daten für die Simulation können durch reale Messungen gewonnen werden oder durch deren Schätzungen. Bei geschätzten Daten sind die Simulationsergebnisse mit Vorsicht zu betrachten, da die Simulationsergebnisse stark von den hinterlegten Daten abhängen. Somit ist es möglich, dass die Übereinstimmung mit der Realität nicht zwingend gegeben ist.

### 2.4.1 BPSim

Die BPSim Spezifikation ist ein Standard, welcher Parameter und den Austausch von Prozessanalysedaten für die Simulation definiert. Er wurde von der WfMC<sup>5</sup> erstellt. Die aktuelle Version 1.0 liegt seit dem 07. Februar 2013 vor. Die Spezifikation wurde entworfen, weil verschiedene Softwarehersteller die Simulationsparameter unterschiedlich definieren.

---

<sup>5</sup><http://www.wfmc.org/standards/bpswg>

In der BPMN 2.0 Spezifikation ist die Simulation nicht Bestandteil. Sie stellt jedoch das Element *extensionElements* bereit. In diesem können Erweiterungen, wie beispielsweise Simulationsparameter, definiert werden.

„In addition, operational simulation, monitoring, and deployment of Business Processes are out of scope of this International Standard.“ (Object Management Group & andere 2014, S.20)

BPSim erlaubt es, in BPMN Version 2.0 und XPD L Version 2.2 Dokumenten zusätzliche Informationen für die Simulation und Analyse zu hinterlegen. Die Spezifikation definiert (via XML) Parameter, die für die Strukturelle- und Kapazitativeanalyse von Geschäftsprozessen notwendig sind und ermöglicht somit eine Modellierungswerkzeuginteroperabilität. Die Definition der Elemente und Attribute erfolgt via XML Schema Definition (XSD)<sup>6</sup>. Zudem existiert eine Implementationsrichtlinie<sup>7</sup>. Die Abbildung 2.5 veranschaulicht das Konzeptuellmodell. (vgl. WfMC 2012, S. 4 ff.)

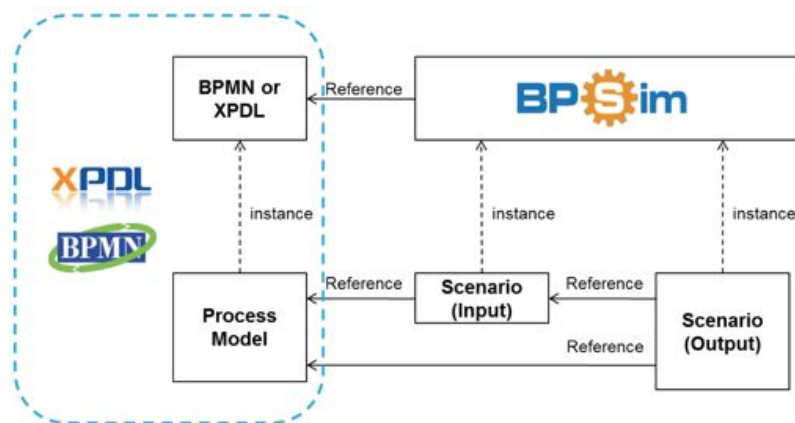


Abbildung 2.5: BPSim Scope (WfMC 2012, S.5)

Jeder definierte Simulationsparameter des Szenarios ist über die Element ID eindeutig mit einem Prozessmodellelement verknüpft. Jedes Prozessmodellelement kann wiederum kein oder mehrere Simulationsparameter besitzen. Es gibt 5 Parametergruppen. Diese Gruppen sind Zeit, Kontroll-, Resource-, Kosten-, Instanz- und Prioritätsparameter.

Der Quellcode 2.1 enthält Ausschnitte der Definition von Simulationsparametern. Das Beispiel stammt aus dem Kapitel 4.2.2.

<sup>6</sup><http://www.bpsim.org/schemas/1.0/>

7 <http://www.bpsim.org/implementers/1.0.1/BPSim%20Implementer's%20Guide%20v1.0.1.zip>

```

1  ... xmlns:bpsim="http://www.bpsim.org/schemas/1.0" ... xsi:schemaLocation="...
    http://www.bpsim.org/schemas/1.0 bpsim.xsd"
2  ...
3  <bpmn2:extensionElements>
4    <bpsim:BPSimData>
5      <bpsim:Scenario xsi:type="bpsim:Scenario" id="default" name="
        Simulationsscenario">
6        <bpsim:ScenarioParameters xsi:type="bpsim:ScenarioParameters"
          baseTimeUnit="s"/>
7        <!-- sequenzfluss reference -->
8        <bpsim:ElementParameters xsi:type="bpsim:ElementParameters" elementRef
          ="_B0...BE" id="_...">
9          <bpsim:ControlParameters xsi:type="bpsim:ControlParameters">
10           <bpsim:Probability xsi:type="bpsim:Parameter">
11             <bpsim:FloatingParameter value="30.0"/>
12           </bpsim:Probability>
13         </bpsim:ControlParameters>
14       </bpsim:ElementParameters>
15       ...
16       <!-- task reference -->
17       <bpsim:ElementParameters xsi:type="bpsim:ElementParameters" elementRef
          ="_4F3...3B" id="_...">
18         <bpsim:TimeParameters xsi:type="bpsim:TimeParameters">
19           <bpsim:ProcessingTime xsi:type="bpsim:Parameter">
20             <bpsim:NormalDistribution mean="4.0" standardDeviation="0.0"/>
21           </bpsim:ProcessingTime>
22         </bpsim:TimeParameters>
23         <bpsim:ResourceParameters xsi:type="bpsim:ResourceParameters">
24           <bpsim:Availability xsi:type="bpsim:Parameter">
25             <bpsim:FloatingParameter value="8.0"/>
26           </bpsim:Availability>
27           <bpsim:Quantity xsi:type="bpsim:Parameter">
28             <bpsim:FloatingParameter value="0.0"/>
29           </bpsim:Quantity>
30         </bpsim:ResourceParameters>
31         <bpsim:CostParameters xsi:type="bpsim:CostParameters">
32           <bpsim:UnitCost xsi:type="bpsim:Parameter">
33             <bpsim:FloatingParameter value="10.0"/>
34           </bpsim:UnitCost>
35         </bpsim:CostParameters>
36       </bpsim:ElementParameters>
37     ...

```

Quellcode 2.1: Evaluation.v1.bpmn2

## 2.4.2 EPC-Simulator

Der Event-driven Process Chain (EPC)-Simulator wird seit 2011 von Prof. Dr. Christian Müller und Studierenden der Technische Hochschule Wildau [FH] entwickelt. Es ist „... ein[es] modellgetriebene[n]s Simulationssystem[s], mit dessen Hilfe es möglich ist, aus modellierten Ereignisgesteuerten Prozessketten (EPK bzw. EPC) ein ausführbares Simulationsmodell zu generieren.“ (Müller 2013, S.1).

Die Architektur dieses Simulationssystems basiert auf den open Source EPK

Modellierungswerkzeug bflow<sup>8</sup> und dem freien Discrete-Event Simulation Modelling in Java (DESMO-J)<sup>9</sup> Simulationsframework. Bflow wurde 2007 von Frank Rump und Markus Nüttgens initiiert und an der Fachhochschule Oldenburg, Ostfriesland, Wilhelmshaven und der Universität in Hamburg entwickelt. Das ereignisorientierte, diskrete Simulationsframework DESMO-J wird seit 1999 ebenfalls von der Universität Hamburg entwickelt.

Der EPC-Simulator arbeitet mit zwei Dateien. Die Modell-Datei (modell.epc) beschreibt die Rahmenbedingungen des Simulationsmodells. Die Prozess-Datei (prozess.epc) beschreibt den Prozess und enthält die simulationsspezifischen Attribute. Die allgemein verwendeten Attribute sind die TimeUnit, Distributionen, Prozesspfade, Id-, AddOn-, Statistik- und Time Preference Attribute. Nachdem das Modell und die Parameter definiert sind, wird es validiert und ein Java basiertes Simulationsmodell generiert. Dieses Simulationsmodell kann via DESMO-J und Java Parallelize Processing Framework (JPPF)<sup>10</sup> verteilt ausgeführt werden kann. Die Abbildung 2.6 veranschaulicht den Gesamtaufbau. (vgl. Müller 2013, S. 17 ff.)

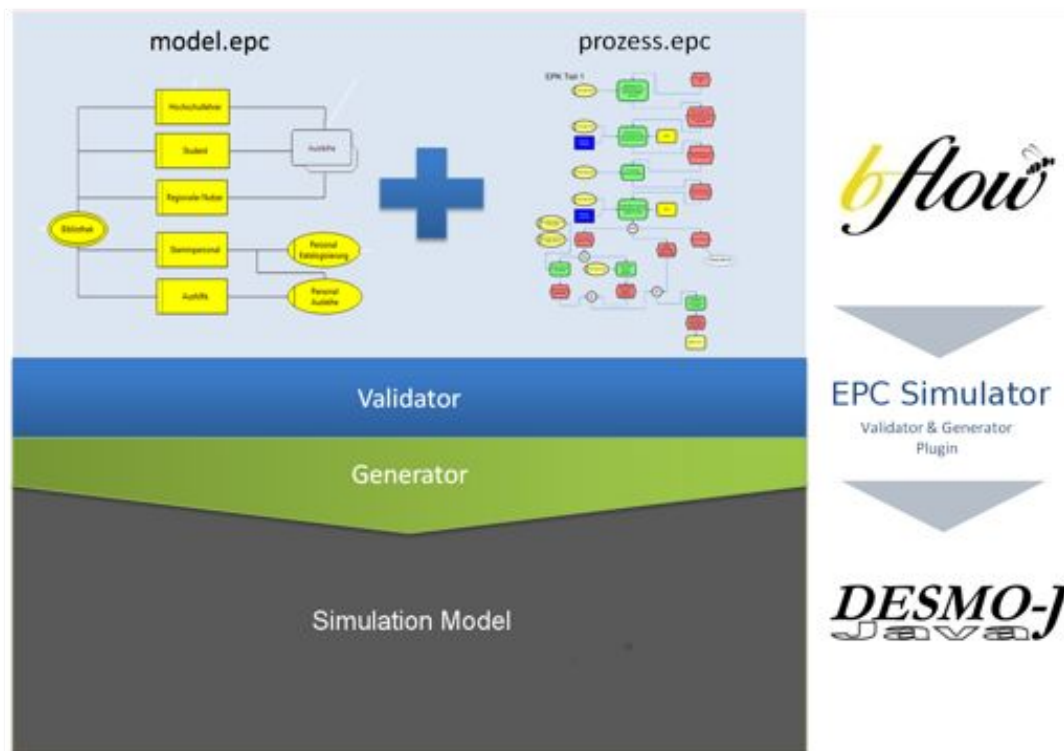


Abbildung 2.6: EPC-Simulator Architektur und Logo (Müller 2013, S. 1)

<sup>8</sup><http://www.bflow.org/>

<sup>9</sup>Discrete-Event Simulation Modelling in Java, <http://desmoj.sourceforge.net/home.html>

<sup>10</sup><http://jppf.org/>





## 3 Auswahl der Modellierungswerkzeuge

In diesem Kapitel wird beschrieben, welche BPMN Modellierungswerkzeuge bei der Internetrecherche gefunden wurden. Darauf folgend wird die Auswahl der BPMN Modellierungswerkzeuge erläutert.

### 3.1 Recherche

Zur Identifizierung möglicher BPMN Modellierungswerkzeuge wurde zunächst eine Internetrecherche durchgeführt. Dabei wurde die Eingrenzung durch folgende Kriterien vorgenommen:

- Modellierung von BPMN 2.0 Prozessmodellen möglich
- Hinweise auf eine freie Verfügbarkeit

Nach der Recherche wurde die Softwareliste 3.1 in alphabetischer Reihenfolge nach Name der Software zusammengestellt. Es ist festzuhalten, dass weitere Modellierungswerkzeuge existieren können. Dies sind die wichtigsten Modellierungswerkzeuge.

Name	Hersteller	Lizenz	Plattform
Activiti Designer	Alfresco Software AG, Community	Apache License v2	web-basiert
Activiti Modeler	Alfresco Software AG, Community	Apache License v2	Eclipse
ADONIS Community Edition	BOC Information Technologies Consulting AG <sup>1</sup>	Freeware, Registrierung erforderlich	Windows
ARIS Express	Software AG	Freeware, Registrierung erforderlich	Windows (inoffiziell Linux und Mac)

---

<sup>1</sup><http://www.boc-group.com/de/>

Name	Hersteller	Lizenz	Plattform
AuraPortal Helium Modeler	AuraPortal <sup>2</sup>	-, Registrierung erforderlich	-
Bizagi Process Modeler	Bizagi	Freeware	Windows
BIMP	Prof. Marlon Dumas, Signavio, Visual Paradigm	GPL v3	web-basiert, Java
Bonita Studio	Bonitasoft, Community	GPL v2	Windows, Linux, Mac
bpmn.io	Camunda Service GmbH, Zalando SE	Freier Service <sup>3</sup>	web-basiert
Camunda Modeler	Camunda	EPL v1	Eclipse
Cubetto Toolsets Community Edition	semture GmbH	-	Windows, Mac
Eclipse BPMN2 Modeler	Eclipse Community	EPL v1	Eclipse
Ebase Xi Community	Ebase Technology	-, Registrierung erforderlich	Windows, Linux, Mac
Innovator for Business Analysts - Personal Edition	MID GmbH	-, Registrierung erforderlich	Windows
Intalio bpms Community Edition	Intalio	-, Registrierung erforderlich	-
INNOVATOR for Business Analysts Personal Edition	MID GmbH	-	Windows
jBPM	Red Hat, Community	Apache License v2	Windows, Linux, Mac
Joget Workflow Community Edition	Joget Inc.	GPL	web-basiert, Java

<sup>2</sup><http://www.auraportal.com/en/auraportal-helium-modeler>

<sup>3</sup><http://bpmn.io/legal/terms.html>

Name	Hersteller	Lizenz	Plattform
iyopro Basic	intellivate GmbH	-	web-basiert, Silverlight
Modelio SYDLE SEED Community	Modeliosoft	Freeware	web-basiert, Cloud
Oryx	HPI-Potsdam	GPL v3	web-basiert
ProcessMaker Open Source	Colosa	AGPL v3	web-basiert
rigrr BPMN editor	Assessments Australia, Asse-tic	-	web-basiert
RunaWFE BPMS	Runa Consulting Group partners <sup>4</sup>	LGPL v2	Windows, Linux, Solaris, FreeBSD
Signavio Academic Initiative	Signavio	Für Lehrzwecke <sup>5</sup>	web-basiert
SIMUL8 Education	SIMUL8 Corporation	-, für Lehrzwecke	-
Software Ideas Modeler	Dušan Rodina, Software Ideas	Frei für nicht Kommerziel- lezwecke	Windows
Visual Paradigm Community Edition	Visual Paradigm International	Frei für nicht Kommerziel- lezwecke	Windows, Linux, Mac
Wapama	Antoine Toulme	MIT Lizenz	web-basiert
Yaoqiang BPMN Editor	Community, Yaoqiang Inc.	v2.2.x - GPL v3, v3.x - kommerziell	Windows, Linux, Mac
yEd	yWorks GmbH	Freeware <sup>6</sup>	Windows, Linux, Mac

- keine Angabe

Tabelle 3.1: Interneteherche nach Modellierungswerkzeuge

<sup>4</sup><http://runawfe.org/Partners>

<sup>5</sup><http://www.signavio.com/bpm-academic-initiative/>

<sup>6</sup><http://yed.yworks.com/support/qa/515/why-not-an-open-source-license>

## 3.2 Auswahl

Im folgenden Abschnitt werden die Kriterien vorgestellt, anhand derer die Auswahl der BPMN Modellierungswerkzeuge erfolgt. Die ausgewählten BPMN Modellierungswerkzeuge werden im Kapitel 4 zu diesen Punkten betrachtet.

**Grafische Modellierung vom BPMN** beschreibt die Unterstützung der Modellierung von BPMN 2.0 Diagrammen. Dabei soll eine einfache Handhabung gewährleistet sein und grundlegende Elemente zur Verfügung stehen.

**Im-/Export Funktionalität** beschreibt die Möglichkeit des Lesens und Speicherns von BPMN 2.0 Diagrammen in den Formaten .bpmn bzw. .bpmn2 sowie das Schreiben von Bildformate (z.B. .png, .jpg, .gif oder .tif).

**Verfügbare Dokumentation** beschreibt das Vorhandensein der Dokumentation. Es beinhaltet neben der normalen Dokumentation der aktuellen Version auch User Guide, Tutorials und Community Plattformen sowie Mailinglisten, Wikis und Foren.

**Projektaktivität** beschreibt die grundlegende Aktivität des Projektes und deren Releasezyklen.

**Plattformunabhängigkeit** beschreibt für welche Betriebssysteme das Modellierungswerkzeug existiert.

**Verfügbarkeit des Quellcodes** beschreibt unter welcher Lizenz die Software steht.

**Integration eigener Erweiterungen** beschreibt die Möglichkeit in der Anwendung bzw. Komponente eigene Erweiterungen zu integrieren.

**Simulationsparameter** beschreibt, ob bereits die Möglichkeit besteht, Simulationsparameter in das BPMN Prozessmodell zu hinterlegen.

Die Auswahl der Modellierungswerkzeuge erfolgt in Zusammenarbeit mit Prof. Dr. Müller und Prof. Dr. Bösing und unter Berücksichtigung der Auswahlkriterien. Dabei standen die Punkte der Verfügbarkeit des Quellcodes und damit der open source Gedanke, die Im-/Export Funktionalität und die Projektaktivität im Vordergrund. Aufgrund der bereits erfolgreichen Umsetzung des EPC-Simulators, liegt der Fokus zudem auf Eclipse basierten Modellierungswerkzeugen. Daraufhin erfolgt die Auswahl auf folgende Modellierungswerkzeuge:

- jBPM
- Activiti
- Camunda
- Alternativ: Yaoqiang und BIMP



## 4 Modellierungswerkzeuge

In diesem Kapitel werden die BPMN 2.0 Modellierungswerkzeuge, die in Kapitel 3.2 ausgewählt worden sind, genauer beschrieben. Mit ihnen ist es möglich BPMN 2.0 Modelle darzustellen und zu editieren. Weiterhin werden Erweiterungsmöglichkeiten aufgezeigt.

Die Umsetzung von möglichen Erweiterung kann in drei Varianten geschehen:

1. Mit normierten Elementen (BPMN defakto Standard, funktioniert mit allen Modellierungswerkzeugen)
2. Mit einem speziellen bzw. separaten BPMN Modellierungswerkzeug (ähnlich dem EPC-Simulators)
3. Aktuell keine Integrationsmöglichkeit einer eigenen Erweiterung sichtbar

Wie in Kapitel 2.4.1 beschrieben, unterstützt die BPMN 2.0 Spezifikation keine Simulationsparameter. BPMN 2.0 Prozessmodelle können über den tag *ExtensionElement* individuell erweitert werden. Um Simulationparameter im Model zu hinterlegen, wurde von der WfMC die Spezifikation BPSim erstellt. Diese Spezifikation wird in dieser Arbeit als Standard für das Hinterlegen von Simulationsparametern angesehen.

Die in den Kapiteln verwendeten Prozessmodelle sind von einfacher Natur. Sie stammen aus der jeweiligen Dokumentation und werden daher hier nicht detailliert betrachtet. Die Reihenfolge der Modellierungswerkzeuge erfolgt in ihrer geschichtlichen Entwicklung.

### 4.1 Voraussetzungen

Den folgenden Kapiteln liegt eine Reihe Software zugrunde, welche für den Betrieb und die Entwicklung der Modellierungswerkzeuge erforderlich ist. Die Versionsangaben beziehen sich auf die Mindestvoraussetzungen. Die benötigte Software umfasst Oracle Java<sup>1</sup> Version 1.7, Apache Ant<sup>2</sup> Version 1.7, Apache Maven<sup>3</sup> Version 3. und Eclipse Kepler<sup>4</sup> Version 4.3. Die Installation auf Ihrem

---

<sup>1</sup><http://www.oracle.com/technetwork/java/javase/downloads/index.html>

<sup>2</sup><http://ant.apache.org/bindownload.cgi>

<sup>3</sup><http://maven.apache.org/download.cgi>

<sup>4</sup><http://eclipse.org/downloads/>

System ist den jeweiligen Projekt-Webseiten zu entnehmen.

Für diese Arbeit wurden Betriebssystem Mac OS X Mavericks Version 10.9.3 und die Software Oracle Java 1.8 Update 5, Apache Ant 1.9.4, Apache Maven 3.2.1 und Eclipse Kepler 4.3.2 64 bit verwendet. Als alternativ System wurde Microsoft Windows 7 in einer Virtuellen Maschine von VirtualBox<sup>5</sup> mit Oracle Java 1.7 Update 55, Apache Ant 1.9.3, Maven 3.2.1 und Eclipse Kepler 4.3.2 64 bit verwendet.

## 4.2 jBPM

JBoss Business Process Management (jBPM) wird als erstes von den Modellierungswerkzeugen dargestellt. Es wurde von der JBoss community<sup>6</sup> und RedHat<sup>7</sup> entwickelt. Die Entwicklung der Version 1.0 geht auf das Jahr 2003<sup>8</sup> zurück. Die aktuelle Version für den Produktiveinsatz ist 6.0.1. Die nächste Version 6.1.0 befindet sich bereits in Vorbereitung (Candidate for Release (CR)<sup>9</sup>). Die rege Weiterentwicklung lässt sich auch in der Aufgabenliste<sup>10</sup> und dem Quellcode Verzeichnis<sup>11</sup> entnehmen. In der jetzigen Version steht die Software unter der Apache License Version 2<sup>12</sup>.

jBPM ist eine in Java geschriebene BPM Suite. Mit der leichtgewichtigen und erweiterbaren Workflow Engine können BPMN 2.0 Modelle durchweg modelliert, ausgeführt und überwacht werden. Die flexible Suite ist mittlerweile auch auf nicht technische Personen ausgerichtet. Durch die flexible Java Implementierung kann jBPM direkt in bestehende Anwendungen integriert werden oder als separater Web Service (WS) laufen. Durch den Service-Oriented Architecture (SOA) Gedanke können auch nicht Java Anwendungen angebunden werden. (vgl. jBPM team 2014a)

Seit der Version 6 werden Werkzeuge zum Erstellen, Deployen, Auszuführen und Managen von Geschäftsprozessen über ihren Lebenszyklus hinweg bereitgestellt. Die Abbildung 4.1 gibt eine Übersicht der verschiedenen Komponenten der jBPM Suite.

Die *Core Engine* ist für die Ausführung der Prozessmodelle verantwortlich und damit das Herzstück des Projektes. Ebenfalls zur Ausführung gehören die Kom-

---

<sup>5</sup><https://www.virtualbox.org>

<sup>6</sup><http://jbpm.jboss.org>

<sup>7</sup><http://www.redhat.com>

<sup>8</sup><http://sourceforge.net/projects/jbpm/files/Old%20Files/jBPM%201/>

<sup>9</sup><http://sourceforge.net/projects/jbpm/files/jBPM%206/jbpm-6.1.0.CR1/>

<sup>10</sup><https://jira.jboss.org/browse/JPBM>

<sup>11</sup><https://github.com/droolsjbpm/jbpm>

<sup>12</sup><https://www.apache.org/licenses/LICENSE-2.0.html>



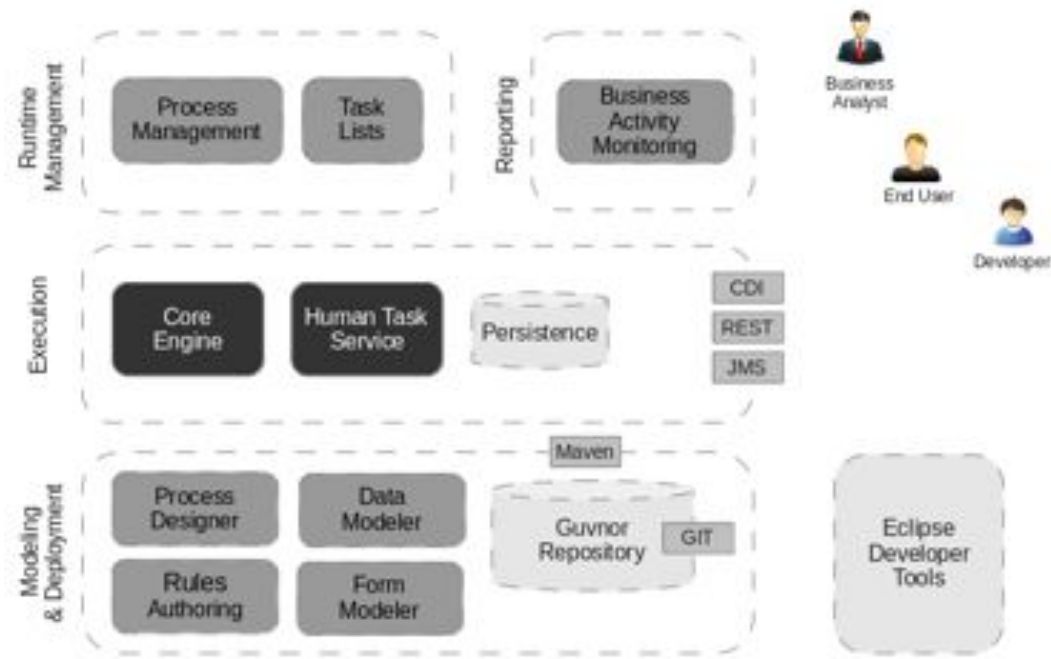


Abbildung 4.1: jBPM Übersicht (jBPM team 2014a)

ponenten *Human Task Service* und *Persistence*. Der *Human Task Service* ist für die Verwaltung der Benutzeraufgaben verantwortlich. Die *Persistence* Komponente speichert alle Informationen über vergangene und aktuell ausgeführte Prozesse. Die *jBPM console* ist eine vollständig neugestaltete web-basierte Management Konsole. Diese kann neue Prozesse starten, deren Status überwachen, beinhaltet eine Aufgabenliste und bietet viele weitere Funktionen. Der web-basierte *Process Designer* ist für diese Arbeit eine wichtige Komponente. Die Komponente richtet sich an Prozessanalysten. Mit ihr können Prozesse betrachtet, bearbeitet und simuliert werden. Ähnlich den *Eclipse Developer Tools*, welche jedoch nicht die Simulation von BPMN Prozessmodellen erlauben. Diese Eclipse Plugins richten sich an Entwickler. Auf die anderen Komponenten wird hier nicht weiter eingegangen. (vgl. jBPM team 2014a)

In den folgenden Kapiteln wird der *Eclipse Developer Tools* und der *Process Designer* untersucht.

### 4.2.1 Eclipse Developer Tools

Die Eclipse Developer Tools (deutsch Eclipse Entwickler Werkzeuge) sind Eclipse Plugins für technische Benutzer. Damit wird die Entwicklungsumgebung um die Funktion erweitert, Prozesse zu editieren, zu testen sowie jBPM in die eigene Anwendung zu integrieren. Die Werkzeuge umfassen folgende Funktionen.

- Wizards für das Anlegen neuer jBPM Projekte

- Grafischer Editor für BPMN 2.0 Prozessmodelle
- Möglichkeit eigene Domain spezifische Elemente hinzuzufügen
- jBPM Laufzeitunterstützung (z.B. jBPM 5, jBPM 6 usw.)
- Grafische Fehlersuche bei laufenden Prozessen
- Validierung von BPMN 2 Modellen
- Hinterlegung von Simulationsparametern  
(vgl. jBPM team 2014a)

Als Modellierungswerkzeug wird seit Version 6 der *Eclipse BPMN 2 Modeler*<sup>13</sup> verwendet. Der Eclipse BPMN 2 Modeler steht unter der Eclipse Foundation. Die JBoss community und RedHat unterstützt weiterhin die Entwicklung. Die aktuelle Version für Eclipse Kepler ist Version 1.0.2. In der kommenden Eclipse Version Luna wird das Plugin in der Version 1.1.0 enthalten sein. Der Modeler selbst steht unter der Eclipse Public License (EPL)<sup>14</sup> Version 1.0.

#### 4.2.1.1 Eclipse Developer Tools Installation

Die Installation des Eclipse Modelers kann über zwei Varianten erfolgen. Die erste Variante ist über den jBPM installer<sup>15</sup>. Die zweite Variante ist direkt über Eclipse. Die erste Variante wird in Kapitel 4.2.2 beschrieben.

Die zweite Variante bezieht sich ausschließlich auf Eclipse Entwicklungsumgebung. Dafür muss Eclipse gestartet und das Menü *Help / Install New Software...* ausgewählt werden.

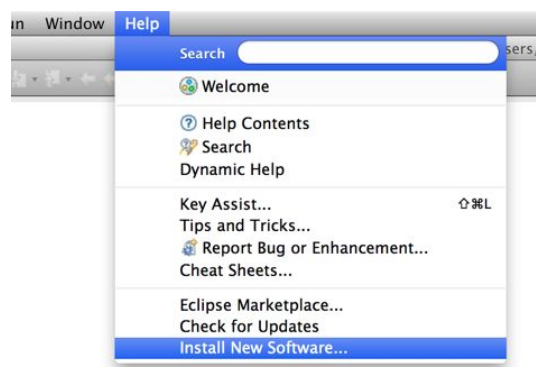


Abbildung 4.2: jBPM Eclipse Plugins installieren

Nun können, über den *Add...* Button die Repositories „jBPM“ mit „<http://downloads.jboss.org/jbpm/release/6.0.1.Final/updatesite/>“ und „BPMN2 Modeler“ mit „<http://download.eclipse.org/bpmn2-modeler/updates/kepler/1.0.2>“ hinzugefügt werden. Ein Repository kann mehrere Eclipse Plugins zur

<sup>13</sup><http://projects.eclipse.org/projects/soa.bpmn2-modeler>

<sup>14</sup><http://www.eclipse.org/legal/epl-v10.html>

<sup>15</sup><http://docs.jboss.org/jbpm/v6.0.1/userguide/jBPMInstaller.html>

Installation bereitstellen. Aus den Repository „jBPM“ muss das Plugin *JBoss Drools Core* und *JBoss jBPM Core* installiert werden. Der folgende Wizard-Schritte überprüfen Plugin-Abhängigkeiten. Der letzte Wizard-Schritt zeigt die Lizenzen der Plugins an und muss für die Installation bestätigt werden. Im Anschluss muss Eclipse neu gestartet werden. Analog dazu wird der „BPMN2 Modeler“ mit allen Plugins installiert. (vgl. jBPM team 2014c)

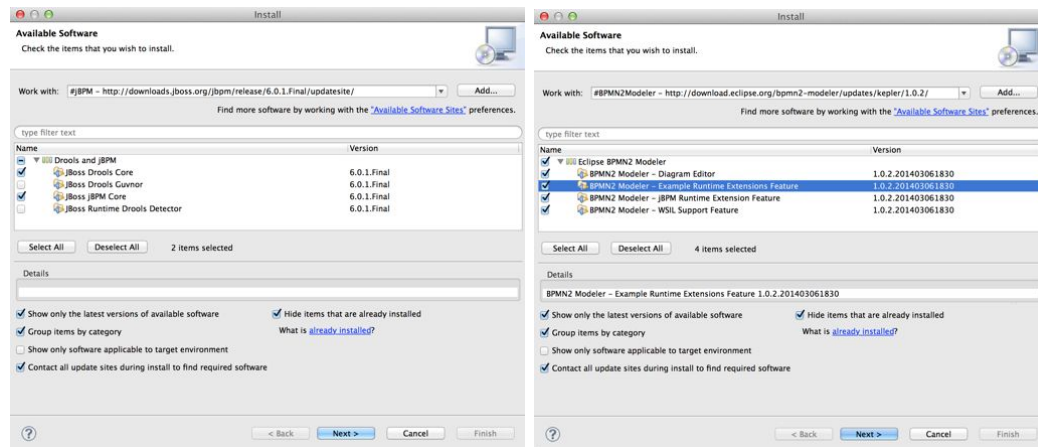


Abbildung 4.3: jBPM Eclipse Plugin Auswahl

### 4.2.1.2 Eclipse Developer Tools Beispiel

In diesem Kapitel wird beschrieben wie ein jBPM Projekt angelegt wird, die grafische Modellierungsoberfläche aussieht und BPMN Prozessmodelle mit Simulationsparametern hinterlegt werden können.

Bevor das Beispielprojekt angelegt wird, sind zwei Einstellungen vorzunehmen. Die erste Einstellung definiert die jBPM Runtime (deutsch Laufzeitumgebung) und die zweite schaltet die Simulationsparameter frei. Hierfür wird zu den Eclipse Einstellungen navigiert. Diese sind über das Menü *Eclipse / Preferences...* erreichbar. Im linken Navigationsbaum befinden sich die Einträge *BPMN2* und *jBPM*. Unter *jBPM / Installed jBPM Runtimes* wird über den *Add...* und *Create a new jBPM Runtime...* Button eine neue Laufzeitumgebung angelegt. Die Abbildung 4.4 verdeutlicht dies. Auf der beigelegten CD befindet sich diese bereits unter *sourcecode/jbpm-runtime* und kann verlinkt werden. Unter dem Navigationszweig *BPMN2 / jBPM Runtime* werden die Simulationsparameter freigeschaltet. Dies zeigt die Abbildung 4.4. Im Anschluss werden die Einstellungen mit dem *OK* Button bestätigt. Ein Neustart von Eclipse ist anschließend notwendig.

Ein neues jBPM Projekt kann über das Menü *File / New...* angelegt werden. Um Bibliotheksabhängigkeiten einfacher zu verwalten, wird das *jBPM Maven*

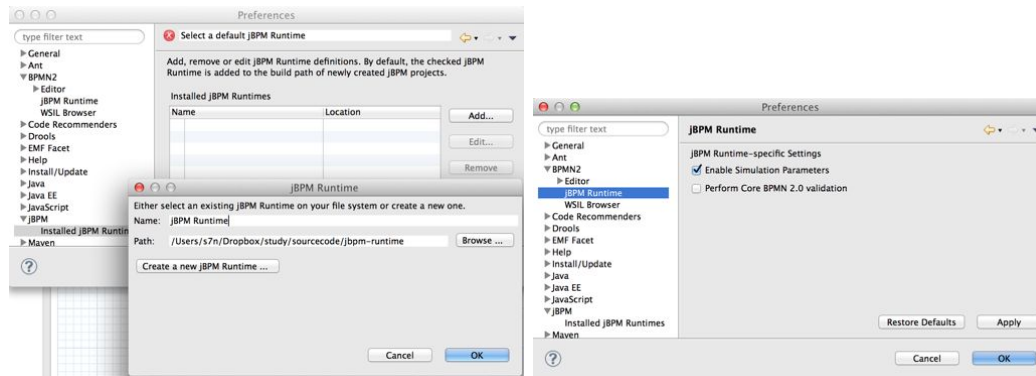


Abbildung 4.4: jBPM Eclipse Einstellungen

*Projekt* ausgewählt und auf *Next* geklickt. Dies zeigt Abbildung 4.5. Im nächsten Fenster muss der Projektname „de.th.wildau.demo.jbpm“ angegeben werden.

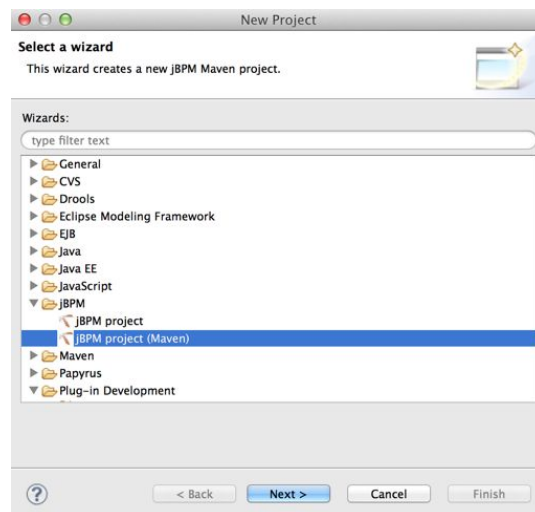


Abbildung 4.5: jBPM neues Projekt anlegen

Es wurde automatisch eine Beispielanwendung mit einem BPMN 2.0 Prozessmodell und einer ausführbaren Java Klasse generiert. Das Prozessmodell ist einfach gehalten und beinhaltet zwischen Start- und End-Event zwei UserTasks (deutsch Benutzeraufgaben). Die Abbildung 4.6 zeigt die Modelleroberfläche und das generierte Prozessmodell.

Der Quellcode 4.1 zeigt das Starten der jBPM Runtime (deutsch Laufzeitumgebung). Zur jBPM Runtime wird eine notwendige locale H2 Datenbank gestartet. Weiterhin ist in diesem Quellcodeausschnitt zu sehen, dass *john* den Task 1 abarbeitet.

```

26 ...
27 RuntimeManager manager = createRuntimeManager(kbase);
28 RuntimeEngine engine = manager.getRuntimeEngine(null);
29 TaskService taskService = engine.getTaskService();

```

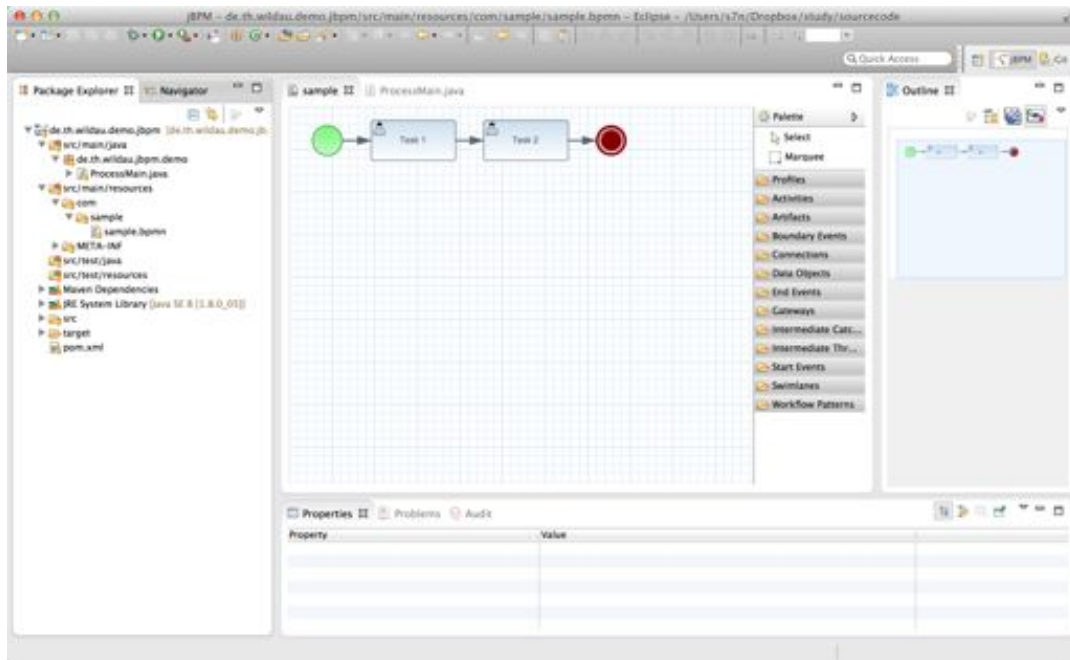


Abbildung 4.6: jBPM Perspektive

```

30 ...
31 /* let john execute Task 1 */
32 List<TaskSummary> list = taskService.getTasksAssignedAsPotentialOwner("john",
    "en-UK");
33 TaskSummary task = list.get(0);
34 System.out.println("John is executing task " + task.getName());
35 taskService.start(task.getId(), "john");
36 taskService.complete(task.getId(), "john", null);
37 ...

```

Quellcode 4.1: ProcessMain.java

Der Eclipse BPMN2 Modeler unterstützt die drei Diagrammtypen Prozess-, Choreografie- und Kollaborations-Diagramm. Siehe Abbildung 4.7.

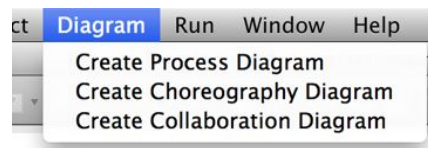


Abbildung 4.7: jBPM / Eclipse BPMN2 Modeler Diagrammtypen

Die grafische Benutzeroberfläche bietet die gebräuchlichsten BPMN 2 Elemente zur Auswahl. Die Abbildung 4.8 zeigt einige aufgeklappte Gruppen der Palette. In der Eclipse BPMN2 Modeler Version 1.1.0 existiert eine aufgeräumtere Palette mit mehr Profilen und BPMN Elementen wie beispielsweise dem Pool bzw. Participant.

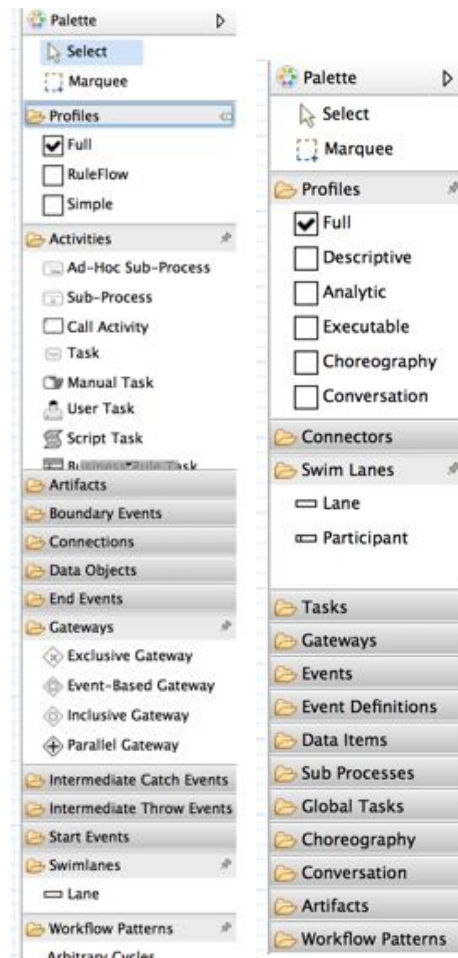


Abbildung 4.8: jBPM / Eclipse BPMN2 Modeler Palette (links v1.0.2, rechts v1.1.0)

Zudem kann das Prozessmodell über das Kontextmenü validiert und in eine Grafik exportiert werden. Siehe Abbildung 4.9.

Durch die Aktivierung der Simulationsparameter in den Einstellungen befindet sich in der *Properties*-Ansicht die Gruppe *Simulation*. Die Abbildung 4.10 zeigt die Gruppe *Simulation* des *Task 1*. Aufgrund des neuen Namespaces <http://www.bpsim.org/schemas/1.0> können jedem Element, entsprechend der BPSim Spezifikation, die Simulationsparameter zugeordnet werden.

Jedoch existiert ein Fehler, der das Bearbeiten der Simulationparameter über die grafische Benutzeroberfläche unmöglich macht. Der Fehler stellt sich wie folgt dar. Ist ein BPMN Element ausgewählt, ist die Eingabe der Simulationsparameter in das jeweilige Eingabefeld nicht möglich. Der Mauszeiger wechselt in der Gruppe *Simulation* fortlaufend zwischen Auswahl und Eingabe und die Oberfläche zeichnet sich fortlaufend neu. Durch das Neuzeichnen der Oberfläche verschwindet die Eingabe. Zudem steht bei jedem Element in der Titelleiste *... Task{0} ...*. Der Fehler existiert unter der Mac OS und Microsoft Windows



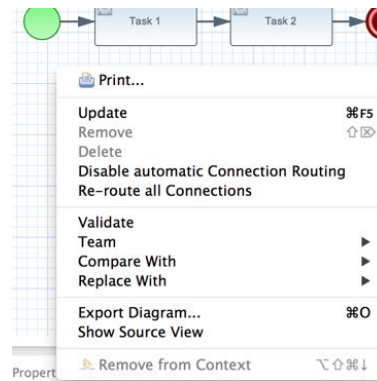


Abbildung 4.9: jBPM / Eclipse BPMN2 Modeler Kontextmenü



Abbildung 4.10: jBPM / Eclipse BPMN2 Modeler Simulationsparameter

Umgebung sowie in der Eclipse BPMN2 Modeler Version 1.0.2 und 1.1.0. Ein Bugeintrag dazu konnte leider nicht gefunden werden. Eine Alternative Modellierungsvariante mit dem *Process Designer* ist in Kapitel 4.2.2. beschrieben.

Das Öffnen von BPMN 2.0 Prozessmodellen, die nicht mit dem Editor erstellt worden sind, funktioniert. Getestet wurde dies mit der Datei *ch7CreditAppSimulation.bpmn*. Beim Öffnen erscheinen jedoch zwei Fehlermeldungen. Diese weisen auf die Zeile 291 hin, in dem die Simulationsparameter definiert wurden. Diese Fehlermeldungen und die im Anschluss geöffnete Datei sind in den Abbildungen 4.11 und 4.12 zu sehen.

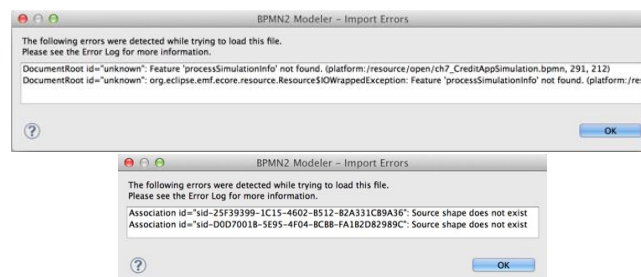


Abbildung 4.11: jBPM / Eclipse BPMN2 Modeler Fehlermeldung beim Öffnen der externen Datei

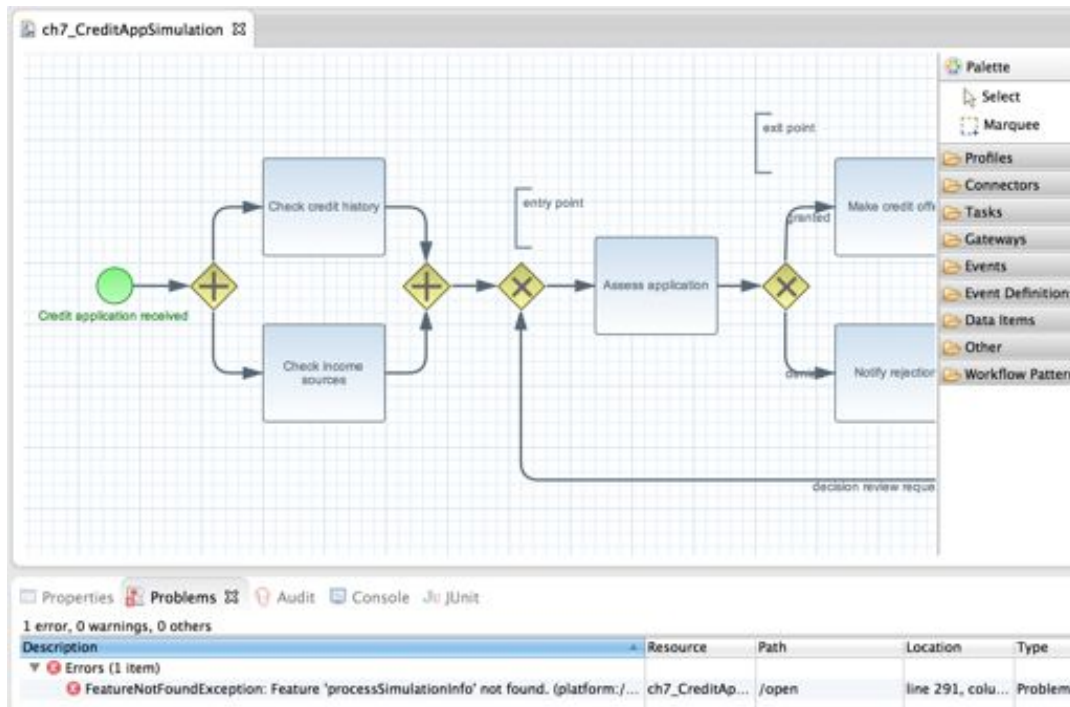


Abbildung 4.12: jBPM / Eclipse BPMN2 Modeler Anzeige externer Datei

#### 4.2.1.3 Eclipse Developer Tools Erweiterungen

Sowohl jBPM als auch der Eclipse BPMN2 Modeler, sind open source Projekte und bieten Schnittstellen für Erweiterungen an. Daraus lassen sich grundsätzlich zwei Ansatzpunkte ableiten. Zum einen ist das die direkte Unterstützung und Weiterentwicklung des Projektes und zum zweiten ist das die Nutzung der Schnittstellen. Hier wird ausschließlich die Nutzung der Schnittstellen weiter betrachtet. Folgende Erweiterungen sind vorstellbar.

1. Das Auslesen des BPMN Prozessmodelles
2. Die Validierung der BPSim Attribute zur Simulation
3. Die Ausführung einer Simulation mit einer beliebigen Software

Folgende prototypische Implementierungen für jBPM bzw. Eclipse BPMN2 Modeler wurden umgesetzt.



Projektname	Beschreibung
de.th.wildau.demo.eclipse.extension	Eigenes Eclipse Plugin zum Ausführen von eigenem Quellcode über die Eclipse Oberfläche
de.th.wildau.demo.bpmn2.modelreader	Auslesen des BPMN 2 Modelles
de.th.wildau.demo.jbpm.simulation	Simulation eines Modelles mit jBPM

Tabelle 4.1: Prototypische Eclipse BPMN2 Modeler und jBPM Projekte

Diese drei Projekte befinden sich auf der CD in dem Verzeichnis *sourcecode* und sind unter <https://github.com/search> zu finden.

Das Eclipse Plugin Projekt *de.th.wildau.demo.eclipse.extension* stellt in der Menüzeile den Eintrag *TH-Wildau*, mit der Aktion *Demo Menubar Action* und in der Toolbar Icon zur Verfügung. Beide Aktionen lösen ein Beispiel-Code aus. Es öffnet sich ein Dialogfenster. Die Abbildung 4.13 zeigt dies.

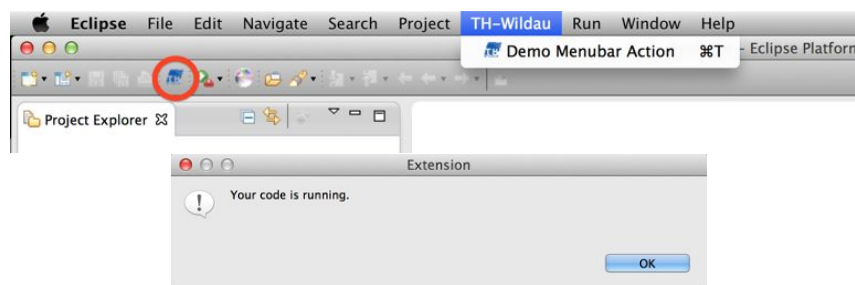


Abbildung 4.13: jBPM / Eclipse BPMN2 Modeler Extension Aktionsaufruf

Ebenfalls wurde eine *TH-Wildau* Einstellungsseite hinzugefügt. In dieser können projektspezifische Einstellungen editiert und hinterlegt werden. Siehe Abbildung 4.14.

Das Projekt *de.th.wildau.demo.bpmn2.modelreader* liest BPMN 2 Prozessmodelle aus, die im Dateisystem hinterlegt sind. Weiterhin gibt es den Namen und die ID der Elemente aus. Die Ausgabe des im Projekt hinterlegten Modelles sieht wie folgt aus. Siehe Quellcode 4.2.

```

1  2014-06-23 18:16:51 [main] de.th.wildau.demo.bpmn2.modeler.ModelReader.
    printProcess()
2  INFO: Name=Start Event 1 ID=StartEvent_1
3  2014-06-23 18:16:51 [main] de.th.wildau.demo.bpmn2.modeler.ModelReader.
    printProcess()
4  INFO: Name=User Task 1 ID=UserTask_1
5  2014-06-23 18:16:51 [main] de.th.wildau.demo.bpmn2.modeler.ModelReader.
    printProcess()

```

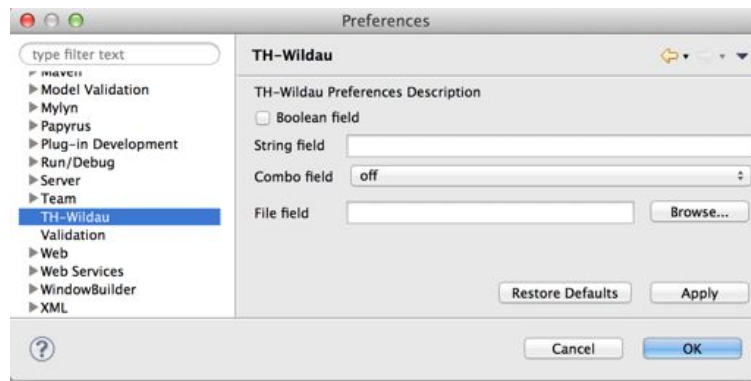


Abbildung 4.14: jBPM / Eclipse BPMN2 Modeler Extension Einstellungen

```

6  INFO: Sequence Flow: StartEvent_1 -> UserTask_1
7  2014-06-23 18:16:51 [main] de.th.wildau.demo.bpmn2.modeler.ModelReader.
    printProcess()
8  INFO: Name=User Task 2 ID=UserTask_2
9  2014-06-23 18:16:51 [main] de.th.wildau.demo.bpmn2.modeler.ModelReader.
    printProcess()
10 INFO: Sequence Flow: UserTask_1 -> UserTask_2
11 2014-06-23 18:16:51 [main] de.th.wildau.demo.bpmn2.modeler.ModelReader.
    printProcess()
12 INFO: Name=End Event 1 ID=EndEvent_1
13 2014-06-23 18:16:51 [main] de.th.wildau.demo.bpmn2.modeler.ModelReader.
    printProcess()
14 INFO: Sequence Flow: UserTask_2 -> EndEvent_1

```

#### Quellcode 4.2: Ausgabe der Sample.bpmn2 Elemente

Das Projekt *de.th.wildau.demo.jbpm.simulation* basiert auf dem Quellcode-Projekt <https://github.com/droolsjbpm/droolsjbpm-integration/jbpm-simulation>. Die Klasse *SimulationTest* wird mittels JUnit ausgeführt. In der Methode *testSimulationRun* wird das Prozessmodell *BPMN2-UserTask.bpmn2* simuliert. Die Simulationsparameter sind jedoch in der Klasse *HardCodedSimulationDataProvider* und nicht im Prozessmodell hinterlegt. Die Ausgabe der oben genannten Testmethode ist in Abbildung 4.3 zu sehen.

```

1 [main] INFO de.th.wildau.demo.jbpm.simulation.SimulationTest - start
    testSimulationRun
2 [main] INFO org.drools.compiler.kie.builder.impl.KieRepositoryImpl - KieModule
    was added:MemoryKieModule[ ReleaseId=org.default:artifact:1.0.0-SNAPSHOT]
3 SIMULATION-->Duration for human activity Place request is avg: 1447965.0; min:
    1447965.0; max:1447965.0 is avgwt: 0.0; minwt: 0.0; maxwt:0.0 is avgru:
    0.0; minru: 0.0; maxru:0.0 count 1 avg cost 24.0 min cost 24.0 max cost
    24.0
4 SIMULATION-->Duration for activity Auto evaluation is avg: 91264.0; min:
    91264.0; max:91264.0 count 1
5 SIMULATION-->Duration for activity Check applicant creditability is avg:
    190685.0; min: 190685.0; max:190685.0 count 1
6 SIMULATION-->Duration for human activity Prepare contract is avg: 3270657.0;
    min: 3270657.0; max:3270657.0 is avgwt: 0.0; minwt: 0.0; maxwt:0.0 is avgru
    : 7.0; minru: 7.0; maxru:7.0 count 1 avg cost 108.0 min cost 108.0 max cost
    108.0

```

```

7 SIMULATION-->Duration for activity Accepted notification is avg: 70540.0; min:
  70540.0; max:70540.0 count 1
8 SIMULATION-->Duration for end event is avg: 5071111.0; min: 5071111.0; max
  :5071111.0
9 SIMULATION-->Duration for processs is avg: 5071111.0; min: 5071111.0; max
  :5071111.0 [defaultPackage.banl-loan@Bank loan process@null]
10 ...
11 [main] INFO de.th.wildau.demo.jbpm.simulation.SimulationTest - done

```

Quellcode 4.3: Ausschnitte der Ausgabe der jBPM Simulation

## 4.2.2 Process Designer

Der web-basierte Process Designer erlaubt es über den Webbrowser BPMN Modelle zu editieren. Er ist im Gegensatz zu den Eclipse Developer Tools an betriebswirtschaftlich Benutzer gerichtet. Er ermöglicht ebenso die Modellierung via Drag and Drop, das Zusammenspiel mit der Eclipse IDE und stellt darüber hinaus das Simulieren von BPMN 2.0 Prozessmodellen bereit.

Der Process Designer ist in der Suite oder als alleinstehende<sup>16</sup> Webanwendung verfügbar. Er basiert auf der KIE Workbench<sup>17</sup>. Folgend wird die jBPM Suite betrachtet und auf die wesentlichsten Punkte der Simulation eingegangen.

### 4.2.2.1 Process Designer Installation

Die jBPM installer Variante liefert eine vollständige Plattform inklusiver Beispiele. Diese Umgebung beinhaltet eine H2 Datenbank, den JBoss Application Server (JBoss AS) und Eclipse. Für die Installation muss die Datei *jbpm-6.0.1.Final-installer-full.zip*<sup>18</sup> heruntergeladen, entpackt und via Terminal<sup>19</sup> in das Verzeichnis navigiert werden. Dort werden die Kommandos des Quellcodes 4.4 ausgeführt.

```

1 ant install.demo # installiert die Beispielumgebung
2 #ant install.demo.eclipse # nur Eclipse Installation
3 # ...
4 ant start.demo.noecclipse
5 #ant start.demo # startet die Beispielumgebung inkl. Eclipse
6 #ant start.demo.eclipse + startet nur Eclipse
7 # ...
8 #ant stop.demo # stoppt die Beispielumgebung

```

Quellcode 4.4: jBPM installer Kommandos

<sup>16</sup><https://github.com/droolsjbpm/jbpm-designer>

<sup>17</sup><http://docs.jboss.org/jbpm/v6.0.1/userguide/wb.Workbench.html>

<sup>18</sup><http://sourceforge.net/projects/jbpm/files/jBPM%206/>

<sup>19</sup>Kommandozeilen Programm

Aufgrund von Inkompatibilität des JBoss AS Version 7.1.1 mit der Oracle Java VM Version 1.8 muss die Alternative Microsoft Windows 7 Umgebung benutzt werden. Die Abbildung 4.15 zeigt das Starten der Anwendung über die PowerShell.

```
PS C:\Users\win\Desktop\jbpm-installer-6.0.1> ant start.demo
Buildfile: C:\Users\win\Desktop\jbpm-installer-6.0.1\build.xml

start.h2.check:
[echo] Checking if h2 db should be started ...

start.h2:
start.jboss:
start.eclipse:
start.demo:

BUILD SUCCESSFUL
Total time: 14 seconds
PS C:\Users\win\Desktop\jbpm-installer-6.0.1> _
```

Abbildung 4.15: jBPM installer Start

#### 4.2.2.2 Process Designer Simulationsbeispiel

Ist die jBPM Suite gestartet, erfolgt der Zugriff über den Webbrowser und der URL `http://localhost:8080/jbpm-console`. Die Anmeldeseite ist in Abbildung 4.16 zu sehen.

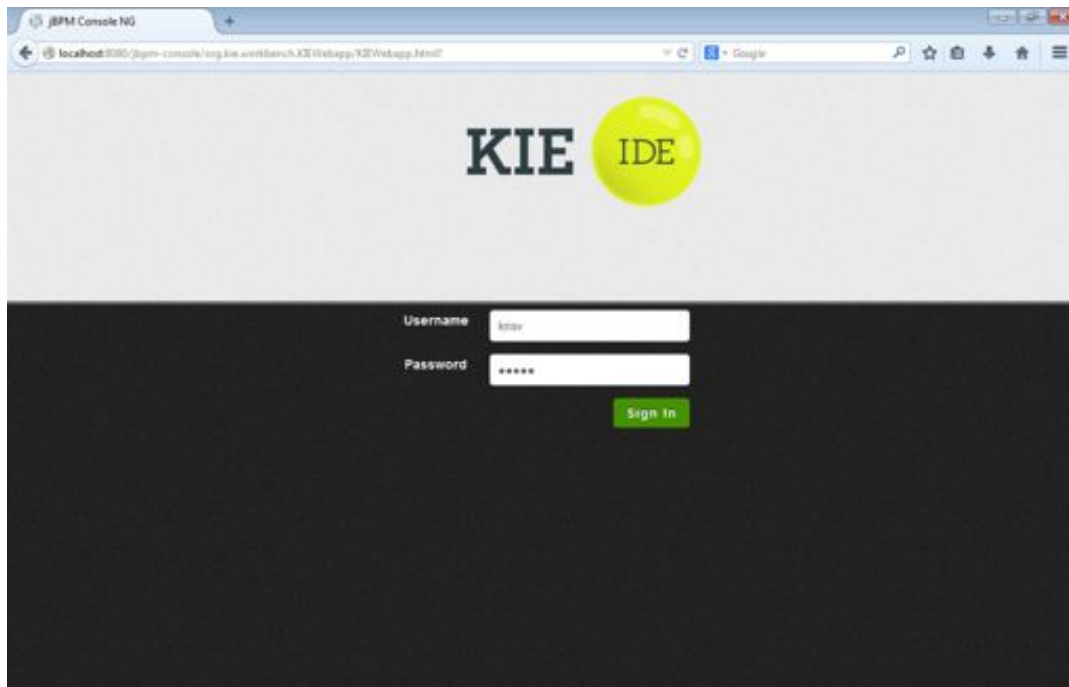


Abbildung 4.16: jBPM Process Designer Anmeldung

Die Standardanmeldung erfolgt mit dem Benutzernamen und Passwort „krisv“. Nach erfolgreicher Anmeldung ist die Übersichtsseite zu sehen. Von dort wird

über das Menü *Authoring / Administration* in die *Repository*-Übersicht navigiert. In dieser befinden sich die mitgelieferten Beispiele, inklusive der BPMN Prozessmodelle. Es wird das mitgelieferte Repository *jbpm-playground* und der darin enthaltene *Evaluation* Prozess verwendet. Durch das Klicken den *File Explorer* wird über *Repositories / jbpm-playground / Evaluation / src / main / resources* zu dem *evaluation.bpmn2* Prozessmodell navigiert. Das Prozessmodell öffnet sich daraufhin in der rechten Bildschirmhälfte. Die Abbildung 4.17 verdeutlicht dies. Das verwendete Prozessmodell *Evaluation.v1.bpmn2* befindet sich zudem auf der CD.

Wird ein BPMN Element selektiert, erscheint auf der rechten Bildschirmseite die *Properties* Ansicht. In dieser können sämtliche Einstellungen, die das Element betreffen, vorgenommen werden. Die Simulationsparameter befinden sich in der letzten Gruppierung.

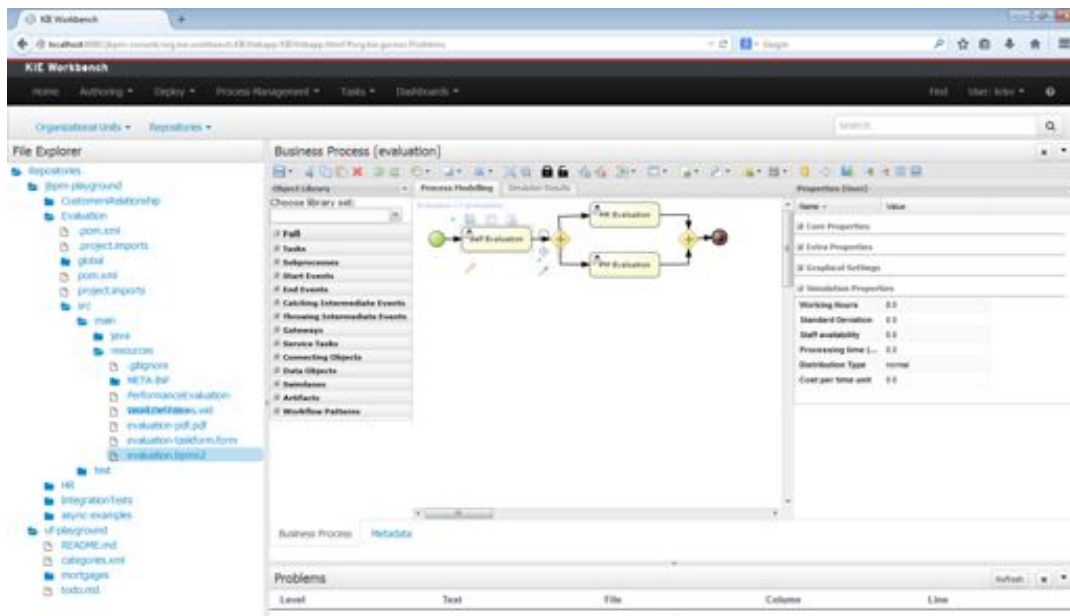


Abbildung 4.17: jBPM Process Designer Prozessmodell

Neben dem Editieren von Modellen bietet die *Toolbar* einige weitere Funktionen. Diese Funktionen sind u.a. der Im- und Export, die Validierung und die Simulation des Prozessmodelles.

Importiert werden können BPMN 2.0 und JavaScript Object Notation (JSON) Dateien. Der Export des Modelles kann in den Bildformaten PNG und SVG, den Dokumentenformaten PDF und eRDF sowie BPMN2 und JSON erfolgen.

Neben der Validierung des Prozessmodelles kann dieses simuliert werden. Dazu muss in der *Toolbar* die Funktion *Run Simulation* aufgerufen werden. Dies ist in Abbildung 4.19 zu sehen.

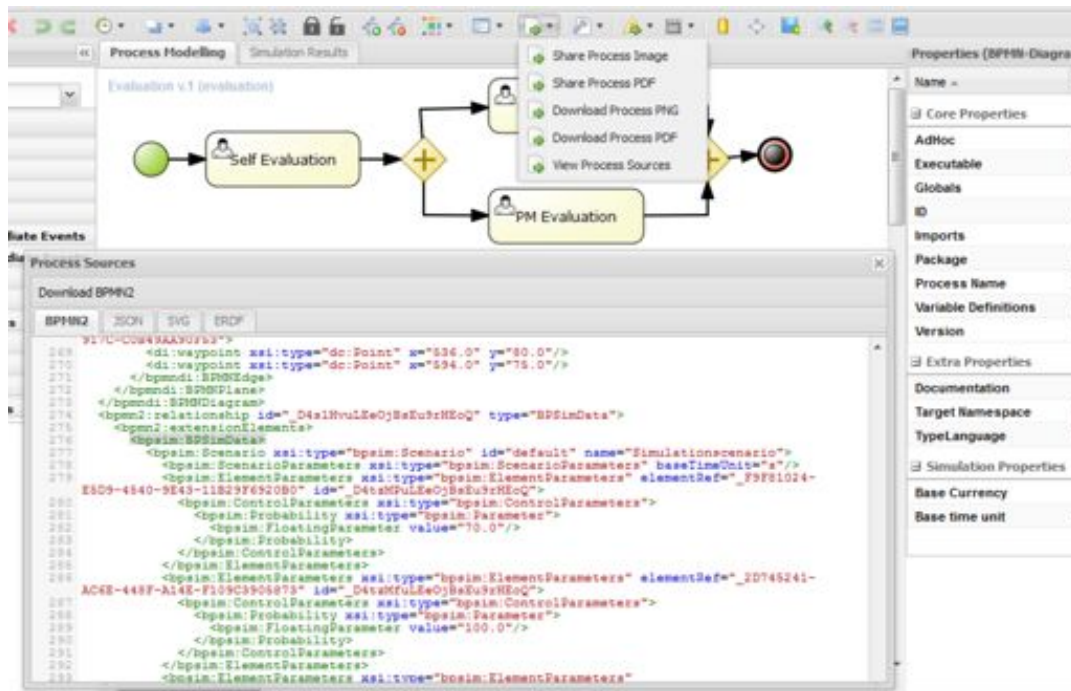


Abbildung 4.18: jBPM Process Designer Export

Bevor die Simulation gestartet werden kann, müssen die Laufzeitparameter *Number of instances*, *Interval* und *Interval unit* definiert werden. Der Parameter *Number of instances* definiert wie viele Instanzeinheiten diesen Prozess ausführen. Der Parameter *Interval* und *Interval unit* definieren in welchem Zeitintervall des Prozessmodell angestoßen wird. Sind diese Parameter definiert, so kann man die Simulation mit dem Button *Run Process Simulation* starten. Die Abbildung 4.20 zeigt die Eingabe der erforderlichen Parameter. (vgl. jBPM team 2014b)

Ist die Simulation erfolgreich beendet worden, werden in dem Tab *Simulation*

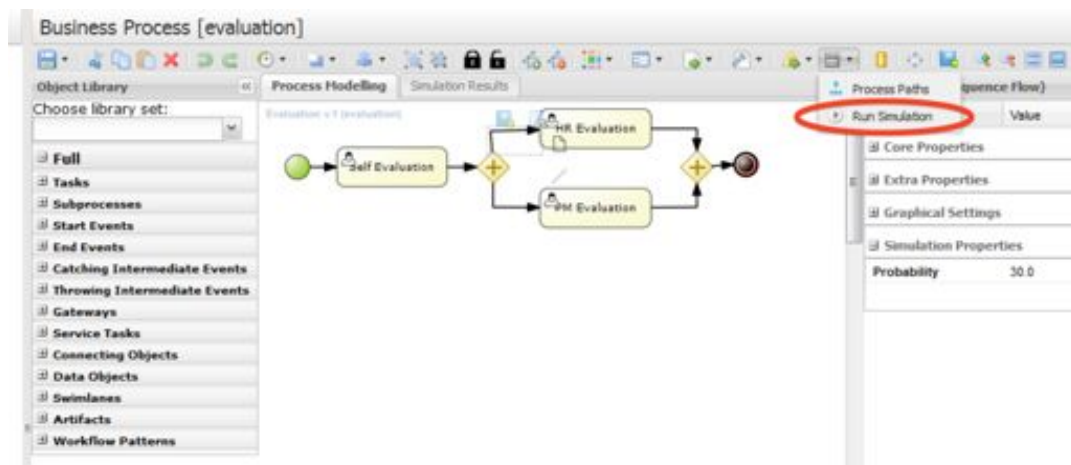


Abbildung 4.19: jBPM Process Designer Simulation starten

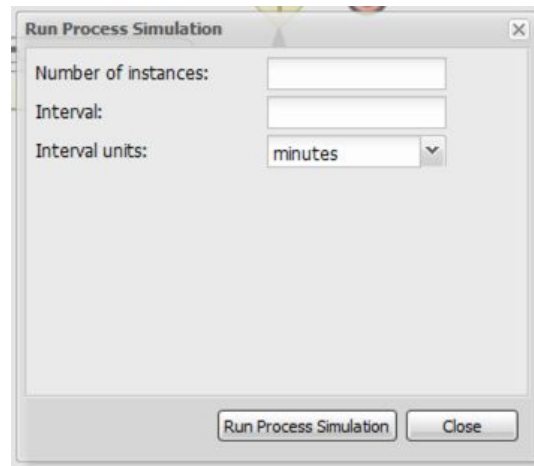


Abbildung 4.20: jBPM Process Designer Simulation - Laufzeitparameter

*Results* die Ergebnisse dynamisch dargestellt. Die Abbildung 4.21 zeigt einen Ausschnitt der Simulationsergebnisse.

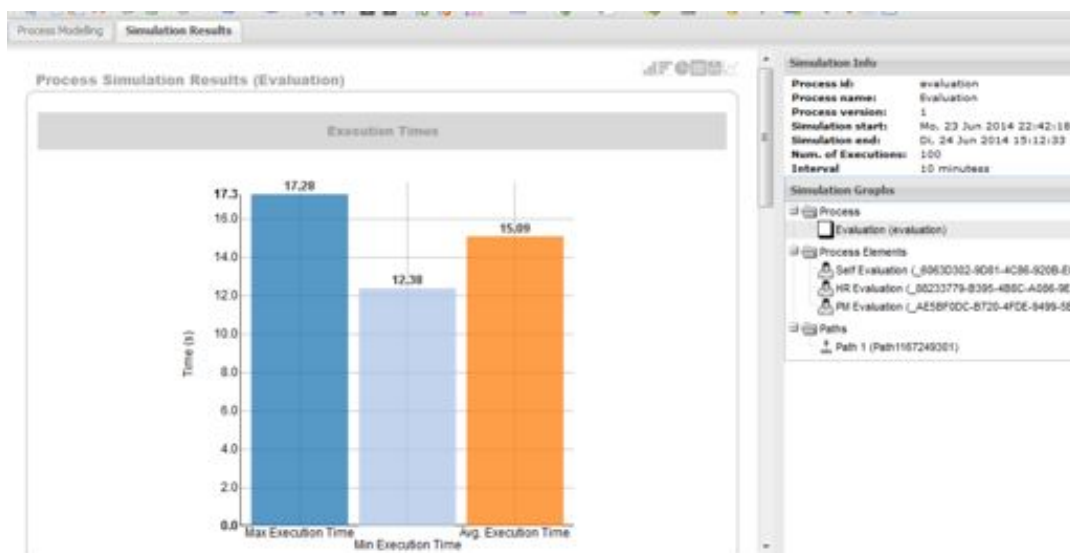


Abbildung 4.21: jBPM Process Designer Simulationsergebnisse

Die Ergebnisse können in unterschiedlichen Diagrammen dargestellt werden und umfassen folgende Punkte.

- Prozessergebnisse:  
Ausführungszeit, Aktivität der Instanzen, Gesamtkosten
- Humen Task Ergebnisse:  
Ausführungszeit, Ressourcenverbrauch, Ressourcenkosten
- Alle anderen Elemente: wie z.B. Ausführungszeit
- Prozess-Pfade: wie z.B. Pfade der Ausführung  
(vgl. jBPM team 2014b)



## 4.3 Activiti

Das Activiti Projekt wurde am 17. Mai 2010 von Alfresco<sup>20</sup> gestartet. Es ist eine BPM Suite unter der Apache Lizenz Version 2<sup>21</sup>. Die Activiti Engine unterstützt nativ BPMN 2.0 Prozessmodelle. Tom Baeyens und Joram Barrez waren Hauptentwickler der RedHat JBoss jBPM Workflow Engine und führten die Entwicklung des Projektes. Unterstützt wird das Projekt u.a. von Signavio, SprintSource und Camunda. Der komplett neu entwickelte Quellcode startete mit der Versionsnummer 5. Grund hierfür ist, dass die beiden Hauptentwickler so auf ihre Erfahrungen von jBPM Version 1 bis 4 hinweisen wollten. Die aktuelle Version ist 5.15.1. (vgl. Rademakers 2012, S. xvii ff.) (vgl. Baeyens 2010)

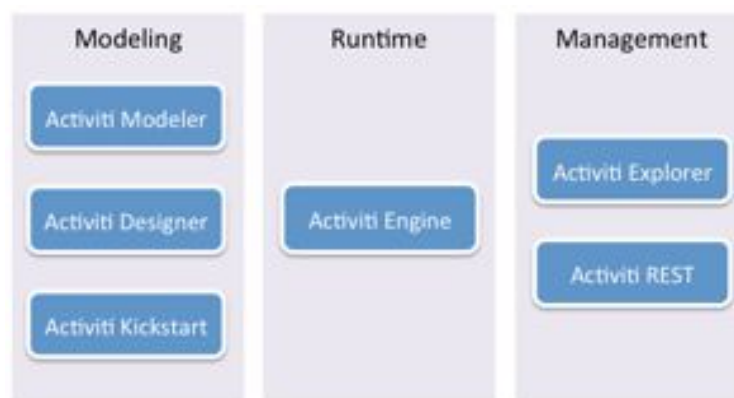


Abbildung 4.22: Activiti Komponentenübersicht (Activiti n.d.)

Die Activiti Komponenten sind in die drei Kategorien *Management*, *Runtime* und *Modeling* geteilt. Zur Kategorie Management gehören die Activiti Explorer und Activiti REST Komponenten. Beides sind *.war* Archive, welche auf einem Java Application Server (AS) deployt werden können. Sie dienen der Steuerung der Activiti Engine. In der Kategorie *Runtime* befindet sich die Activiti Engine. Sie ist das Herzstück von Activiti und ist für die Bereitstellung und Ausführung der Prozessmodelle zuständig. Die Kategorie *Modeling* enthält den Activiti Modeler, Activiti Designer und Activiti Kickstart. Diese Komponenten dienen zur Modellierung der Prozessmodelle. (vgl. Activiti n.d.) (vgl. Rademakers 2012, S. 4 f.)

Der Activiti Modeler ist eine web-basierte open source Version der KIS BPM process solution<sup>22</sup>. Dieser speichert die Modelle in einem Datenbank-Repository. Der Modeler stellt keine Eingabe von Simulationsparameter und keine Simulationsfunktion zur Verfügung. Activiti Kickstart ist ein web-basiertes Tool. Mit

<sup>20</sup><https://www.alfresco.com>

<sup>21</sup><https://www.apache.org/licenses/LICENSE-2.0.html>

<sup>22</sup><http://www.kisbpm.com/>



der einfachen Benutzeroberfläche können schnell einfache prototypische Prozessmodelle erstellt werden. Auf die Komponenten Activiti Modeler und Activiti Kickstart wird nicht weiter eingegangen. (vgl. Activiti n.d.)

### 4.3.1 Activiti Designer

Der Activiti Designer ist ein Eclipse Plugin mit dem BPMN 2.0 Prozessmodelle grafisch modelliert, getestet und deployed werden können. In der Dokumentation sind aktuell keine Simulationsfunktionen zu entnehmen. Das bezieht sich auch auf das Hinterlegen von Simulationsparameter im Prozessmodell. Bei der Recherche wurde das Projekt Crystalball bzw. Activiti-Crystalball<sup>23</sup> entdeckt. Weitere Informationen zu diesem Projekt sind dem Kapitel 4.3.3 zu entnehmen. Um die prototypische Entwicklung einer Extension (deutsch Erweiterung) zu realisieren ist eine Installation des Activiti Designers notwendig.

#### 4.3.1.1 Activiti Designer Installation

Die aktuelle Version des Activiti Designers ist 5.14.1. Aus der Dokumentation ist zu entnehmen, dass die Eclipse Versionen ab Helios (Version 3.6) nicht unterstützt werden. Jedoch funktioniert die identische Installation und Verwendung, ohne ersichtliche Fehler, unter Eclipse Kepler Version 4.3.2. (vgl. Activiti 2014)

Nachdem Eclipse gestartet ist, kann über das Menü *Help / Install New Software...* und den *Add...* Button das Repository „<http://activiti.org/designer/update/>“ mit dem Namen „Activiti Designer“ hinzugefügt werden. Die Abbildung 4.23 zeigt die Auswahl der Repository Plugins.

Nachdem das Plugin ausgewählt worden ist, kann der *Next* Button gedrückt werden. Die Abhängigkeiten des Plugins werden aufgelöst und ggf. mit installiert. Der Activiti Designer hat eine feste Abhängigkeit zu Graphiti<sup>24</sup> Version 0.10.1. Ist bereits eine andere Version von Graphiti installiert, muss diese zuerst deinstalliert werden. Im Anschluss muss die EPL Version 1.0 akzeptiert und Eclipse neu gestartet werden.

Jetzt ist die *Activiti* Perspektive vorhanden. Es können über das Menü *File / New...* Activiti Projekte und BPMN 2.0 Diagramme angelegt werden. Zudem besteht die Möglichkeit Prozessmodelle zu validieren und als z.B. PNG oder JPG zu exportieren. Das Anlegen eines Activiti Projektes wurde beispielhaft mit Projekt *de.th.wildau.demo.activiti* umgesetzt. In der Abbildung 4.24 ist der

<sup>23</sup><http://gro-mar.github.io/activiti-crystalball/>

<sup>24</sup>Framework welches ein einfaches erstellen grafischer Editoren ermöglicht  
<http://projects.eclipse.org/projects/modeling.gmp.graphiti>

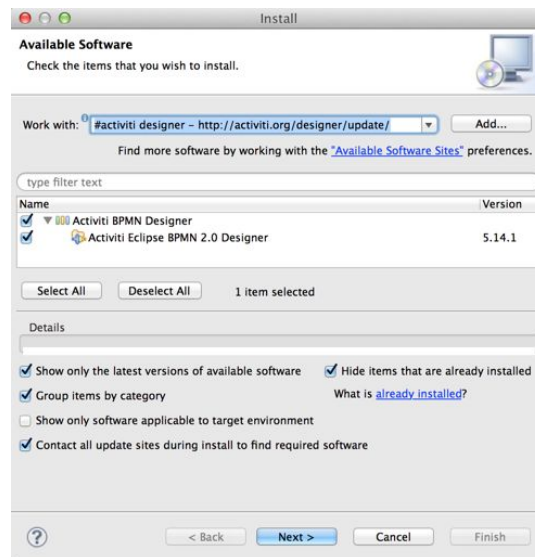


Abbildung 4.23: Activiti Designer Repository Plugin Installation

grafischen Editor, der JUnit test case (deutsch Testfall) sowie der Properties-Ansicht zu sehen. In der Properties-Ansicht können selektierte BPMN Elemente bearbeitet werden können.

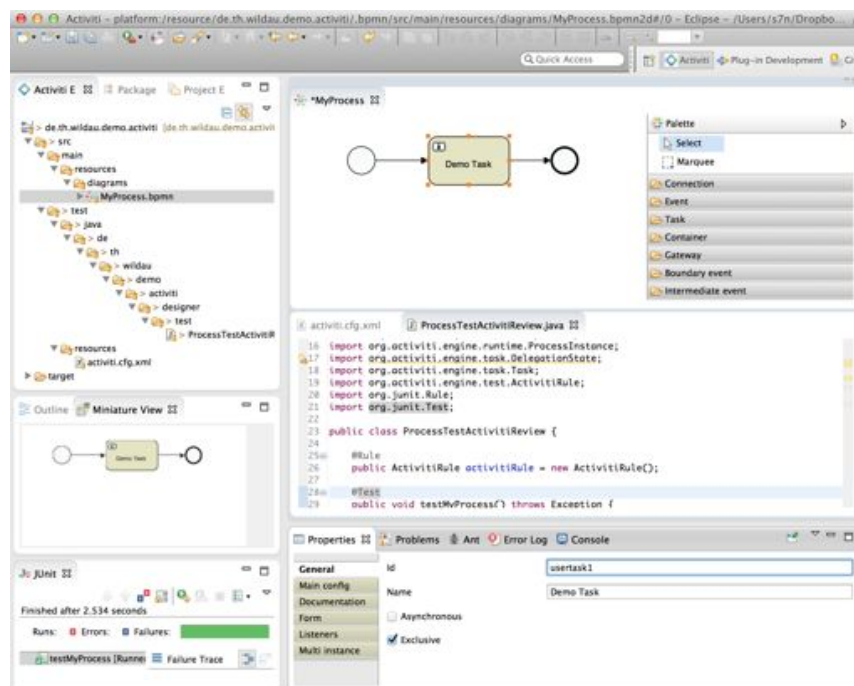


Abbildung 4.24: Activiti Designer Beispielprojekt

Der JUnit test case startet ein Prozessmodell in der Activiti Engine. Der Quellcode 4.5 zeigt den Aufruf der Activiti Engine.

```

26 ...
27 File file = new File("src/main/resources/diagrams/MyProcess.bpmn");

```

```

28 RepositoryService repositoryService = activitiRule.getRepositoryService();
29 Deployment deploy = repositoryService
30     .createDeployment()
31     .addInputStream("activitiReview.bpmn20.xml",
32         new FileInputStream(file.getAbsolutePath()))
33         .deploy();
34 RuntimeService runtimeService = activitiRule.getRuntimeService();
35 ...

```

Quellcode 4.5: Activiti Engine JUnit test case - ProcessTestActivitiReview.java

Das Öffnen von Prozessmodellen, die nicht mit diesem Editor erstellt worden sind, wurde wieder mit der Datei *ch7CreditAppSimulation.bpmn* durchgeführt. Das Öffnen erfolgt ohne Fehlermeldung. Jedoch wird kein grafisches Prozessmodell dargestellt. Dies zeigt die Abbildung 4.25.

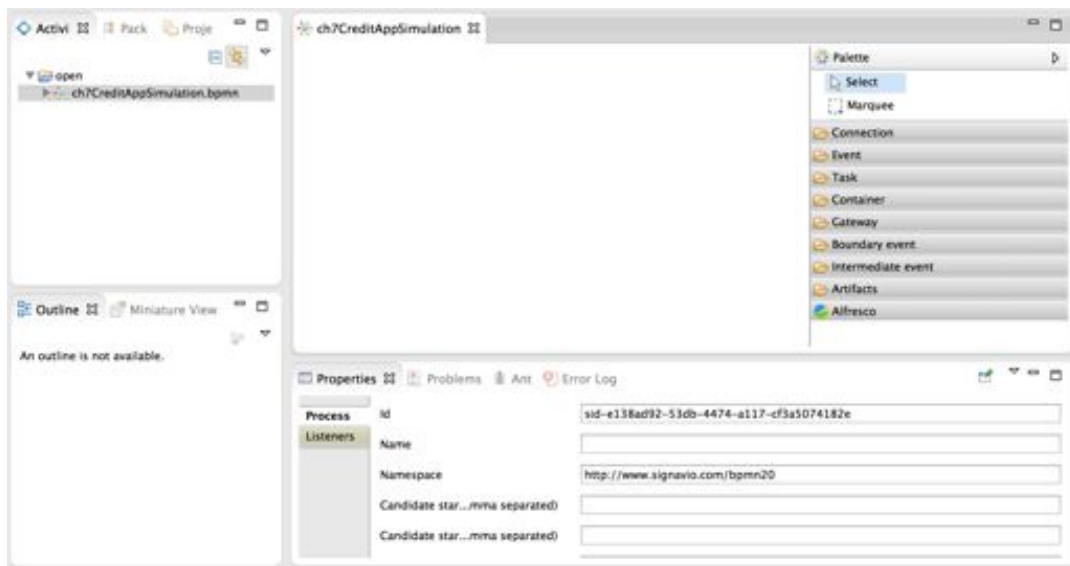


Abbildung 4.25: Activiti Designer Anzeige externer Datei

### 4.3.2 Activiti Designer Erweiterung

Der Activiti Designer kann um Funktionalitäten, die erforderlich oder branchenspezifisch sind, erweitert werden. Die möglichen Erweiterungen sind in die zwei Kategorien Paletten- und Validierung-/Export-Erweiterung unterteilt. Jede dieser Kategorien erfordert eine andere Herangehensweise bei der Umsetzung. (vgl. Activiti 2014)

Mit den Erweiterungsmöglichkeiten sind mindestens zwei Umsetzungen möglich. Zum einen kann die Re-Implementierung der BPMN Elementpalette umgesetzt werden. Dadurch können Simulationsparameter hinterlegt werden. Die zweite Variante ist das Exportieren des Prozessmodelles und eine separate Anreicherung mit Simulationsparametern. Unter der Apache Lizenzierung besteht

die Möglichkeit direkt Erweiterungen bei zu tragen. Die folgenden zwei Abschnitte zeigen eine prototypische Implementierung dieser beiden Erweiterungsmöglichkeiten.

#### 4.3.2.1 Palette

Um diese Erweiterungen umzusetzen, muss eine .jar<sup>25</sup> Datei erzeugt und in den installierten Activiti Designer eingebunden werden. Wie diese Integration funktioniert, wird am Ende dieses Kapitels beschrieben.

Um diese .jar Datei zu erzeugen, wird in Eclipse ein Maven Projekt angelegt. In der *pom.xml* Datei muss das *org.activiti.designer.integration* als Abhängigkeit definiert werden. Diese Abhängigkeit stellt die API zum Activiti Designer Palette bereit. Der Quellcode 4.6 zeigt den Ausschnitt der *pom.xml*.

```

22 ...
23 <repositories>
24     <repository>
25         <id>Activiti</id>
26         <url>https://maven.alfresco.com/nexus/content/groups/public/</url>
27     </repository>
28 </repositories>
29
30 <dependencies>
31     <dependency>
32         <groupId>org.activiti.designer</groupId>
33         <artifactId>org.activiti.designer.integration</artifactId>
34         <version>5.14.1</version>
35         <scope>compile</scope>
36     </dependency>
37 </dependencies>
38 ...

```

Quellcode 4.6: Activiti Designer Palette Erweiterung - pom.xml

Der Quellcode 4.7 zeigt wie in dieser Erweiterung alle bestehenden Elemente der Palette ausgeblendet werden.

```

13 ...
14 public class PaletteCustomizer extends AbstractDefaultPaletteCustomizer {
15     @Override
16     public List<PaletteEntry> disablePaletteEntries() {
17         List<PaletteEntry> result = new ArrayList<PaletteEntry>();
18         result.add(PaletteEntry.ALL);
19         return result;
20     }
21 }

```

Quellcode 4.7: Activiti Designer Palette Erweiterung - PaletteCustomizer.java

<sup>25</sup>Java Archive

Der Quellcode 4.8 zeigt wie ein neues Element und Gruppe hinzugefügt werden kann. Aus der Beispielimplementierung und dem User Guide sind weitere Property-Typen zu entnehmen. Die Gruppe *Connection*, mit den Elementen *SequenceFlow* und *Association* lassen sich nicht entfernen. (vgl. Activiti 2014)

```
1 ...
2 @Runtime(javaDelegateClass = "org.th.wildau.activiti.designer.extension.
   palette.task.ThWildauTask")
3 @Help(displayHelpShort = "Help short", displayHelpLong = "Help long")
4 public class ThWildauTask extends AbstractCustomServiceTask {
5 ...
6     @Property(type = PropertyType.MULTILINE_TEXT, displayName = "Comments"
7         , required = true, defaultValue = "Your comment")
8     @Help(displayHelpShort = "Provide comments", displayHelpLong = "You
9         can add comments to the node to provide a brief description.")
10    private String comments;
11 ...
12    @Override
13    public String contributeToPaletteDrawer() {
14        return "TH-Wildau Palette";
15    }
16 ...
17    @Override
18    public String getName() {
19        return "TH Wildau node";
20    }
21 ...
22    @Override
23    public String getSmallIconPath() {
24        return "icons/th.png";
25    }
26 }
```

Quellcode 4.8: Activiti Designer Palette Erweiterung - ThWildauTask.java

Bevor die Erweiterung integriert werden kann, muss das Projekt via Rechtsklick auf *Run As... / Maven install* gebaut werden. Nachdem dieser Schritt erfolgreich abgeschlossen ist, muss die erzeugte Datei *target/org.th.wildau.activiti.designer.extension-0.0.1-SNAPSHOT.jar* außerhalb des Workspaces kopiert, um dann via Menü *Eclipse / Preferences... / Java / Build Path / User Library* hinzugefügt zu werden. In diesem Einstellungsfenster erscheint (mit installiertem Activiti Designer) der Eintrag *Activiti Designer Extensions*. Ist dieser Eintrag ausgewählt, so kann über den Button *Add External JARs...* die .jar Datei, welche außerhalb des Workspaces liegt, hinzugefügt werden. Dies zeigt die Abbildung 4.26. (vgl. Activiti 2014)

Damit der Editor korrekt mit der Erweiterung arbeitet, muss sich das BPMN Prozessmodell in einem Activiti Projekt befinden. Die integrierte Erweiterung zeigt Abbildung 4.27.

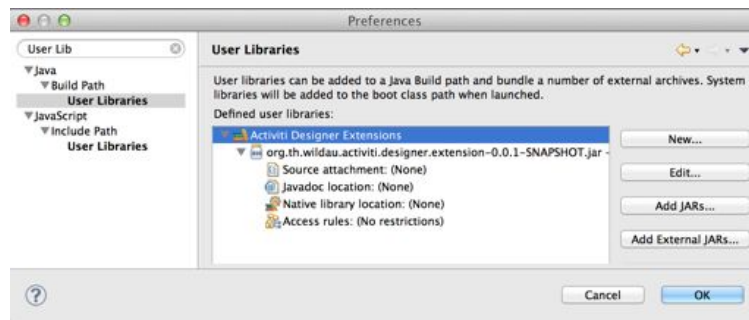


Abbildung 4.26: Activiti Designer Integration der Palettenerweiterung

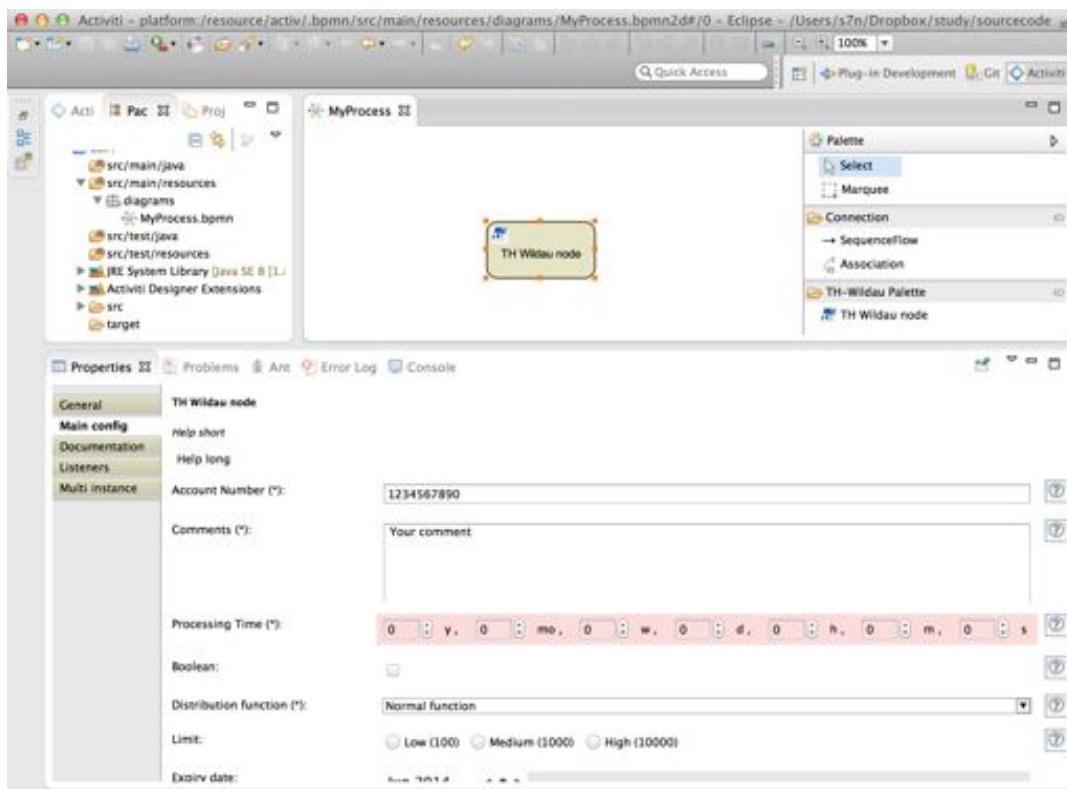


Abbildung 4.27: Activiti Designer Extension Palette Customizing

#### 4.3.2.2 Validierung und Export

Der Activiti Designer bietet einen eingebauten Erweiterungspunkt, in dem individuelle Validierungen und Exporte integriert werden können. Eine grundlegende Validierung der Standard BPMN Elemente ist bereits eingebaut. Durch die Erweiterung können beispielsweise selbst erstellte Elemente, Modellierungskonventionen oder Eigenschaften überprüft werden. (vgl. Activiti 2014)

Dies wird mit der prototypischen Plugin-Entwicklung *de.th.wildau.demo.activiti.designer.extension.vb* verdeutlicht. Die Implementierung erfolgt nach dem User Guide Beispiel. Sie läuft ohne Fehler. Das Plugin ist auch in der Eclipse IDE korrekt eingebunden und das Beispielenü wird angezeigt. Dies zeigt die Ab-

bildung 4.28. Da der Beispielsexport nicht angezeigt wird, konnte dieser nicht getestet werden. Für die genaue Ursachenforschung wären beispielsweise das Forum oder die Mailingliste zu kontaktieren.



Abbildung 4.28: Activiti Designer Extension Validierung und Export

Die Java Klasse *ThExportMarshaller.java* erweitert die Abstrakte Klasse *AbstractExportMarshaller*. Dadurch steht u.a. die Methode *doMarshallDiagram* zur Verfügung. Diese Methode muss für die Exportfunktionalität überschrieben werden. In der Methode kann auf das Prozessmodell zugegriffen und selbstständig eine Export-Datei geschrieben werden. Der Export wird in dem Menü zur Verfügung gestellt.

Die Java Klasse *ThProcessValidator* erweitert die Abstrakte Klasse *AbstractProcessValidator*. Dadurch steht u.a. die Methode *validateDiagram* zur Verfügung. Diese Methode muss für die Validierungsfunktionalität überschrieben werden. In der Methode können eigenständige Validierungen vorgenommen werden. Die Validierung wird im Zuge der grundlegenden Validierung ausgeführt. Der Quellcode 4.9 zeigt den Ausschnitt, in dem die Prozessanzeige der Validierung, zeitlich verzögert wird.

```

30 ...
31 @Override
32 public boolean validateDiagram(Diagram diagram, IProgressMonitor arg1) {
33     // demonstrates progress using
34     for (int i = 0; i < 5; i++) {
35         try {
36             Thread.currentThread().sleep(20001);
37         } catch (final InterruptedException ex) {
38             ex.printStackTrace();
39         }
40         getDiagramWorkerContext().getProgressMonitor().worked(5);
41     }
42     getDiagramWorkerContext().getProgressMonitor().done();
43 ...

```

Quellcode 4.9: Activiti Designer VE Erweiterung - ThProcessValidator.java



### 4.3.3 Crystalball

Crystalball wird seit 2012 von Martin Grofčík<sup>26</sup> entwickelt und ist eine eigenständige Simulations-Engine für die Activiti BPM Suite. Das Projekt wurde am 04. Juni 2014 in das Activiti Projekt aufgenommen<sup>27</sup>. Damit wird es vermutlich in der nächsten Activiti Version 5.16 integriert sein. Die Simulations-Engine basiert auf der Activiti Engine und kann für Entscheidungsprozesse produktiver Workflows, Verständnis und Optimierung sowie zur Übung (bevor das Prozessmodell in die produktive Phase gehen) genutzt werden. Dabei muss kein separates Simulationsmodell erstellt werden. Die Activiti Process Engine wird überwacht um Informationen für die Berichterstellung zu sammeln. Die Simulations-Engine benötigt wenig Start- und Prozessmodellparameter. Benötigte Daten werden aus der History Tabelle der Datenbank gewonnen. (vgl. Grofčík 2014a)

Die Installation von Crystalball in der Activiti Version 5.16-SNAPSHOT via Maven funktioniert in der Windows Umgebung, jedoch nicht unter Mac OS. Die Problematik der Plattformunabhängigkeit ist Activiti<sup>28</sup> bekannt. Grund hierfür ist die intern verwendete Java Bibliothek zur Erzeugung der Diagrammbilder. Sie erzeugt auf verschiedenen Systemen unterschiedlich Grafiken, wodurch die Testfälle unter Mac OS fehl schlagen. Der Quellcode 4.10 zeigt einen Ausschnitt der Fehlermeldung.

```

1  ...
2  -----
3  T E S T S
4  -----
5  Running org.activiti.crystalball.examples.tutorial.step01.
   FirstSimulationRunTest
6  Tests run: 1, Failures: 0, Errors: 1, Skipped: 0, Time elapsed: 0.116 sec <<<
   FAILURE!
7  Running org.activiti.crystalball.process.SimulationRunTaskTest
8  Tests run: 1, Failures: 0, Errors: 1, Skipped: 0, Time elapsed: 0.009 sec <<<
   FAILURE!
9  Running org.activiti.crystalball.simulator.SimpleEventCalendarTest
10 Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.061 sec
11 Running org.activiti.crystalball.simulator.SimpleSimulationRunTest
12 Tests run: 6, Failures: 0, Errors: 6, Skipped: 0, Time elapsed: 0.02 sec <<<
   FAILURE!
13 Running org.activiti.crystalball.simulator.SimulationEventComparatorTest
14 Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.005 sec
15 Running org.activiti.crystalball.simulator.impl.
   MultiInstanceScriptEventHandlerTest
16 Tests run: 1, Failures: 0, Errors: 1, Skipped: 0, Time elapsed: 0.004 sec <<<
   FAILURE!
17 Running org.activiti.crystalball.simulator.impl.ScriptEventHandlerTest

```

<sup>26</sup><https://github.com/gro-mar>

<sup>27</sup><https://github.com/Activiti/Activiti/pull/267>

<sup>28</sup><http://forums.activiti.org/content/processdiagramgeneratorgeneratediagram-platform-dependent>



```

18 Tests run: 1, Failures: 0, Errors: 1, Skipped: 0, Time elapsed: 0.004 sec <<<
    FAILURE!
19 Running org.activiti.crystalball.simulator.impl.playback.
    PlaybackProcessStartTest
20 Tests run: 4, Failures: 0, Errors: 4, Skipped: 0, Time elapsed: 0.01 sec <<<
    FAILURE!
21 Running org.activiti.crystalball.simulator.impl.playback.PlaybackRunTest
22 Tests run: 1, Failures: 0, Errors: 1, Skipped: 0, Time elapsed: 0.002 sec <<<
    FAILURE!
23 Running org.activiti.crystalball.simulator.impl.replay.ReplayRunTest
24 Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.81 sec
25
26 Results :
27
28 Tests in error:
29     testSimulationRun(org.activiti.crystalball.examples.tutorial.step01.
        FirstSimulationRunTest)
30     testBasicSimulationRun(org.activiti.crystalball.process.
        SimulationRunTaskTest)
31 ...

```

Quellcode 4.10: Activiti Crystalball Kommando mvn clean install unter Mac OS

Das Projekt kann laut Dokumentation in der Design- und Monitoringphase eingesetzt werden. In der Designphase kann die Test-Prozesspfadabdeckung überprüft werden. Als Ergebnis wird eine Grafik erzeugt. Diese Grafik markiert die Prozessmodellelemente in unterschiedlichen Farben. Grün markierte Elemente wurden mehr als 8, gelbe zwischen 1-7 und rot 0-mal durchlaufen. Ein solches Beispiel ist in Abbildung 4.29 zu sehen.

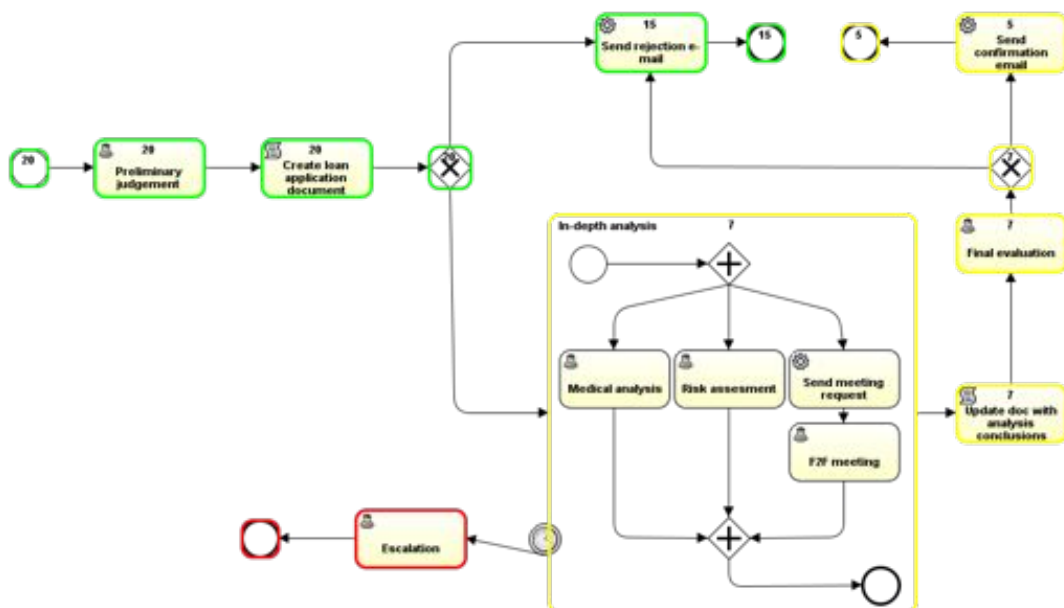


Abbildung 4.29: Activiti Crystalball Testabdeckung

In der Monitoringphase wird die Activiti Process Engine überwacht. Mit den ge-

wonnenen Daten können Testfälle wiederholt werden. Wodurch Prozessmodelle auf bestimmte Szenarien hin optimiert werden können. Es können hier Eskalationsfallwahrscheinlichkeiten berechnet werden. Das Ergebnis wird ebenfalls grafisch dargestellt und ist in Abbildung 4.30 zu sehen.

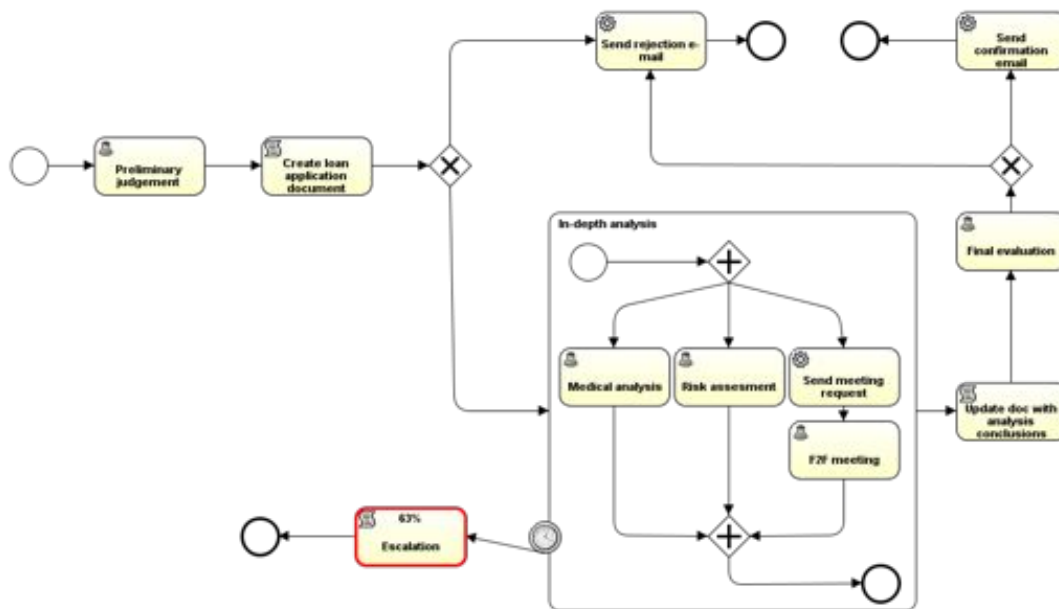


Abbildung 4.30: Activiti Crystalball Eskalationswahrscheinlichkeiten

Die letzte der drei genannten Funktion ist die Kostenanalyse. Die Kosten werden hier über eine separate XML-Datei konfiguriert. In dem angegebenen Beispiel ist dies die *mortgages-costsanalysis-h2-context.xml*<sup>29</sup> Datei. Zusätzlich wird im Prozessmodell das Schema *xmlns:sim="http://crystalball.org/simulation"*<sup>30</sup> definiert. An den Task Elementen wird zudem das *sim:behavior* Attribut definiert. In dem Attribut steht die Klasse, welche bei der Simulation aufgerufen wird und die Kosten protokolliert. Die Ergebnisse der Simulation werden in dem Beispiel in der H2 Datenbank *target/crystalball-costsanalysis.h2.db* gespeichert.(vgl. Grofcik 2014b)

Die Activiti Crystalball Ressourcen sind auf der CD in dem Ordner *resources* zu finden.

<sup>29</sup><https://github.com/gro-mar/activiti-crystalball/blob/master/examples/src/test/resources/org/activiti/crystalball/examples/costsanalysis/mortgages-costsanalysis-h2-context.xml>

<sup>30</sup><https://github.com/gro-mar/activiti-crystalball/blob/master/examples/src/main/process/org/activiti/crystalball/examples/mortgages/MortgageDemo-2-costsanalysis.bpmn>

## 4.4 Camunda

Camunda Service GmbH wurde am 18. März 2009 als Beratungsunternehmen mit Sitz in Berlin gegründet. Ende 2010 wurde die *camunda fox* Plattform entwickelt. Mitte März 2013 entfernte camunda den Namen fox von ihrer BPM-Plattform und stellte Software unter die Apache Lizenz Version 2.0. Nicht alle Werkzeuge wurden von Camunda selbst entwickelt, sondern stammen u.a. aus dem Activiti Projekt. Die open source BPM-Plattform besteht aus 5 Gruppen: Prozessausführung, Prozessdesign, Prozessbetrieb, Aufgabenmanagement und sonstiges. Die Abbildung 4.31 zeigt die Architekturübersicht der camunda Plattform. (vgl. Freund & Rücker 2012, S. 273 ff.) (vgl. Meyer 2013) (vgl. Humble 2013)

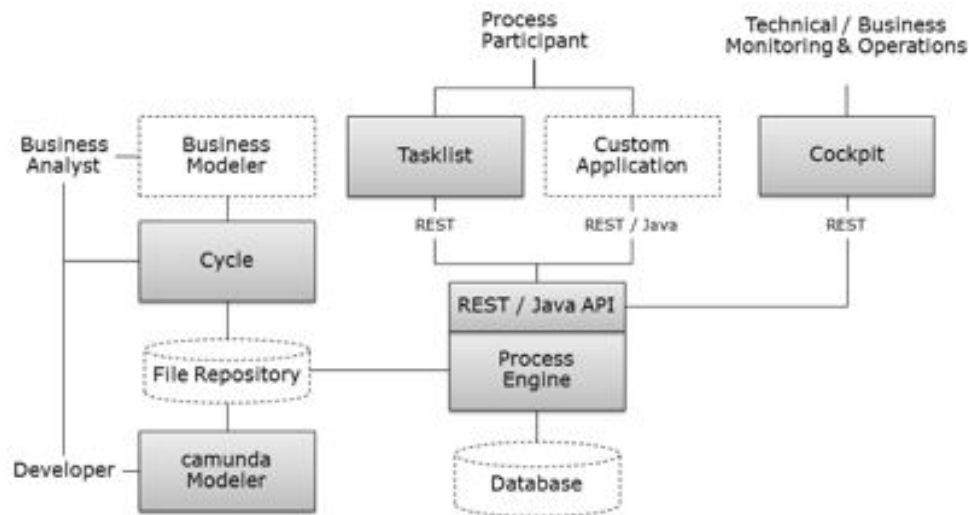


Abbildung 4.31: Camunda Architekturübersicht (Camunda 2014)

Der Eclipse basierte *camunda Modeler* gehört zur Gruppe des Prozessdesigns und die *Process Engine* zum Prozessausführung und -betrieb. Sie sind Forks (deutsch Abzweige) der gleichnamigen Activiti Projekte. Zur Process Engine gehört ein REST Service, ein Spring Framework, eine Contexts and Dependency Injection (CDI) Integration. Die Process Engine soll auf allen Java Enterprise Edition (Java EE) Servern laufen. Der *Cycle* dient dem BPM Projektmanagement und organisiert das Zusammenspiel zwischen dem Betrieb und der IT. Die *Tasklist* und das *Cockpit* gehören der Gruppe Aufgabenmanagement und Prozessbetrieb an. (vgl. Meyer 2013)

Neben diesen Grundkomponenten existieren eine Reihe weiterer Projekte, die auf <http://github.com/camunda> zu finden sind. Hierunter fallen diverse Anwendungsintegrationen wie zum Beispiel OSGi, Apache Camel und Elasticsearch, sowie Testintegrationen und das BPMN Model Interchange Working

Group (BPMN MIWG)<sup>31</sup> Projekt. Das *camunda-bpmn.js* Projekt welches die Verarbeitung, Ausführung und Anzeige von BPMN Version 2.0 Modellen ermöglicht, wird nicht mehr aktiv weiterentwickelt. Diese Stelle übernimmt das 2014 gestartete *bpmn.io*<sup>32</sup> Projekt. Das Projekt befindet sich noch in der Entwicklungsphase, steht unter der bpmn-js License<sup>33</sup> und wird in Zusammenarbeit mit Zalando<sup>34</sup> entwickelt.

Die camunda network community bietet wie die jBPM und Activiti Projekte eine reichhaltige Dokumentation mit Foren und Tutorials (deutsch Übungen), zudem im deutschsprachigen Raum eine Vielzahl von Treffen<sup>35</sup>. Aktuell liegt die Process Engine in der Version 7.1.3 und der camunda Modeler in der Version 2.7.0 vor. Welche Elemente aktuell von der Process Engine unterstützt werden, ist dem Dokumentation<sup>36</sup> zu entnehmen. Betrachtet man die Releasezyklen, seit der Process Engine Version 7.0.0 (31.08.2013), so ergibt sich bis zur Version 7.1.3 eine durchschnittliche Aktualisierungsrate von 13,3 Tagen. (vgl. Meyer 2014a)

Informationen zu Simulationsfunktionen wurden in der Dokumentation nicht gefunden. Im camunda Forum gibt es einen Hinweis auf das Crystalball Projekt von Martin Grofčík. Jedoch wurden diesbezüglich keine weiteren Informationen gefunden. Auf Nachfrage wurde im Forum bestätigt, dass es aktuell keine konkreten Pläne gibt eine Simulationsfunktion einzuführen. Jedoch wurde auf die Masterarbeit des Camunda Service GmbH Geschäftsführers Bernd Rücker<sup>37</sup> verwiesen. (vgl. Meyer 2014b) Diese trägt den Titel „Building an open source Business Process Simulation tool with JBoss jBPM“. Es wurde in dieser Masterarbeit ein Simulationwerkzeug auf Basis von jBPM und DESMO-J entwickelt, welches jedoch nicht den BPSim Standard berücksichtigt und es bietet keine grafische Benutzeroberfläche. (vgl. Meyer 2014b)

#### 4.4.1 Camunda Modeler

Wie bereits im vorigen Kapitel erwähnt, ist der camunda Modeler ein Fork des gleichnamigen Activiti Projektes und ein Eclipse Plugin zur Bearbeitung von BPMN 2.0 Modellen. Das Projekt liegt aktuell in der Version 2.6.0 vor und steht unter der Eclipse Public Licence Version 1.0<sup>38</sup>. Es kann BPMN 2.0 Modelle le-

<sup>31</sup><http://www.omgwiki.org/bpmn-miwg/doku.php?id=start>

<sup>32</sup><https://github.com/bpmn-io>

<sup>33</sup><http://bpmn.io/license/>

<sup>34</sup><http://tech.zalando.com>

<sup>35</sup><http://network.camunda.org/meetings>

<sup>36</sup><http://docs.camunda.org/latest/api-references/bpmn20/>

<sup>37</sup><http://camunda.com/download/20080201-bernd-ruecker-business-process-simulation-with-jbpm.pdf>

<sup>38</sup><http://www.eclipse.org/legal/epl-v10.html>

sen, schreiben, validieren und speichert die BPMN DI<sup>39</sup> Schema-Informationen. Zudem werden Prozess- und Kollaborations-Diagrammtypen unterstützt. In der aktuellen Version ist der camunda Modeler für Eclipse Kepler verfügbar. Die Komponente ist als Eclipse Plugin<sup>40</sup> und Standalone<sup>41</sup> Variante verfügbar. Die Eclipse Plugin Variante lässt sich wie das Activiti Plugin über die Update URL <http://camunda.org/release/camunda-modeler/update-sites/kepler/latest/site/> installieren. Die Standalone Variante ist eine minimale Eclipse Variante, mit dem integrierten camunda Modeler Plugin. Die Version kann von der Webseite<sup>42</sup> heruntergeladen werden.

Der camunda Modeler bietet eine ähnliche Modellierungspalette wie das Activiti Plugin. Hinzu kommt, dass neben der grafischen Darstellung der Modelle auch das darunterliegende BPMN 2.0 XML angezeigt werden kann. Für die XML Ansicht ist am unteren linken Rand der Tab *Source* verfügbar. Die beiden Tabs *Design* und *Source* werden im Hintergrund automatisch synchronisiert, so dass Änderungen umgehend sichtbar sind. Die Abbildung 4.32 zeigt die camunda Modeler Oberfläche.

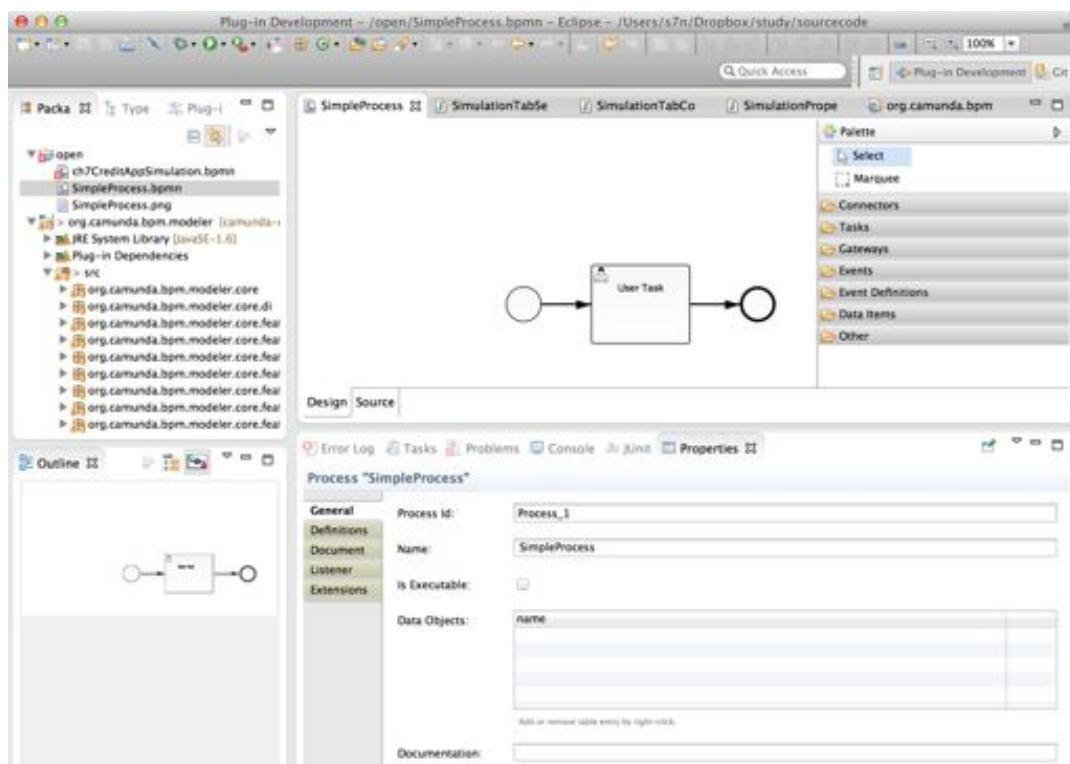


Abbildung 4.32: Camunda Modeler Ansicht

<sup>39</sup>BPMN 2.0 Diagram Interchange Extensible Markup Language Schema

<sup>40</sup><https://github.com/camunda/camunda-modeler>

<sup>41</sup><https://github.com/camunda/camunda-modeler-standalone>

<sup>42</sup><http://camunda.org/bpmn/tool/download/>

Das Öffnen von anderen BPMN 2.0 Modellen, die nicht mit dem Editor erstellt wurden, funktioniert. Für diesen Test wurde wieder die Datei *ch7CreditAppSimulation.bpmn* verwendet. Es erscheint, wie in dem jBPM Plugin eine Fehlermeldung, dass ein Import nicht durchgeführt werden konnte. Es wird jedoch das Diagramm korrekt angezeigt. Bei der integrierten Validierung tritt der Fehler, identisch zum jBPM Plugin, in Zeile 291 auf. Diese Fehlermeldung lautet „FeatureNotFoundException: Feature 'processSimulationInfo' not Found.“. Die Abbildung 4.33 und 4.34 zeigen die Fehlermeldung und das geöffnete Prozessmodell.

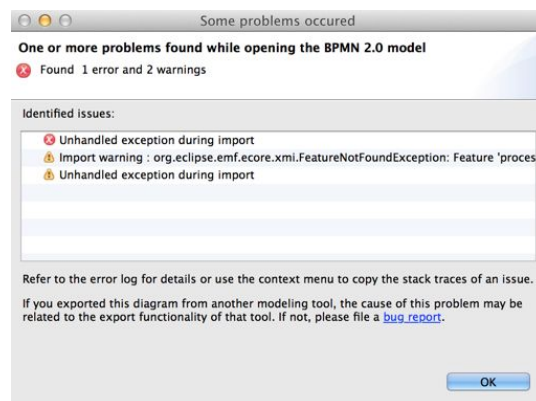


Abbildung 4.33: Camunda Modeler Fehler beim öffnen der externen Datei

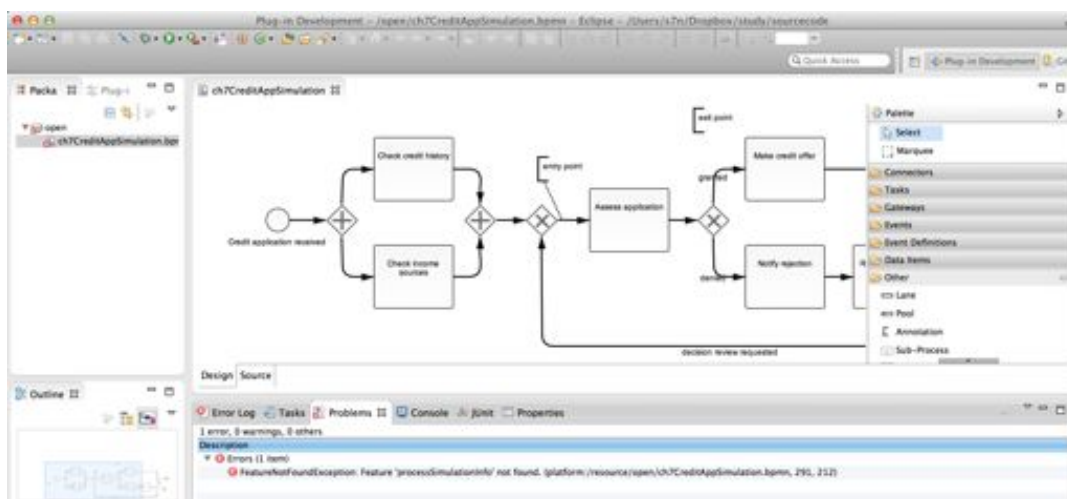


Abbildung 4.34: Camunda Modeler externe Datei geöffnet

Aufgrund des Forks ist camunda modeler identisch erweiterbar wie der Activiti Modeler. Eine prototypische Implementierung wurde nicht umgesetzt. Wie dies jedoch im Detail funktioniert, ist der gut strukturierten Dokumentation<sup>43</sup> zu entnehmen.

<sup>43</sup><http://docs.camunda.org/latest/real-life/how-to/#modeler-creating-custom-tasks-for-camunda-modeler>

## 4.4.2 BPMN Model API

Die camunda Plattform ist Modular aufgebaut und bietet eine Vielzahl separater APIs. In diesem Kontext sind zwei Projekte *camunda-bpmn-model*<sup>44</sup> und *camunda-xml-model*<sup>45</sup> interessant. Diese beiden APIs wurden in dem Projekt *de.th.wildau.demo.camunda.model.api* betrachtet.

Die Überlegung ist hier die Realisierung einer Java API, die es ermöglicht, programmatisch spezifizierte Simulationsparameter im Prozessmodell zu hinterlegen. Dazu muss der XML Namespace und das Schema hinterlegt werden. Im Anschluss muss eine Java Klasse bereit gestellt werden, die die Parameter entgegen nimmt und in das Model integriert. Somit kann ein separates Werkzeug ein BPMN Prozessmodell mit Simulationsparametern erweitern.

Mit dem Projekt *camunda-bpmn-model* können via Java BPMN Modelle gelesen, geschrieben und erstellt werden. Beim Erstellen werden jedoch keine BPMN DI<sup>46</sup> Informationen geschrieben, wodurch eine spätere grafische Anzeige nicht möglich ist. Die fluent (deutsch fließende) API ermöglicht eine schnelle Modelerstellung. Der Quelltext 4.11 zeugt die Prozessmodellerstellung. Das erzeugte Prozessmodell ist im Quellcode 4.12 zu sehen.

```

62 ...
63 BpmnModelInstance modelInstance = Bpmn.createProcess().id("ProcessId")
64     .name("ProcessName")
65     .startEvent()
66     .name("Start")
67     .userTask()
68     .name("User Task")
69     .camundaAssignee("Fred")
70     .endEvent().done();
71 ...
72 Bpmn.writeModelToFile(new File("src/main/resources/fluentApiProcess.bpmn"),
73     modelInstance);
74 ...

```

Quellcode 4.11: Camunda Model API - FluentApi.java

```

1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <definitions xmlns:camunda="http://activiti.org/bpmn" targetNamespace="http://
   camunda.org/examples" xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL">
3   <process id="ProcessId" name="ProcessName">
4     <startEvent id="startEvent_9f...77" name="Start">
5       <outgoing>sequenceFlow_fae...10</outgoing>
6     </startEvent>
7     <userTask camunda:assignee="Fred" id="userTask_5ff...6a" name="User Task">
8       <incoming>sequenceFlow_fae...10</incoming>
9       <outgoing>sequenceFlow_d7e...fe</outgoing>

```

<sup>44</sup><https://github.com/camunda/camunda-bpmn-model>

<sup>45</sup><https://github.com/camunda/camunda-xml-model>

<sup>46</sup>Diagramm Information



```

10     </userTask>
11     <sequenceFlow id="sequenceFlow_fae...10" sourceRef="startEvent_9f...77"
        targetRef="userTask_5f...6a"/>
12     <endEvent id="endEvent_d6...c9">
13         <incoming>sequenceFlow_d7...fe</incoming>
14     </endEvent>
15     <sequenceFlow id="sequenceFlow_d7e...fe" sourceRef="userTask_5f...6a"
        targetRef="endEvent_d6...c9"/>
16 </process>
17 </definitions>

```

Quellcode 4.12: fluentApiProcess.bpmn

Das programmatische Hinzufügen des BPSim Namespaces funktioniert. Die ist im Quellcode 4.13 zu sehen. Verschiedene Schematas sind fest im Quelltext definiert. Es ist momentan nicht möglich eigene Schematas via API hinzuzufügen. Das Hinzufügen eines Namespaces funktioniert hingegen und ist im Quellcode 4.13 und 4.14 dargelegt.

```

20 ...
21 modelInstance.getDocument().registerNamespace("bpsim", "http://www.bpsim.org/
    schemas/1.0");
22 ...

```

Quellcode 4.13: Auszug aus ModelApiExtElement.java

```

1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <bpmn2:definitions xmlns:bpmn2="http://www.omg.org/spec/BPMN/20100524/MODEL"
    xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI" xmlns:bpsim="http://
    www.bpsim.org/schemas/1.0" xmlns:camunda="http://activiti.org/bpmn"
    xmlns:dc="http://www.omg.org/spec/DD/20100524/DC" xmlns:di="http://www.omg.
    org/spec/DD/20100524/DI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" exporter="camunda modeler" exporterVersion="2.6.0"
    expressionLanguage="http://www.w3.org/1999/XPath" id="_oVY-
    oAD8EeSaj_dHbxBDXA" targetNamespace="http://activiti.org/bpmn" typeLanguage
    ="http://www.w3.org/2001/XMLSchema" xsi:schemaLocation="http://www.omg.org/
    spec/BPMN/20100524/MODEL BPMN20.xsd">
3 ...

```

Quellcode 4.14: Erweitertes Prozessmodell  
modelApiExtElementProcess.bpmn

Das andere camunda Projekt *camunda-xml-model* ermöglicht die Implementierung eines Metamodelles via Java. Hier wurde versucht ein eigenes Simulationselement zu erzeugen, welches dann im XML gespeichert wird. Wodurch das einfache Hinzufügen von Simulationsparametern via Java-Quellcode erfolgen kann. Die prototypische Implementierung ist in der Datei *SimElement.java* zu finden. Leider schlägt das Erstellen des Modelles, mit der Fehlermeldung „org.camunda.bpm.model.xml.impl.util.ModelTypeException: Model element type simType is abstract and no instances can be created.“ fehl. Aufgrund der



mangelnder Dokumentation zu dieser Thematik, wurde an dieser Stelle die prototypische Entwicklung abgebrochen.

## 4.5 Alternativen

Neben den bisher betrachteten Eclipse basierten Ansätzen, wurden die beiden Anwendungen Yaoqiang und BIMP betrachtet.

### 4.5.1 Yaoqiang

Der Yaoqiang BPMN Editor ist ein Java basiertes Modellierungswerkzeug, welches seit 2012 entwickelt wird. Seit 2013 gibt es die Version 3.x. Diese wird kommerziell<sup>47</sup> vertrieben. Die Version 2.x steht unter der GPL<sup>48</sup> v3 Lizenz. Sie wird von „blenta“ entwickelt. Leider konnte zu diesem Modellierungswerkzeug keine separate Dokumentation gefunden werden. Es existiert ein Forum, welches jedoch nicht umfangreich ist. Der Projektseite ist zu entnehmen, dass folgende Funktionen unterstützt werden.

- Im- und Export von OMG BPMN 2.0 Dateien
  - Syntax Validierung
  - BPMN 2.0 DI Informationen
  - Plugin Architektur (z.B. für Activiti)
  - Ausführbare BPMN Simulation
  - Import von Visio .vdx und GraphML .graphml Dateien
  - Export in diverse Bildformaten und OpenDocument Text (.odt)
- (vgl. Blenta n.d.)

Die aktuelle Version kann unter <http://sourceforge.net/projects/bpmn/files/2.2-GPLv3/> heruntergeladen werden. Im Anschluss kann die heruntergeladene .jar Datei via Terminal gestartet werden. Nachdem das Terminal geöffnet und zu der Datei navigiert wurde, kann diese mit dem Befehl „java -jar yaoqiang-bpmn-editor-VERSIONSNUMMER.jar“ gestartet werden. Die Abbildung 4.35 zeigt die gestartete Anwendung.

Der Editor ist lediglich für die reine Modellierung geeignet, da sich über die Oberfläche keine Eigenschaften definieren lassen. Es gibt die drei Tab-Ansichten Organisation, BPMN View und das Diagram. In der Ansicht Organisation kann die Organisationsstruktur modelliert werden. Die Ansicht BPMN View stellt den XML-Quellcode des BPMN Diagrammes und die Diagrammansicht stellt

<sup>47</sup><http://bpmn.sourceforge.net/>

<sup>48</sup><https://www.gnu.org/copyleft/gpl.html>

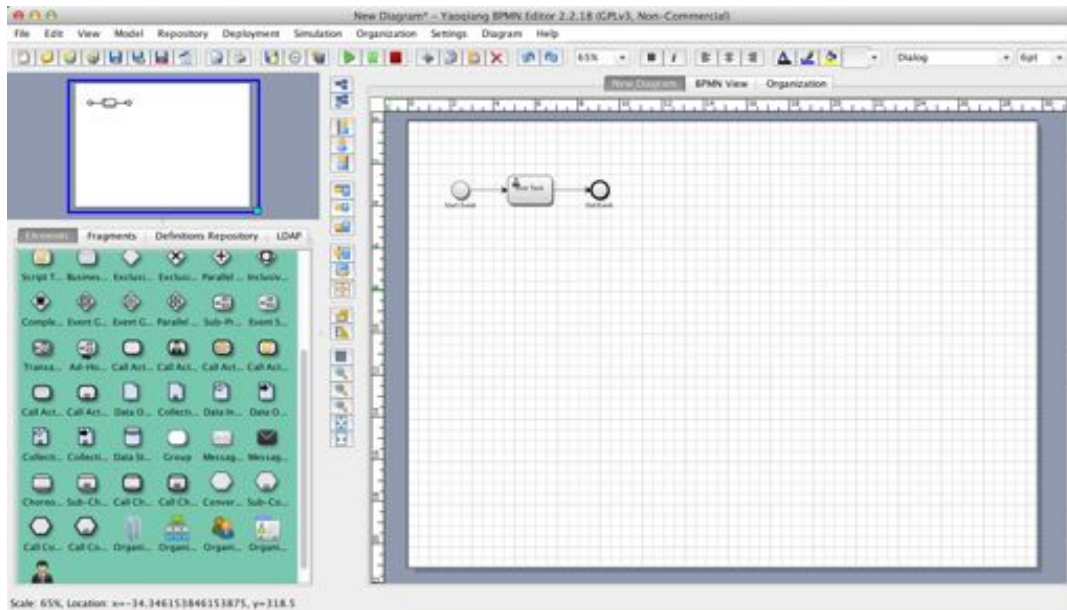


Abbildung 4.35: Yaoqiang Anwendung Version 2.2.18

das BPMN grafisch dar. Für die Modellierung stehen im Editor 61 Elemente zur Verfügung. Das Öffnen der *ch7CreditAppSimulation.bpmn* Datei funktioniert nicht. Die Abbildung 4.36 zeigt die Fehlermeldungen.

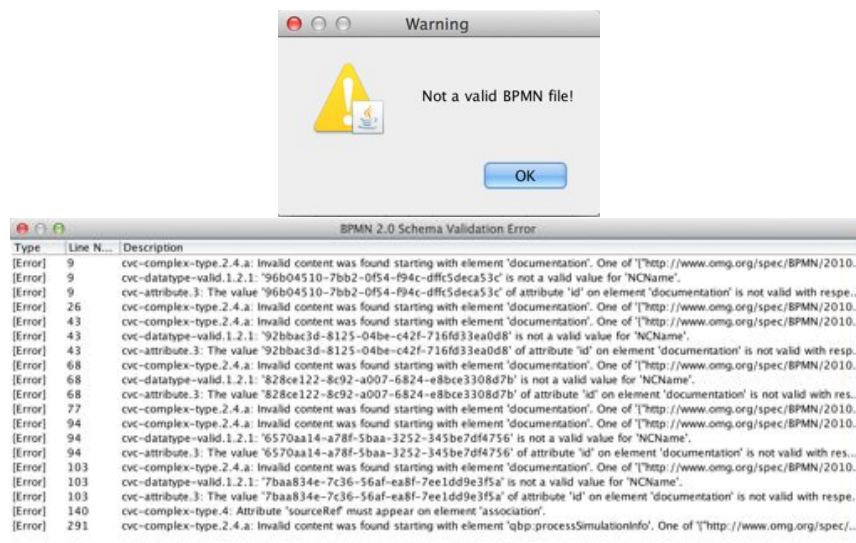


Abbildung 4.36: Yaoqiang Fehlermeldung beim Öffnen

Die Simulationsfunktion konnte in der Version 2.2.18 nicht gestartet werden. Im Terminal erscheint folgende Fehlermeldung.

```
1 Exception in thread "AWT-EventQueue-0" java.lang.NoClassDefFoundError: sun/org
  /mozilla/javascript/internal/Scriptable
2     at org.yaoqiang.bpmn.editor.BPMNEditor.startSimulation(Unknown Source)
3     at org.yaoqiang.bpmn.editor.action.ModelActions.actionPerformed(
      Unknown Source)
```

```

4      at org.yaoqiang.bpmn.editor.swing.BaseEditor$14.actionPerformed(
      Unknown Source)
5      at javax.swing.AbstractButton.fireActionPerformed(AbstractButton.
      java:2022)
6      at javax.swing.AbstractButton$Handler.actionPerformed(AbstractButton.
      java:2346)
7      at javax.swing.DefaultButtonModel.fireActionPerformed(
      DefaultButtonModel.java:402)
8      ...
9  Caused by: java.lang.ClassNotFoundException: sun.org.mozilla.javascript.
      internal.Scriptable
10     at java.net.URLClassLoader$1.run(URLClassLoader.java:372)
11     at java.net.URLClassLoader$1.run(URLClassLoader.java:361)
12     at java.security.AccessController.doPrivileged(Native Method)
13     at java.net.URLClassLoader.findClass(URLClassLoader.java:360)
14     at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
15     at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:308)
16     at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
17     ... 40 more

```

#### Quellcode 4.15: Yaoqiang Fehlermeldung der Simulation

Der Fehler deutet darauf, dass die interne Klasse *Scriptable* nicht geladen werden konnte. Im Forum wurde über diese Fehlermeldung oder die Simulationsfunktion kein Eintrag gefunden. Die Bedeutung der Simulationsfunktion darf hinterfragt werden, da keine Möglichkeit gefunden wurde Simulationsparameter zu hinterlegen. Aufgrund fehlender Dokumentation wird die Plugin-Schnittstelle nicht weiter betrachtet.

### 4.5.2 BIMP

BIMP ist ein in Java geschriebenes, web-basiertes BPMN Simulationstool. Die Entwicklung ging von Marlon Dumas<sup>49</sup> aus, welcher Software Engineering Professor der Universität von Tartu in Estland ist. Der erste Quellcode<sup>50</sup> geht auf den September 2011 zurück und steht unter der GNU General Public License (GPL) Version 3. Das Tool kann selbst aufgesetzt oder direkt über die Webseite<sup>51</sup> verwendet werden. Signavio Academic<sup>52</sup> verwendet BIMP für die Simulation.

Die Herangehensweise des BIMP Werkzeuges ist eine andere als des EPC-Simulators. Mit ihm können keine Prozesse modelliert, sondern ausschließlich mit Simulationsparametern angereichert und simuliert werden. Als Beispiel wird das *credit application scenario*<sup>53</sup> Prozessmodell verwendet. (

<sup>49</sup><http://math.ut.ee/~dumas/>

<sup>50</sup><https://code.google.com/p/bimp-simulator/source/list?path=/.project>

<sup>51</sup><http://bimp.cs.ut.ee/simulator>

<sup>52</sup><http://www.signavio.com/bpm-academic-initiative/>

<sup>53</sup>[http://fundamentals-of-bpm.org/wp-content/uploads/2013/11/ch7\\_CreditAppSimulation.bpmn\\_.zip](http://fundamentals-of-bpm.org/wp-content/uploads/2013/11/ch7_CreditAppSimulation.bpmn_.zip)

Dumas et al. 2013, S. 104)

Die Datei *ch7CreditAppSimulation.bpmn* befindet sich zudem auf der CD unter *resources*.

Hierzu wird über die Webseite ein BPMN Prozessmodell mit der Dateinamenserweiterung *.bpmn* oder *.vsdx*<sup>54</sup> hochgeladen. Dies zeigt die Abbildung 4.37.

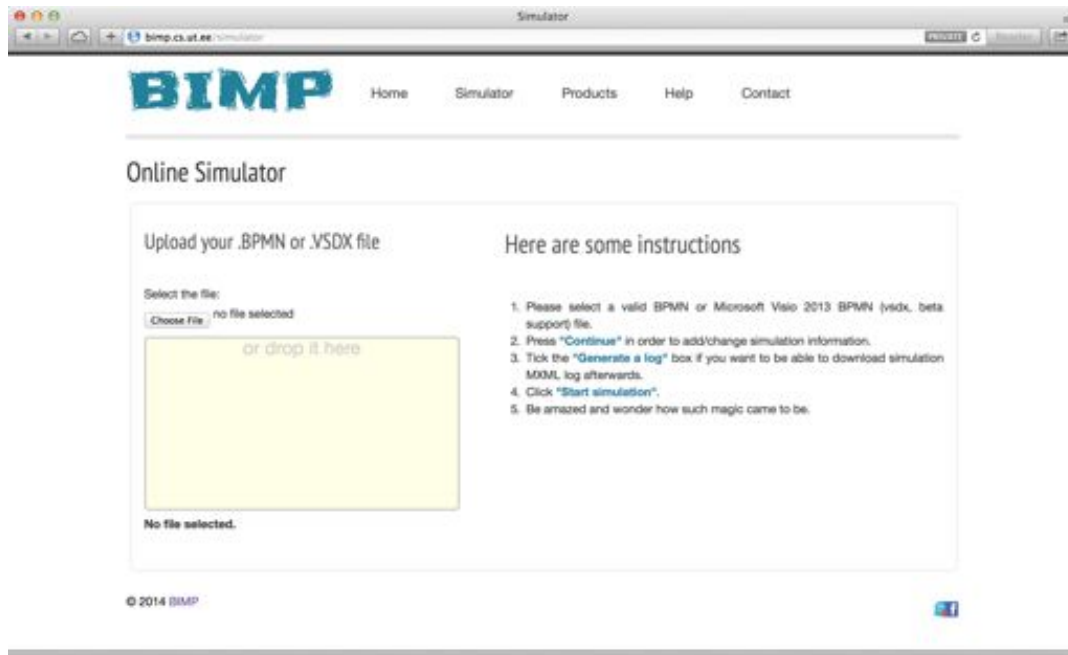


Abbildung 4.37: BIMP Simulator Prozessmodell hochladen

Nachdem die Datei hochgeladen und verarbeitet wurde, ist die Eingabe der Szenario-Parameter erforderlich. Wurde ein Szenario für dieses Prozessmodell bereits in BIMP definiert und in der Datei gespeichert, so werden die Simulationsparameter automatisch übernommen. Das ist auch bei dem gewählten Beispiel der Fall. Die Eingabe der Parameter gliedert sich in die fünf Kategorien: Process simulation specification, Resources, Timetable/Work schedule, Tasks und Gateway. Die Abbildung 4.38.

Die Gruppe Process simulation specification definiert allgemeine Parameter. Darunter fallen die Arravil rate, welche das Intervall zwischen zwei Prozessinstanzen definiert. Die Number of instances definiert, wie viele Prozessinstanzen während der Simulation erzeugt werden. Zudem kann die Startzeit und Währung der Simulation bestimmt werden.

Die Gruppe Resources definiert welche Personen oder Systeme an dem Prozess beteiligt sind. Zu hinzugefügten Ressourcen kann der Name, Nummer, Kosten pro Stunde und deren Arbeitszeit definiert werden.

<sup>54</sup>XML basierte Microsoft Visio Datei

Die Gruppe Timetable / Work schedule definiert die Arbeitszeiten, welche den Ressourcen zugewiesen werden. Neben dem Start- und Endtag, kann die Start- und Endzeit der Tage bestimmt werden. Werden neben der Standarddefinition Zeiten definiert, so tragen diese einen Namen.

Die Gruppe Tasks definiert für jeden im Prozessmodell gefundenen Task die Resource, die entstehenden Fixkosten und deren Ausführungszeit.

Die Gruppe Gateways definiert alle im Prozessmodell gefundenen Gateways. Bei zum Beispiel XOR Gateways wird die prozentuale Wahrscheinlichkeit für die einzelnen Wege angegeben.

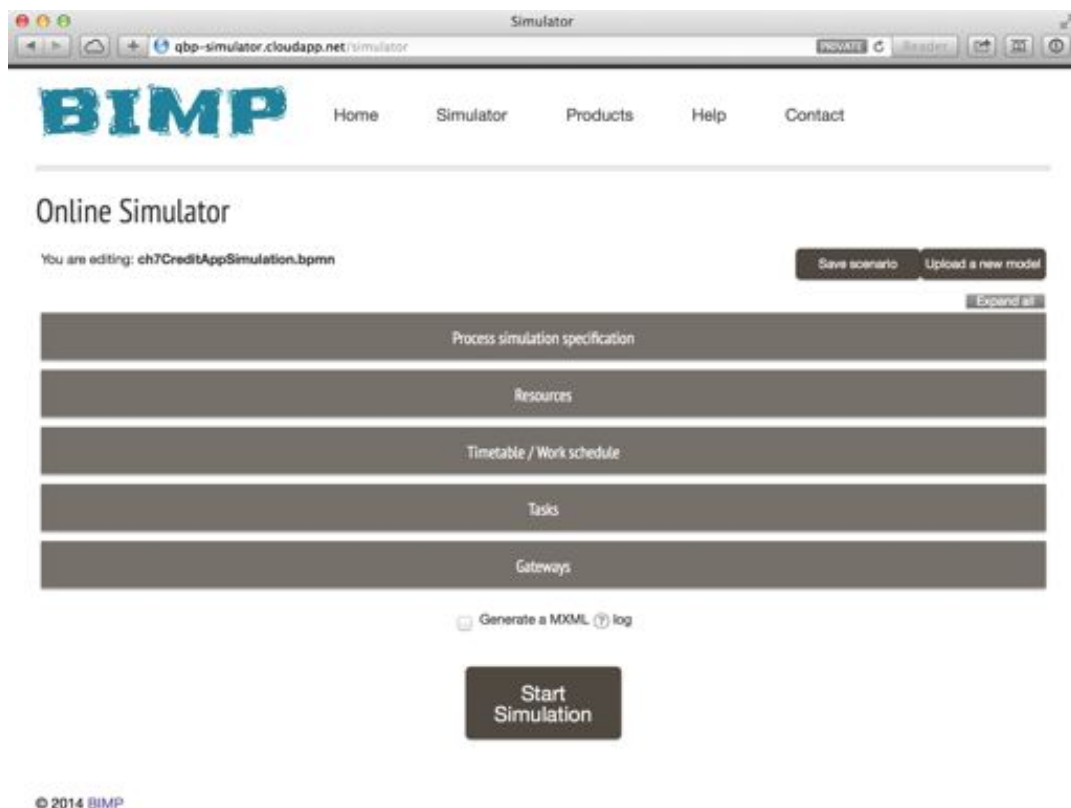


Abbildung 4.38: BIMP Simulator Kategorien

Die Option *MXML log* speichert die Event logs in einer XML Datei. Diese kann mit anderen Processmining<sup>55</sup> Werkzeugen ausgewertet werden. Die Eingabe des Beispiel-Prozessmodelles ist in Abbildung 4.39 zu sehen.

Nachdem die Simulation gestartet und erfolgreich durchlaufen wurde, werden die Ergebnisse grafisch dargestellt. Sie umfassen allgemeine Informationen wie zum Beispiel die Gesamtkosten und die Laufzeit, sowie Diagramme zu den einzelnen Prozesszeiten, Wartezeiten, Prozesskosten und die prozentuale Resour-

<sup>55</sup><http://www.processmining.org>

Abbildung 4.39: BIMP Simulator Eingaben

cennutzung. Die weiteren zwei Gruppen sind Prozessinstanz-Kosten und Wartezeiten, sowie die Task-Kosten und Wartezeiten. Die Abbildung 4.40 zeigt die Ausgabe.

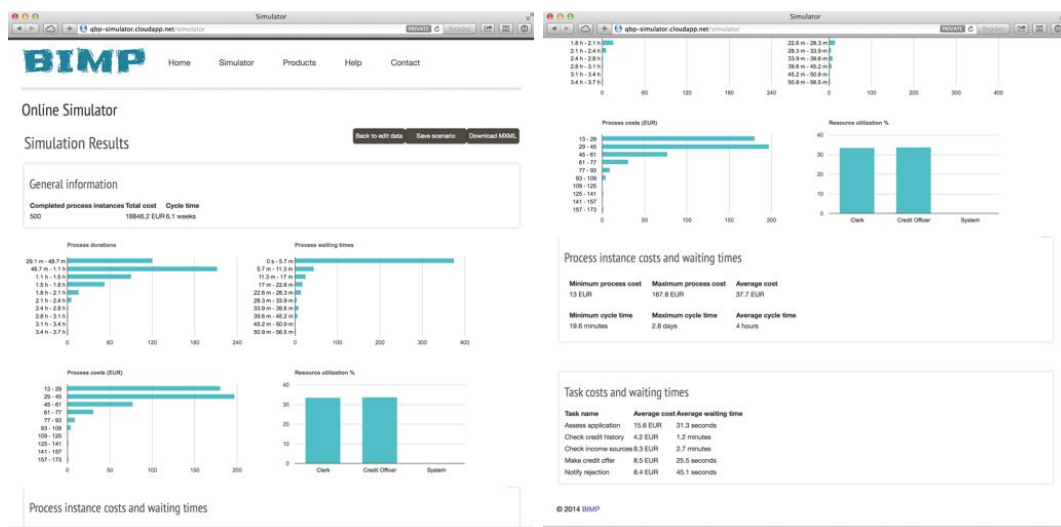


Abbildung 4.40: BIMP Simulator Ergebnisse

Das Definieren der Szenario-Parameter konnte aufgrund der späteren Veröffentlichung der BPSim Spezifikation nicht in diesem Format erfolgen. Es wurde hierfür das separate qbp Schema definiert. Die Definition der Parameter erfolgt im *document* Element des Start-Events. Die Verknüpfung mit den IDs der Prozessmodellelemente erfolgt im *qbp:processSimulationInfo* Element. Relevante Auszüge der Datei sind in dem Quellcode 4.16 zu sehen.

```

1 ... xmlns:qbp="http://www.qbp-simulator.com/Schema201212" ...
2 <startEvent id="sid-47...B6" name="Credit application received">

```



```

3      <extensionElements><signavio:signavioMetaData metaKey="bgcolor" metaValue=
        "#ffffff"/></extensionElements>
4      <outgoing>sid-E4...72</outgoing>
5      <documentation id="96...3c">{"arrivalRateDistribution":{"type":"exponential"
        ,"mean":1800,"value":0,"stdev":0,"min":0,"max":0,"timeUnit":"minutes"},
        "instances":"5000","resources":{"Clerk":{"name":"Clerk","costPerHour":"25"
        ,"amount":"3"},"CreditOfficer":{"name":"Credit Officer","costPerHour":"50"
        ,"amount":"3"},"System":{"name":"System","costPerHour":"0","amount":"1"
        }}, "startAt":"2012-08-03 17:27:09","timetable":{"Clerk":{"Mon-Fri":"09
        :00:00-17:00:00"},"CreditOfficer":{"Mon-Fri":"09:00:00-17:00:00"}}, "
        currency":"EUR"}</documentation></startEvent> ...
6  <qbp:processSimulationInfo xmlns:qbp="http://www.qbp-simulator.com/
        Schema201212" currency="EUR" processInstances="500" startDateTime="
        2013-07-03T20:11:36.000+00:00" id="qbp_81...18"><
        qbp:arrivalRateDistribution type="EXPONENTIAL" mean="0" arg1="1800" arg2="0"
        " id="qbp_da...9f"><qbp:timeUnit id="qbp_d9...88">minutes</qbp:timeUnit></
        qbp:arrivalRateDistribution><qbp:resources id="qbp_70...3f"><qbp:resource
        id="qbp_17...02" name="Clerk" costPerHour="25" totalAmount="3" timetableId=
        "DEFAULT_TIMETABLE"/> ...
7  <qbp:element fixedCost="0" elementId="sid-62...2D" id="qbp_d2...76"><
        qbp:durationDistribution type="NORMAL" mean="600" arg1="120" arg2="0" id="
        qbp_d8...a4"><qbp:timeUnit id="qbp_7b...4e">minutes</qbp:timeUnit></
        qbp:durationDistribution><qbp:resourceIds id="qbp_08...0d"><qbp:resourceId
        id="qbp_b1...11">qbp_4a...c5</qbp:resourceId></qbp:resourceIds></
        qbp:element> ...
8  <qbp:sequenceFlow elementId="sid-78...FB" executionProbability="0.8" id="
        qbp_7c...e5"/></qbp:sequenceFlows></qbp:processSimulationInfo></definitions
        >

```

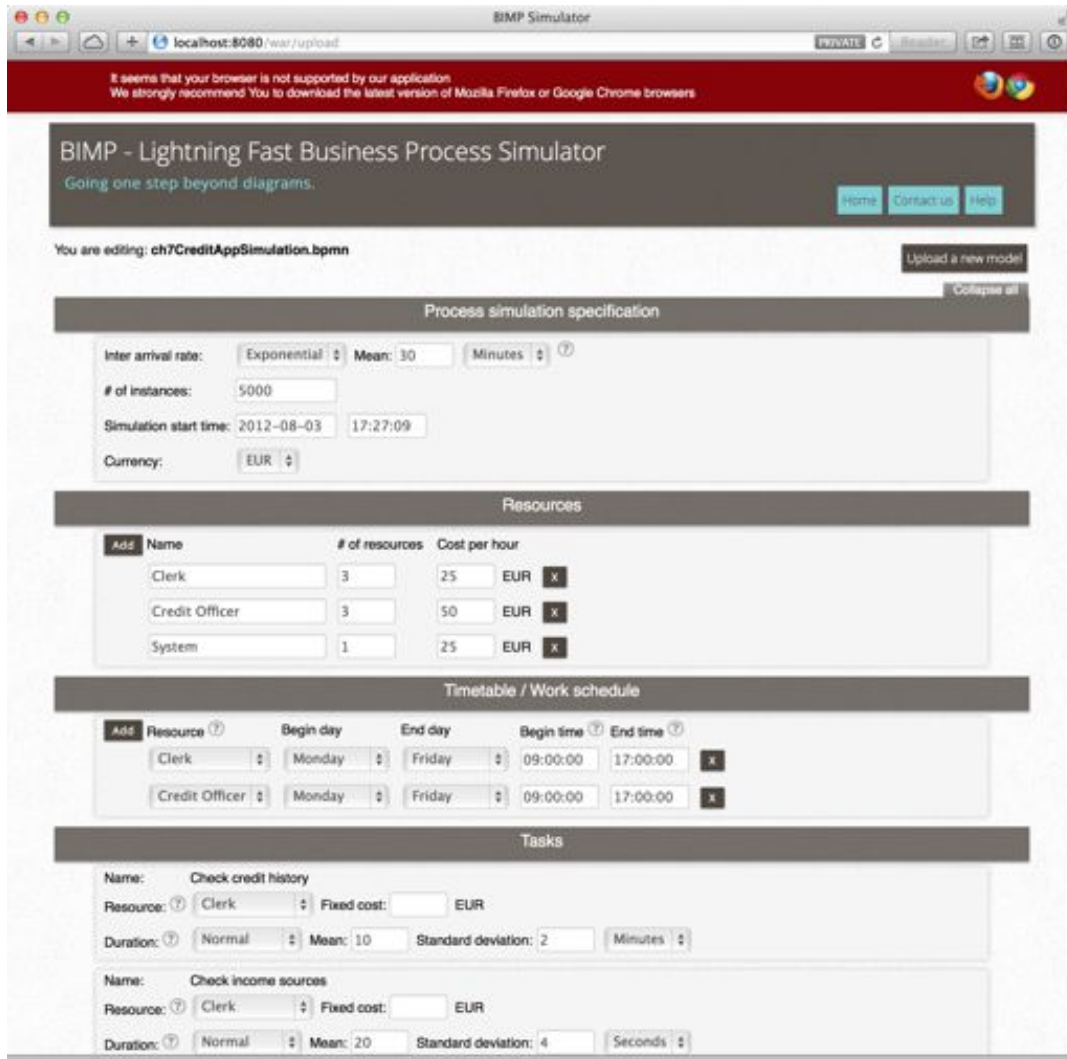
Quellcode 4.16: BIMP Simulator Prozessmodell-Datei

Die Aufgabe, die Simulation mit BIMP auf dem eigenen Computer laufen zu lassen, schlug aus mehreren Gründen leider fehl. Zum ersten fehlt in dem aktuellen Quellcode-Repository die Datei *bimplib.jar*. Wodurch sich das Projekt nicht compilieren ließ. Auf der Suche nach dieser .jar Datei wurde das Projekt *ut-bpsimulator*<sup>56</sup> gefunden. Dies ist eine BPMN 2.0 Simulations-Engine aus dem Jahr 2011. Damit funktionierte das Projekt nicht. Aus diesem Grund wurde in der History des BIMP-Repository nach der entsprechenden Datei gesucht und gefunden<sup>57</sup>. Nachdem die Datei eingebunden wurde, ließ sich das Projekt compilieren. Durch das Ant-Script wird zwar der mitgelieferte Apache Tomcat gestartet, aber nicht die Anwendung deployed. Das deployment wurde manuell durchgeführt. Nachdem die Anwendung startete, konnte das HTML Formular nicht benutzt werden, da dessen Verlinkung fest auf das Wurzelverzeichnis des Webserver gelegt war. Nach dessen Behebung, fehlte zudem in dem Formular die Action, an die die Daten gesendet werden und die Simulation gestartet wird. Der Aufruf wurde auf das Servlet *simulate* gelegt. Die Daten wurden anscheinend nicht hochgeladen und die Simulation nicht gestartet. Der

<sup>56</sup><https://code.google.com/p/ut-bpsimulator/>

<sup>57</sup><https://code.google.com/p/bimp-simulator/source/detail?r=39a64340437f05b260d8ad54c9c108f1a44ca9d4>

letzter Schritt, die Eingabe bzw. das automatische Auslesen der Eingabedatei, ist in der Abbildung 4.41 zu sehen. Aufgrund fehlender Dokumentation wurde an dieser Stelle die Forschung und Fehlerbehebung abgebrochen. Der aktuelle Stand des Quellcode ist auf der CD und unter <https://github.com/search> zu finden.



**BIMP - Lightning Fast Business Process Simulator**  
Going one step beyond diagrams.

You are editing: **ch7CreditAppSimulation.bpmn**

**Process simulation specification**

Inter arrival rate: Exponential Mean: 30 Minutes  
 # of instances: 5000  
 Simulation start time: 2012-08-03 17:27:09  
 Currency: EUR

**Resources**

Add	Name	# of resources	Cost per hour
	Clerk	3	25 EUR
	Credit Officer	3	50 EUR
	System	1	25 EUR

**Timetable / Work schedule**

Add	Resource	Begin day	End day	Begin time	End time
	Clerk	Monday	Friday	09:00:00	17:00:00
	Credit Officer	Monday	Friday	09:00:00	17:00:00

**Tasks**

Name: Check credit history  
 Resource: Clerk Fixed cost: EUR  
 Duration: Normal Mean: 10 Standard deviation: 2 Minutes

Name: Check income sources  
 Resource: Clerk Fixed cost: EUR  
 Duration: Normal Mean: 20 Standard deviation: 4 Seconds

Abbildung 4.41: BIMP Simulator auf eigenem Computer



## 5 Fazit

Zusammenfassend kann festgestellt werden, dass es möglich ist, Geschäftsprozesssimulationsmodelle aus BPMN Modellierungswerkzeugen zu erzeugen. Die BPMN 2.0 Spezifikation sieht vor, solche Prozessmodellerweiterungen über das *extensionElements* Element zu integrieren. Die WfMC hat mit BPSim eine allgemeine Spezifikation von Simulationsein- und Simulationsausgabeparametern definiert. In der Spezifikation werden die Simulationsparameter über die Element IDs mit den Prozessmodellelementen referenziert. Bei den fünf betrachteten Modellierungswerkzeugen sind folgende Schlussfolgerungen möglich.

Mit den Modellierungswerkzeugen jBPM Developer Tools bzw. Eclipse BPMN2 Modeler Plugin und dem web-basierten jBPM Process Designer können Geschäftsprozesssimulationsmodelle generiert werden. Die Definition der Simulationsparameter erfolgt ausschließlich im BPSim Standard. Mit dem jBPM Process Designer können zudem die Prozessmodelle simuliert werden. Zudem wurden zwei Java Projekte umgesetzt. In dem ersten wurde über die Eclipse BPMN2 Modeler API ein BPMN 2.0 Prozessmodelle gelesen. Das zweite Projekt startet über die jBPM Simulation API eigenständig eine Geschäftsprozesssimulation.

In dem Activiti Projekt existiert bis dato keine Unterstützung von Simulationsfunktionen. Jedoch lässt sich der Eclipse-basierte Activiti Modeler über eine definierte Plugin-Schnittstelle erweitern. Über die Plugin-Schnittstelle konnten zusätzliche Elemente mit Parametern definiert, eigene Validierungen und Exportfunktionalitäten eingebunden werden. Somit können selbst definierte Attribute in die BPMN 2.0 Prozessmodelle integriert werden. Zudem wurde das unveröffentlichte Activiti Crystalball Projekt betrachtet, welches Simulationsfunktionen zur Verfügung stellt. Die Simulationsfunktionen sind ausschließlich über Java Quellcode verfügbar. Parameter werden in einem eigenen Namespace und einer Konfigurationsdatei definiert. Simulationsdaten bzw. Szenarien stammen aus der Process Engine Datenbank und ermöglichen das Wiederholen von realen Fällen. Voraussichtlich wird das Projekt in der nächsten Activiti Version 5.16 beinhaltet sein.

Die Camunda Plattform stellt aktuell ebenso wie das Activiti Projekt keine Simulationsfunktionen zur Verfügung. Eine solche Funktionalität ist bis dato

nicht in konkreter Planung. Es ist jedoch zu vermuten, dass eine solche Funktionalität integriert wird. Grund zu dieser Annahme ist die Masterarbeit des Geschäftsführers Bernd Rücker über Geschäftsprozesssimulation mit jBPM und DESMO-J. Die Camunda Modeler Plugin-Schnittstelle ist identisch zu der vom Activiti Projekt und stellt damit die gleichen Erweiterungsmöglichkeiten bereit. Das Hinzufügen eines eigenen Elementes wurde hier ebenfalls demonstriert. Aufgrund des Plattformgedankens von Camunda existieren verschiedene Bibliotheken, die die Verarbeitung von BPMN Prozessmodellen ermöglichen und vereinfachen. In einem Projekt sollte über eine eigene Java Klasse ein eigenes BPMN Element definiert werden und der Namespace erweitert werden. Aufgrund unzureichender Dokumentation in diesem Gebiet, konnte das Schreiben des eigenen Elementes nicht realisiert werden. Eine Nutzung der Camunda Projekte würde sich bei der Entwicklung einer eigenen Simulationslösung anbieten.

Das Yaoqiang Version 2.2 ist ein Java-basiertes Modellierungswerkzeug. Auf der Webseite steht, dass es eine Simulationsfunktion hat. Diese konnte jedoch aufgrund fehlender Dokumentation nicht ausgeführt werden. Zudem wird angezweifelt, dass die Funktion ausgeführt werden kann, da sich kaum Parameter im Prozessmodell definieren lassen. Über eine eigene Plugin-Schnittstelle können Prozess Engines wie zum Beispiel von Activiti angebunden werden.

Das web-basierte Werkzeug BIMP verfolgt, im Gegensatz zu den bisher genannten Modellierungswerkzeugen, einen anderen Ansatz. In das Werkzeug wird ein bestehendes Prozessmodell geladen, welches dann analysiert wird und die Eingabe der erforderlichen Simulationsparameter ermöglicht. Es wird ein eigener Namespace definiert, in dem die Simulationsparameter gespeichert werden. Zudem werden Simulationsinformationen in dem Dokumentationselement des Start-Events, gespeichert. Im Anschluss wird das Prozessmodell in der BIMP-unabhängigen Engine *ut-bpsimulator* simuliert. In der BIMP-Weboberfläche werden die Ergebnisse der Simulation angezeigt.

Zusammenfassend betrachtet, können bereits Geschäftsprozesssimulationen auf BPSim Basis mit den jBPM Developer Tools generiert und mit dem web-basierten Process Designer sogar simuliert werden. Die Entwicklung einer leichtgewichtigen, eigenständigen, BPSim- und Web- bzw. Webservice-basierten Simulationslösung aus BIMP, der jBPM-Simulations-Engine und ggf. Camunda-Komponenten sehe ich als eine moderne, skalierbare und attraktive Lösung.

Sämtliche prototypischen Entwicklungen sind auf der beigelegten CD und unter <https://github.com/search> zu finden.

Das Studienfach Wirtschaftsinformatik fasziniert mich persönlich besonders wegen seiner Mischung aus Wirtschaft und Informatik. Es wird nicht nur ein Ge-

biet, sondern das Gesamtbild vom Management bis hin zur technischen Umsetzung betrachtet. Das in diesem Bachelorstudium erlangte methodische und fachliche Wissen konnte ich in diese Arbeit einbringen. Ich bin zudem weiter motiviert, mein Wissen im Masterstudium zu erweitern und zu vertiefen.

## 5.1 Bewertung

Es gibt in dieser Arbeit einige kritisch zu betrachtende Punkte. Zum einen ist die seltene Verwendung der BPSim Spezifikation zu nennen. Auch wenn diese erst seit 2013 existiert, wird sie nicht in neuen Projekten wie Activiti Crystalball eingesetzt. Durch die unterschiedliche Definition der Simulationsinformationen ist eine Interoperabilität zwischen den Modellierungswerkzeugen nicht gegeben. Weiterhin ist bei dem Projekt Activiti bzw. Crystalball (Kapitel 4.3.3) die Plattformunabhängigkeit nicht gegeben.

Ein modellierungswerkzeugunabhängiges Problem sind Schnittstellen. Durch Veränderungen der Plugin-Schnittstellen oder APIs, funktioniert eine Eigenentwicklung ohne Anpassungen nicht mehr. Ausgenommen sind hier APIs die ein Projekt selbst mitbringt und verwendet.

Ein weiterer kritischer Punkt ist die Dokumentation der open source Projekte. Ist diese nicht gegeben oder unzureichend gestaltet, gestaltet sich eine Umsetzung oder Fehlerbehebung schwierig. In dieser Arbeit ist dies in den Kapiteln 4.4.2, 4.5.1 und 4.5.2 zu spüren gewesen. Alternative Lösungswege über die Community oder den Quellcodeanalyse sind im Vergleich meist erheblich aufwendiger.

Des weiteren ist der Fokus auf die Verwendung der Eclipse IDE bedenklich. Das Werkzeug richtet sich vorwiegend an Entwickler. Dadurch wird der Zugang für beispielsweise Prozessanalysten erschwert. Ein web-basierter Ansatz wie ihn beispielsweise BIMP verfolgt, ermöglicht auch nicht technisch versierten Nutzern einen einfachen Zugang zu Simulationsfunktionen.

Eine Zusammenfassende Bewertung ist der Tabelle 5.1 zu entnehmen.

Projekt	Plattform	Schnittstelle	Modellierung	Simulationsparameter	Simulationsfunktion	Import	Export	Projekktivität	Dokumentation
jBPM Developer Tools	Eclipse	o	+	+	-	o	+	+	+
Activiti Modeler	Eclipse	+	+	-	-	-	+	+	+
Camunda Modeler	Eclipse	+	+	-	-	o	+	+	+
Yaoqiang	Java	o	o	-	o	-	+	o	-
BIMP	Web	o	-	+	+	+	+	-	-

+ gut | o mittel | - schlecht

Tabelle 5.1: Übersicht der Modellierungswerkzeugbewertung

## 5.2 Offene Punkte

Allgemein muss weiter an der Thematik gearbeitet werden. Erste Anzeichen für Weiterentwicklungen in den open source Werkzeugen wurden bereits dargestellt. Beispielsweise das Activiti Crystalball Projekt ist nach Veröffentlichung nochmal zu überprüfen.

Wie im Kapitel 4.4.2 bereits erwähnt, ist die prototypische Entwicklung an der Camunda Modeler API nicht lauffähig. Die genaue Ursache der Fehlermeldung beim Anlegen eines eigenen Elementes muss noch erforscht werden. Ebenso erfordert die Bereitstellung einer eigenen BIMP-Installation weiteren Forschungs- und ggf. Entwicklungsaufwand.

Diese Arbeit soll mit der Forschung rund um die Thematik der Geschäftsprozesssimulationsmodelle und prototypischer Entwicklungen im Kapitel 4 Grundlage für weitere Arbeiten bilden.

# Literaturverzeichnis

Activiti, T. (2014), ‘Activiti 5.15 user guide’. Stand 22.06.2014.

**URL:** <http://activiti.org/userguide/index.html#eclipseDesignerExtending>

Activiti, T. (n.d.), ‘Activiti components’. Stand 22.06.2014.

**URL:** <http://activiti.org/components.html>

Allweyer, T. (2009), ‘Bpmn 2.0 - business process model and notation: Einführung in den standard für die geschäftsprozessmodellierung’, Books on Demand GmbH, Norderstedt. Auflage 2,.

Baeyens, T. (2010), ‘Process developments: alfresco creates activiti’. Stand 2014-06-22.

**URL:** <http://processdevelopments.blogspot.de/2010/05/alfresco-creates-activiti.html>

Bartonitz, D. M. (2009), ‘Historische entwicklung wichtiger standards im business process management’. Stand 02.06.2014.

**URL:** <https://goo.gl/aohdut>

Blenta (n.d.), ‘Yaoqiang bpmn editor’. Stand 30.06.2014.

**URL:** <http://sourceforge.net/projects/bpmn/>

Bundesverwaltungsamt, P. F. (n.d.), ‘Organisationshandbuch’, Bundesministerium des Innern, Bundesverwaltungsamt. Stand 02.06.2014.

**URL:** <https://goo.gl/an3RjW>

Camunda, C. (2014), ‘User guide | camunda bpm docs’, Community Camunda. Stand 22.06.2014.

**URL:** <http://docs.camunda.org/7.1/guides/user-guide/>

Dudenverlag (2013a), ‘Duden | prozess | rechtschreibung, bedeutung, definition, synonyme, herkunft’, Dudenverlag. Stand: 03.06.2014.

**URL:** <http://www.duden.de/node/659973/revisions/1195613/view>

Dudenverlag (2013b), ‘Duden | simulieren | rechtschreibung, bedeutung, definition, synonyme, herkunft’, Dudenverlag. Stand: 12.06.2014.

**URL:** <http://www.duden.de/rechtschreibung/simulieren>

- Dumas, M., Rosa, M. L., Mendling, J. & Reijers, H. A. (2013), 'Fundamentals of business process management', Springer. Heidelberg.
- Dumas, P. M. (2014), 'Quantitative process analysis 2', University of Tartu. Stand 02.06.2014.  
**URL:** <https://courses.cs.ut.ee/2014/bpm/uploads/Main/2014ITlecture5.ppt>
- Freund, J. & Rücker, B. (2012), 'Praxishandbuch bpmn 2.0', Carl Hanser Verlag München Wien. Auflage 3.
- Gadatsch, P. D. A. (2012), 'Grundkurs geschäftsprozess-management', Springer Vieweg. 7. Auflage.
- Gehring, H. & Gadatsch, A. (1999), 'Ein rahmenkonzept für die modellierung von geschäftsprozessen und workflows', Fachbereich Wirtschaftswissenschaft der FernUniversität Hagen. Fachbereichsbericht Nr. 274.
- Grofcik, M. (2014a), 'activiti-crystalball by gro-mar'. Stand 22.06.2014.  
**URL:** <http://gro-mar.github.io/activiti-crystalball/>
- Grofcik, M. (2014b), 'Getting-started'. Stand 30.06.2014.  
**URL:** <https://github.com/gro-mar/activiti-crystalball/wiki/Getting-started>
- Humble, C. (2013), 'Camunda forks alfresco activiti', InfoQ. Stand 22.06.2014.  
**URL:** <http://www.infoq.com/news/2013/03/Camunda-Forks-Activiti>
- jBPM team, T. J. (2014a), 'Chapter 1. overview'. Stand 24.05.2014.  
**URL:** <http://docs.jboss.org/jbpm/v6.0.1/userguide/jBPMSOverview.html>
- jBPM team, T. J. (2014b), 'Chapter 12. designer'. Stand 22.06.2014.  
**URL:** <https://docs.jboss.org/jbpm/v6.0.1/userguide/chap-designer.html>
- jBPM team, T. J. (2014c), 'Chapter 18. jbpm eclipse plugin'. Stand 22.06.2014.  
**URL:** <http://docs.jboss.org/jbpm/v6.0.1/userguide/jBPMEclipseJBPM.html>
- Meyer, D. (2013), 'camunda forks activiti and launches camunda bpm', Community Camunda. Stand 22.06.2014.  
**URL:** <http://blog.camunda.org/2013/03/camunda-forks-activiti-and-launches.html>
- Meyer, D. (2014a), 'camunda bpm - camunda jira', Camunda. Stand 30.06.2014.  
**URL:** <https://goo.gl/b7Ubla>

Meyer, D. (2014b), 'Simulation feature–google groups'. Stand 22.06.2014.

**URL:** <https://goo.gl/fGuu0n>

Müller, P. D. C. (2013), 'Epc-simulator spezifikation', TH-Wildau. Stand 22.04.2014.

**URL:** [http://www.tfh-wildau.de/cmuedler/EpcSimulator/EpcSimulator/Specification/EpcSimulator\\_Spec\\_20130124\\_final.pdf](http://www.tfh-wildau.de/cmuedler/EpcSimulator/EpcSimulator/Specification/EpcSimulator_Spec_20130124_final.pdf)

Object Management Group, I. & andere (2014), 'Business process model and notation (bpmn)', Object Management Group, Inc. Stand 23.04.2014.

**URL:** <http://www.omg.org/spec/BPMN/2.0.2/PDF>

Rademakers, T. (2012), 'Activiti in action', Manning Publications Co. United States of America.

Statista (2014), 'Geschäftsziele - mittelstand | umfrage 2011', IBM. Stand: 12.06.2014.

**URL:** <http://de.statista.com/statistik/daten/studie/182660/umfrage/wichtigste-geschaeftsziele-deutscher-unternehmen-im-mittelstand/>

WfMC (2012), 'Business process simulation specification', WfMC. Stand 23.05.2014.

**URL:** <http://www.bpsim.org/specifications/1.0/WFMC-BPSWG-2012-01.pdf>





# A Anhang

## A.1 Inhalt der CD

Die CD beinhaltet folgende Punkte.

- PDF Version dieser Bachelorarbeit
- Quellcode der prototypischen Entwicklungen (Verzeichnis *sourcecode*)
- Ressourcen, Prozessmodelle und Dokumentationen (Verzeichnis *resources*)



## **Selbstständigkeitserklärung**

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Berlin, 14.07.2014  
Ort, Datum

\_\_\_\_\_  
Unterschrift