

Institut für Rechnergestützte Ingenieursysteme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit Nr. 300

Analyse verschiedener BPMN Extensions

Patrick Karner

Studiengang: Softwaretechnik
Prüfer/in: Prof. Dr. Dieter Roller
Betreuer/in: Dipl.-Inf. Felix Baumann

Beginn am: 14. Dezember 2015

Beendet am: 14. Juni 2016

CR-Nummer: H.3.3, H.3.5, H.2

Kurzfassung

BPMN 2.0 ist eine grafische Spezifikationssprache, mit der es möglich ist, Geschäftsprozesse zu modellieren. In der Version 2.0 wurde ein Erweiterbarkeitsmechanismus hinzugefügt, mit dem es möglich ist, Erweiterungen zum Core BPMN Modell hinzuzufügen. Solche Erweiterungen werden dann veröffentlicht, sind aber dennoch schwer aufzufinden. In dieser Bachelorarbeit geht es darum, eine Klassifizierung zu erstellen, mit der man die schon vorhandenen Erweiterungen voneinander unterscheiden, als auch Gemeinsamkeiten herausfinden kann. Damit diese Erweiterungen auch von verschiedenen BPMN Nutzern gefunden werden können, soll noch eine grafische Oberfläche in Form einer Webseite erstellt werden. Auf dieser Webseite können dann bekannte oder neue Erweiterungen hinzugefügt und bearbeitet werden. Durch eine integrierte Suchfunktion kann spezifisch nach einer Erweiterung gesucht werden.

Abstract

BPMN 2.0 is a graphical specification language that is used to model business processes. The current version 2.0 provides the possibility to extend the existing BPMN core model by adding an extension. This is enabled through the extensibility mechanism invented in 2.0. Many extension are published, so it is hard to get an overview in the internet. This bachelor thesis focuses on classifying BPMN extensions, in order to differentiate them or to find out several similarities. The information about the extensions are summarized and published on a website, additionally designed in this work. Users are able to add new or already existing BPMN extensions, as well as to search for them on this site.

Inhaltsverzeichnis

1	Einleitung	9
2	Analyse der BPMN Extensions	11
2.1	Vorgehen	11
2.2	BPMN Extension Auswahl	12
2.3	Klassifikation	12
2.4	Analyse	17
3	Die grafische Oberfläche	59
3.1	Einleitung	59
3.2	Verwendete Technologien	59
3.3	Browser	60
3.4	Anforderungen an die Webseite	60
3.5	Personenkreis	62
3.6	Grafische Oberfläche	62
3.7	UseCase	72
4	Datenbank und Backend	73
4.1	Datenbanksystem	73
4.2	JavaScript	78
4.3	Erstellung der Kategorien der Suchseite	80
4.4	Suchalgorithmus	82
4.5	Erstellen und Bearbeiten von Extensions	87
5	Zusammenfassung und Ausblick	93
	Literatur	95

Abbildungsverzeichnis

3.1	Suchseite	63
3.2	Hinzufügen einer Extension Oberfläche	65
3.3	Hinzufügen einer Extension Universität und Unternehmen	66
3.4	Hinzufügen einer Extension Class & Domain	67
3.5	Hinzufügen einer Extension Symbol	68
3.6	Hinzufügen einer Extension Example	69
3.7	Hinzufügen einer Extension Difficulties	70
3.8	Hinzufügen einer Extension Keywords	71
3.9	Use Case der Webseite	72
4.1	EER Modell der Datenbank	74

Tabellenverzeichnis

2.1	Klassifikations Tabelle	13
-----	-----------------------------------	----

Listings

4.1	JavaScript Controller Struktur	80
4.2	universityCategory.php	81
4.3	Suchanfrage	85
4.4	Titel Suche SQL Query	86
4.5	Keyword SQL Query	86
4.6	Generiere Suchanfrage für Keyword	86
4.7	Hinzufügen einer Klasse mit Subklasse in die Datenbank	89
4.8	Generierung des HTML Code von negativen Schwierigkeiten List aus insert-Database.php	91

4.9 Einfügen des generierten HTML Code in die input Variable 91

Kapitel 1

Einleitung

Das Business Process Model and Notation, kurz BPMN hat sich zum Heutigen de facto Standard im entwickeln von Business Process Models, also Geschäftsprozessen entwickelt. In der aktuellen Version 2.0 von BPMN, die von der Object Management Group (OMG) entwickelt wurde, wurden einige neue Features integriert. Dazu gehört definitiv die Möglichkeit das Standard BPMN, das auch Core Modell genannt wird, zu erweitern. Durch diese Erweiterbarkeit ist es möglich, die von BPMN verabschiedete Notationen domänenspezifisch zu erweitern. Dies ist im Hinblick auf andere Business Process Notationssprachen ein großer Vorteil, denn dadurch können Probleme in einem Geschäftsprozess detaillierter, informativer als auch erst durch das hinzufügen dieser neuen Elemente oder Ausführungen einfacher dargestellt oder erst ermöglicht werden. Es existieren zu diesem Zeitpunkt viele Erweiterungen die nach meinen Recherchen überwiegend an Universitäten entwickelt werden. Ein Problem das alle diese Extensions jedoch teilen ist, das man diese nur durch mühseliges Suchen im Internet auffinden kann. Literatur die alle bekannten Extensions, oder zumindest einen groß Teil dieser beinhalten, existiert zu dem Zeitpunkt dieser Bachelorarbeit genauso wenig wie eine Webseite die alle Extensions enthält. Ein weiteres Problem ist ebenfalls die Einordnung dieser Extensions. Da diese meist als wissenschaftliche Ausarbeitungen an z.B. Konferenzen veröffentlicht werden, kann es auch mal vorkommen, dass diese nur ein Vorschlag zu einer Extension sind ohne jegliche konkrete Ansätze. Somit sind manche Extensions in einer Entwicklungsphase und noch nicht für eine Verwendung brauchbar.

Um dieser Problematik ein Ende zu setzen, ist das Ziel in dieser Bachelorarbeit schon bestehende Extensions zu suchen und diese dann zu analysieren und kategorisieren. Dazu soll noch zusätzlich eine Webseite erstellt werden, in der es möglich sein soll Extensions hinzuzufügen und diese dann anschließend über eine Suchmaske auffindbar zu machen.

Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Analyse der BPMN Extensions: Hier werden die gefunden Extensions erklärt und analysiert. Außerdem wird über das Suchvorgehen und die letztendliche Auswahl der Extensions in dieser Arbeit diskutiert.

Kapitel 3 – Die grafische Oberfläche Hier wird die grafische Oberfläche vorgestellt und die Anforderungen was diese Webseite leisten soll spezifiziert.

Kapitel 4 – Datenbank und Backend Im letzten Kapitel wird näher auf die verwendete Technik eingegangen die für die Erstellung der Webseite notwendig war. Hier wird z.B. die verwendete Datenbank sowie der erstellte Suchalgorithmus vorgestellt.

Kapitel 2

Analyse der BPMN Extensions

2.1 Vorgehen

Bevor erklärt wird, wie die gefundenen BPMN Extensions analysiert und kategorisiert werden, wird davor erst einmal das Vorgehen der Suche nach diesen Extensions verdeutlicht und begründet. Das Hauptmedium der Suche war das Internet mit der Hilfe von wissenschaftlichen Suchmaschinen. Dabei war die erste Anlaufstelle die wissenschaftliche Suchmaschine *Google Scholar*¹ vom Google Unternehmen. Diese lieferte die besten und die meisten Suchtreffer im Vergleich zu anderen wissenschaftlichen Suchmaschinen, die in dieser Arbeit ebenfalls verwendet wurden, wie z.B. *CiteSeerX*² oder die Suchmaschine für wissenschaftliche Arbeiten von der Universität Bielefeld *BASE*³. Die Suche mit wissenschaftlichen Suchmaschinen ist verwendet worden, da BPMN Extensions überwiegend an Universitäten als Forschungsarbeit bearbeitet und dann online, wie z.B. auf der IEEE Webseite, zur Verfügung gestellt werden. Die verwendeten Suchbegriffe waren:

- ◇ BPMN Extension
- ◇ BPMN Enhancement
- ◇ BPMN₄
- ◇ BPMN Extend

Mit dem erst genannten Suchbegriff konnten die meisten erfolgreichen Treffer erzielt werden. Es sind die ersten 10 Seiten der Suchergebnisse betrachtet worden. Der Suchbegriff „BPMN₄“ erscheint etwas verwirrend. Dieser ist aber verwendet worden, da dieser auf Englisch auch die Bedeutung „BPMN For“, also auf Deutsch, „BPMN für“ besitzt. Dies bedeutet, nach der 4 steht meist das Thema das diese Erweiterung erweitert. Eine geringe Anzahl von Erweiterungen benutzt solch eine Schreibweise, wie z.B. 2.4.11. Neben der Suchmaschine Google Scholar, ist auch auf das Dokumenten Archiv der *IEEE Webseite*⁴ zugegriffen worden um weitere BPMN Extensions zu finden, die nicht durch Google Scholar auffindbar waren.

¹<https://scholar.google.de/>

²<http://citeseer.ist.psu.edu/>

³<https://www.base-search.net/>

⁴<http://ieeexplore.ieee.org/Xplore/home.jsp>

Durch das Literaturverzeichnis der bereits gefundenen Extensions, sind weitere Extensions gefunden worden. Es existieren zum Thema BPMN Extensions bis jetzt keine spezifische Literatur, die sich ausschließlich damit beschäftigt.

2.2 BPMN Extension Auswahl

Insgesamt wurden 39 BPMN Extensions gefunden. Davon wurden 30 analysiert und haben es somit in diese Arbeit geschafft. Es konnten nicht alle Erweiterungen analysiert werden, da zum einen die Zeit für die Analyse für alle Erweiterungen gefehlt hat. Zum Anderen decken manche Erweiterungen eine zu alte Version von BPMN ab, wie z.B. die Erweiterung „Visualisation of (distributed) process execution based on extended bpmn“ [21] die im Jahre 2003 publiziert wurde und sich nach der BPMN Version 0.9 richtet. Es wird außerdem darauf geachtet das die ausgewählten Erweiterungen unterschiedliche Themen- und Anwendungsbereiche abdecken. Das bedeutet, es werden Erweiterungen ausgewählt die z.B. den medizinischen Bereich[5], den Herstellungsprozess eines Produktes[33] oder die Sicherheit in Geschäftsprozessen[6] in BPMN erweitern um nur einige zu nennen. Ein weiteres wichtiges Auswahlkriterium für diese Arbeit ist deshalb, dass die hier aufgeführten Erweiterungen nachher Personen bei ihren Modellierungsproblemen helfen sollen und somit im Arbeitsleben verwendet werden können. Erweiterungen die schwer zu verstehen oder dessen Konzept schwer nachvollziehbar oder gar unvollständig ist, wie z.B. bei der Erweiterung „Extending BPMN 2.0 with Sensor and Smart Device Business Functions“ [12], werden ebenfalls aus der engeren Auswahl rausgeworfen.

2.3 Klassifikation

Um die verschiedenen Extensions im späteren Verlauf der Arbeit auf einer Webseite auffindbar, durchsuchbar und unterscheidbar zu machen, müssen die Extensions analysiert und klassifiziert werden. Die Klassifikation ist aus der Tabelle 2.1 zu entnehmen. In dieser werden sowohl die Metadaten, als auch weitere wichtige Informationen zur Erweiterung angegeben. Die Erklärung der verschiedenen Kategorien folgen in 2.3.1.

2.3.1 Klassifikationstabelle Erklärung

Autoren Beinhaltet die Namen aller beteiligten Autoren. Falls mehrere Autoren von unterschiedlichen Universitäten an einer Erweiterung gearbeitet haben, werden der Autor und die zugehörige Universität mit einer Nummer versehen.

Veröffentlichungsjahr Das Jahr, in dem die Arbeit veröffentlicht wurde.

Veröffentlichungsform Gibt an in welcher Form die Arbeit veröffentlicht wurde. Einige Formen die verwendet wurden sind,

Kategorie	Beschreibung
Autoren	Namen der beteiligten Autoren
Veröffentlichungsjahr	Jahreszahl
Veröffentlichungsform	Konferenz, Artikel, Buch
Universität/Unternehmen	Name der beteiligten Universitäten oder Unternehmen
Link	Link zur Arbeit
Veröffentlichung	Veröffentlichungsort
Anwendungsdomäne	Zum Beispiel General, Manufacturing domain, Business domain, Medicine domain, usw.
Klasse	Zum Beispiel Time Management, Management, Process Tailoring, visualization, Requirements, usw.
Anzahl neuer Symbole	Anzahl der neu hinzugefügten Symbole
Ausführbar	Ja oder Nein
Lernschwierigkeit	+ Leicht zu erlernen. - Schwer zu erlernen.
Unterstützte Modellierungsprogramme	Falls ein spezielles Modellierungsprogramm unterstützt wird, wie z.B. Activiti oder Camunda
BPMN 2.0 Regelkonform	Ja oder Nein
BPMN Version	1.2 oder 2.0, bzw. jede mögliche Version.
Keywords	verschiedene Keywords, die die Erweiterung genauer beschreiben sollen.

Tabelle 2.1: Klassifikations Tabelle

- ◇ Article
- ◇ Book
- ◇ Incollection
- ◇ Bachelorthesis
- ◇ Masterthesis
- ◇ Conference

Universität/Unternehmen Der Name der Universität oder Universitäten, die an dieser Arbeit geforscht und gearbeitet haben. Auch Unternehmen arbeiten an weiteren Erweiterungen und können hier natürlich ebenfalls hinzugefügt werden.

Link Der Link der zur Arbeit der Erweiterung führt. Dieser Link soll entweder direkt zur PDF Datei der Arbeit oder zum Verleger, wo man die Arbeit erwerben kann, führen.

Es steht hier z.B. der Name des Verlages, die Universität oder ein privater Server, auf dem die Arbeit sich befindet. Hinter dieser Angabe befindet sich ein Verweis ins Literaturverzeichnis. Dort ist der Link zur Erweiterung zu finden.

Veröffentlichung Gibt an wo die Erweiterung veröffentlicht wurde. Dies kann auf einer Konferenz, in einem Buch oder auch als Artikel im Internet oder in einer Fachzeitschrift sein.

Anwendungsdomäne Gibt an, in welchem Anwendungsbereich die Erweiterung angewendet werden kann. Wenn **General** als Antwort ausgewählt wird, bedeutet es, es gibt keine genaue Begrenzung des Anwendungsbereiches. Sprich, die Erweiterung kann in verschiedenen Anwendungsdomänen verwendet werden und ist nicht an eine spezifische Anwendungsdomäne gebunden. Spezifische Anwendungsdomänen sind z.B. die Manufacturing domain, Business domain oder Medicine domain. Es kann auch eine sogenannte Subdomäne verwendet werden. Diese soll die Domäne genauer beschreiben. Um ein kurzes Beispiel dafür zu geben, man kann die Domäne Medicine verwenden, um zu sagen, dass eine Extension im medizinischen Bereich verwendet werden soll. Jedoch ist dieser Begriff oberflächlich, wenn der genaue Anwendungsbereich z.B. in der Zahnmedizin liegt und in keinem anderen medizinischen Bereich verwendet werden kann. Um jetzt nicht die komplette Domäne zu spezifisch werden zu lassen, kann mit der Subdomäne der genaue Anwendungsbereich, zur Oberdomäne, angegeben werden. In diesem Beispiel wäre es Medicine mit der Subdomäne dentistry.

Die Auswahl der Anwendungsdomäne ist rein subjektiv. Das bedeutet, die Anwendungsdomäne wird aus der Erweiterung selbst abgeleitet. Diese Einordnung ergibt sich meist aus dem Text der Erweiterung oder durch die gezeigten Beispiele der Autoren.

Klasse Beschreibt die Klassifizierung der Erweiterung. Mögliche Klassen sind z.B. Security, Risk oder Time. Die Kategorie Klasse ist sozusagen der Überbegriff für den Zweck der Erweiterung. Damit können Erweiterungen aus verschiedenen Anwendungsdomänen und verschiedenen Ansätzen für ein gleiches übergeordnetes Ziel zusammengefasst werden. In manchen Situationen ist ein Begriff nicht exakt genug, um eine Extension zu beschreiben. Aus diesem Grund kann bei einigen Erweiterungen in Klammern eine sogenannte Subklasse stehen, die die Oberklasse näher beschreiben soll. Zum Beispiel beschreibt die Klasse *Requirements*, eine Erweiterung, die sich mit Anforderungen beschäftigt. Jedoch ist nicht klar ersichtlich, um welche Art von Anforderungen es sich hier genau handelt. Mit einem Zusatz wird daraus zum Beispiel *Requirements(Quality)* und beschreibt somit, dass die Erweiterung sich mit Qualitätsanforderungen beschäftigt. Damit ist gewährleistet, dass eine Extension sowohl mit anderen Extensions aus der selben Oberklasse zusammengefasst werden können, als auch dass sich diese wieder im Detail unterscheiden lassen können. Auch hier ist die Auswahl, genau wie bei der Anwendungsdomäne, rein subjektiv. Die Einordnung ergibt sich aus dem Text der Erweiterung. Der genaue Begriff der Klasse ergibt sich entweder, indem nach häufig verwendeten Schlagwörtern im Text gesucht wird, wie z.B. Security oder Manufacturing, oder wenn das Thema der Erweiterung direkt von den Autoren benannt wird.

Anzahl neuer Symbole Gibt an, wie umfangreich eine Extension mit neu hinzugefügten Symbolen ist. Damit auch hier die Extensions zusammengefasst werden können, wird hier definiert, dass eine Extension mit einer Anzahl von weniger als 5 Symbolen als Extension mit kleinem Umfang bezeichnet wird. Eine Extension mit mehr als 5 aber weniger als 10 Symbolen wird als Extension mit mittel großem Umfang bezeichnet und eine Extension mit mehr als 10 Symbolen wird als eine Extension mit großem Umfang bezeichnet. Eine Extension mit genau 5 Symbolen gehört noch zu einer Extension mit mittel großem Umfang und eine Extension mit genau 10 Symbolen, gehört laut Definition zu einer Extension mit großem Umfang.

Die Größe der Symbolpalette soll hierbei zeigen, wie detailreich und umfangreich die Erweiterung ist und wie viele neue Symbole erlernt werden müssen, um diese anzuwenden. Diese Anzahl berücksichtigt nur die konkrete Syntax, also die Symbole die grafisch neu hinzugefügt und explizit in der Arbeit genannt und dargestellt werden. Symbole die nicht explizit dargestellt, aber in einer exemplarischen Aufgabe verwendet werden, werden daraus abgeleitet und in eckigen Klammern geschrieben. Dazu wird auch, falls Vorhanden, dass Metamodell hinzugezogen. Farbliche Veränderungen eines Elements oder auch Transformationen, werden z.B. nicht zur Anzahl berücksichtigt. Der Grund dafür ist, dass diese Symbole nicht einheitlich sind. Das bedeutet, es kann je nach dem Nutzen des Problems eines Nutzers angepasst werden. Im Beispiel der farblichen Veränderung [2.4.11], kann der Nutzer z.B. die Farben selbst wählen, oder werden von einer Process Engine automatisch gewählt. Dadurch hat das gleiche Symbol eine unterschiedliche Aussage und alle verschiedenen farblichen Kombinationen sind teilweise nicht exakt nennbar oder würden den Rahmen sprengen.

Ausführbar Mit diesem Eintrag soll dargestellt werden, ob die Extension ausführbar ist oder nicht. Im generellen Fall ist jedes BPMN Modell ausführbar. Mit diesem Eintrag soll im Falle von „Ja“ verdeutlicht werden, dass die Erweiterung dazu gedacht ist ausgeführt zu werden, damit es den vorgestellten Nutzen erreicht.

Lernschwierigkeit Die Lernschwierigkeit soll beschreiben, wie schwer die neue Extension zu erlernen ist. Da dies nicht einfach zu identifizieren ist, ist es in diesem Fall kaum möglich ein objektives Ranking zu erstellen. Deswegen sollen hier Pro und Contra Punkte hinzugefügt werden, die beschreiben sollen, worin subjektiv betrachtet mögliche Schwierigkeiten gesehen werden, oder auch mögliche Erleichterungen beim Erlernen der Extension. Diese sollen dann gegenübergestellt werden mit „-“ und „+“. Die Punkte die einem Lernschwierigkeiten bereiten sind mit einem „-“ gekennzeichnet, die Punkte die einem das Erlernen der Extension erleichtern sind mit einem „+“ gekennzeichnet.

Unterstützte Modellierungsprogramme Beinhaltet ein oder auch mehrere Modellierungsprogramme die schon von der Erweiterung unterstützt werden. Manche Erweiterungen sind speziell für Modellierungsprogramme entwickelt oder besitzen schon ein fertiges Plugin.

BPMN 2.0 Regelkonform Seit BPMN 2.0 gibt es einen Erweiterbarkeitsmechanismus. Die Erweiterungen sollten darauf basieren, sodass diese mit dem Core BPMN Regelkonform sind. Manche Erweiterungen bauen nicht auf diesem Mechanismus auf. Selbst wenn diese nur einzelne Attribute der Core BPMN Elemente verändern, sind diese

nicht Regelkonform mit der Definition die für diese Kategorie vereinbart wurde. Aus diesem Grund werden diese als nicht Regelkonform bezeichnet. Ob eine Extension Regelkonform ist oder nicht wird aus der jeweiligen Extension abgeleitet, indem dies von den Autoren explizit angegeben wird oder wenn das Metamodell der Extension auf die Verwendung des Mechanismus hinweist.

BPMN Version Gibt an, für welche BPMN Version die Erweiterung entwickelt worden ist.

Keywords Die Keywords sollen die Extension in einfachen Worten beschreiben. Dies ist wichtig für die spätere Suche auf der Webseite. Die Keywords werden aus subjektiver Sicht gebildet. Meistens sind es Wörter, die beim Lesen des Textes mehrfach auffallen oder genannt werden.

2.4 Analyse

2.4.1 Time BPMN

2.4.1.1 Metadaten

Autoren	Denis Gagne ¹ , Andre Trudel ²
Veröffentlichungsjahr	2009
Veröffentlichungsform	Conference
Universität/Unternehmen	Acadia University ² & Trisotech ¹
Link	IEEE Xplore Digital Library, [11]
Veröffentlichung	Commerce and Enterprise Computing, 2009. CEC '09. IEEE Conference on
Anwendungsdomain	General
Klasse	Time
Anzahl neuer Symbole	15
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Zwei gut verständlich erklärte Beispiele. + Neue Semantik und Symbole gut erklärt. - Einarbeitung in die vorhergehende Arbeit empfehlenswert. - Viele verschiedene Temporal Constraint Decorator Marken, die sich recht ähnlich sehen.
Unterstützte Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Nein
BPMN Version	1.2
Keywords	Symbol Extension, Time, Time dependencies, Time restrictions, scheduling, Time Capturing, Real World

2.4.1.2 Beschreibung

In dieser Erweiterung wird die zeitliche Perspektive in die Standard BPMN Notation hinzugefügt. Unter zeitliche Perspektive versteht sich hier die **zeitliche Abhängigkeit** und die **zeitliche Einschränkungen**. Zeitliche Abhängigkeiten werden hier definiert als eine Aktivität, die in Beziehung mit einer anderen Aktivität zeitlich Abhängig ist. So können Aktivitäten voraus oder verspätet gestartet werden, im Bezug zueinander. Bei zeitlichen

Einschränkungen wird eine Aktivität durch Start- und Endzeiten eingeschränkt. Diese Erweiterung stützt sich auf eine vorhergegangene Arbeit der beiden oben genannten Autoren⁵. Weitere zeitliche Faktoren wie Dauer oder Zeitpunkte werden in dieser Erweiterung nicht näher betrachtet. Jedoch werden diese in Form des TimeDateExpressions, welches in BPMN 1.2 Standardmäßig enthalten ist, hinzugefügt, besitzen aber keine grafische Notation.

⁵<http://pheasant.acadiau.ca/Publications/BPMhandbook.pdf>

2.4.2 BPMN Meta-Model Extension with Deployment and Security

2.4.2.1 Metadaten

Autoren	Mouna Rekik ¹ , Khouloud Boukadi ² , Hanane Ben-Abdallah ²
Veröffentlichungsjahr	2013
Veröffentlichungsform	Conference
Universität/Unternehmen	InfoRmation systems & Advanced Computing Laboratory ¹ , University of Sfax ²
Link	IEEE Xplore Digital Library, [22]
Veröffentlichung	The 13th International Arab Conference on Information Technology ACIT'2012 Dec.10-13
Anwendungsdomain	Cloud System Domain
Klasse	Security
Anzahl neuer Symbole	-
Ausführbar	Ja
Lernschwierigkeit	<ul style="list-style-type: none"> + Metamodell gut dargestellt und erklärt. + Die verwendeten Sicherheitsbegriffe werden verständlich erklärt. + Ausführliches Beispiel. - Der Optimierungsalgorithmus muss selber erstellt oder organisiert werden. - Genaue Ausführungsmechanik wird nicht genau erklärt.
Unterstützte Modellierungsprogramme	-
BPMN Regelkonform	Nein
BPMN Version	2.0
Keywords	Hybrid Cloud, Cloud, risk, security, secure, outsourcing, execution, Cloud computing, Artificial Intelligence

2.4.2.2 Beschreibung

Die Frage, welche Geschäftsprozesse in einer Cloud ausgelagert werden können, ist eine wichtige und sicherheitstechnisch nicht leicht zu beantwortende Frage. Durch das veröffent-

lichen von internen wichtigen Daten können bei einem Datenleck des Cloud Providers ein großer Schaden des Unternehmens folgen. Mit dieser Frage beschäftigt sich jedoch diese Erweiterung. Durch diese Erweiterung ist es möglich in BPMN Annotationen an sicherheitskritischen BPMN Elementen vorzunehmen. Mögliche Annotationen sind z.B. *access control* oder *privacy*. Durch die Ausführung des Modells via einer künstliche Intelligenz, resultiert daraus ein BPMN Modell, dass die Auslagerung der Aktivitäten in die Cloud darstellt.

2.4.3 Extending BPMN for Supporting Customer-Facing Service Quality Requirements

2.4.3.1 Metadaten

Autoren	Kawther Saeedi ¹ , Liping Zhao ¹ , Pedro R. Falcone Sampaio ²
Veröffentlichungsjahr	2010
Veröffentlichungsform	Conference
Universität/Unternehmen	School of Computer Science ¹ & Manchester Business School ² , The University of Manchester
Link	IEEE Xplore Digital Library, [30]
Veröffentlichung	2010 IEEE International Conference on Web Services
Anwendungsdomain	General
Klasse	Requirements(Quality)
Anzahl neuer Symbole	1
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Die Begriffe der nichtfunktionalen Anforderungen verständlich erklärt. + Metamodell gut erklärt. - Beispiel relativ kurz.
Unterstütze Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Ja
BPMN Version	2.0
Keywords	Quality Requirements, Workflow, Services Composition, Symbol Extension, Non Functional Requirements, Cost, Reliability, Time

2.4.3.2 Beschreibung

Diese Erweiterung bietet eine Möglichkeit an in BPMN nichtfunktionale Anforderungen darzustellen, genauer gesagt handelt es sich um die Qualitätsanforderungen. Dazu gehören die drei Faktoren, **Zeit**, **Kosten** und **Zuverlässigkeit**. Damit soll die Erweiterung folgende Lücken in BPMN, laut den Verfassern, schließen.

- ◇ Das Problem, ähnliche Dienste mit gleichen funktionalen Anforderungen von unterschiedlichen Anbietern zu unterscheiden.
- ◇ Die Schwierigkeit, nichtfunktionale Anforderungen schon in der Modellierungsphase zu erfassen, da der Hauptfokus in dieser Phase hauptsächlich in den funktionalen Anforderungen liegt.

2.4.4 A Proposal of BPMN Extensions for the Manufacturing Domain

2.4.4.1 Metadaten

Autoren	Sema Zor, David Schumm, Frank Leymann
Veröffentlichungsjahr	2011
Veröffentlichungsform	Conference
Universität/Unternehmen	Universität Stuttgart
Link	CiteSeerX, [33]
Veröffentlichung	Proceedings of the 44th CIRP International Conference on Manufacturing Systems
Anwendungsdomäne	Manufacturing domain
Klasse	Manufacturing
Anzahl neuer Symbole	9
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Ausführliches Beispiel. + Neue Symbole gut erklärt. - Viele neue Symbole.
Unterstützte Modellierungsprogramme	ORYX ⁶
BPMN 2.0 Regelkonform	Ja
BPMN Version	2.0
Keywords	Manufacturing Process Modeling, Production Process Modeling, Symbol Extension, Production Flow, Product Flow, Tools, Workflow

⁶ <http://workflow-tool.de/oryx-webasierter-bpmn-editor/>

2.4.4.2 Beschreibung

Diese Erweiterung macht es möglich, dass BPMN in Herstellungsdomänen verwendet werden kann. Das bedeutet es können Produktionsprozesse exakter und detaillierter modelliert werden, um z.B. darzustellen mit welchem Werkzeug ein bestimmtes Material verarbeitet wird oder wohin das Material im Produktionsfluss hinfließt, sich aufteilt oder wieder zusammengefügt wird. Des Weiteren können auch handwerkliche Aufgaben beschrieben werden, wie z.B. das reparieren eines Motors oder der gleichen. Die Autoren möchten deswegen durch diese Erweiterung auch eine einheitliche Sprache zwischen der Geschäfts-

und Herstellungsebene schaffen. Wirtschaftsanalytiker könnten damit besser den Herstellungsprozess nachvollziehen, und verstehen und vor allem mit den jeweiligen Arbeitern sich besser verständigen.

2.4.5 A BPMN Extension for Including Data Quality Requirements in Business Process Modeling

2.4.5.1 Metadaten

Autoren	Alfonso Rodriguez ¹ , Angelica Caro ¹ , Cinzia Cappiello ² , Ismael Caballero ³
Veröffentlichungsjahr	2012
Veröffentlichungsform	Conference
Universität/Unternehmen	University of Bio Bio ¹ , Dipartimento di Elettronica e Informazione - Politecnico di Milano ² , University of Castilla-La Mancha ³
Link	Springer Verlag, [29]
Veröffentlichung	Proceedings of the 44th CIRP International Conference on Manufacturing Systems
Anwendungsdomäne	General
Klasse	Requirements(Quality)
Anzahl neuer Symbole	6
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Neue Symbole gut erklärt. + Metamodell gut erklärt. + Beispiel erklärt sowohl neue Symbole, als auch das gesamte Vorgehen der Methodologie. - Einarbeitung in die Methodologie.
Unterstütze Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Ja
BPMN Version	2.0
Keywords	Data Quality Requirements, Symbol Extension, Requirements, Procedure Model, Data, Data Integrity, Data Quality Methodology, Methodology

2.4.5.2 Beschreibung

Datenqualität ist eines der Hauptgründe für den Erfolg eines Geschäftsprozesses. Jedoch ist es in BPMN 2.0 momentan nicht möglich diese Qualitätsanforderungen in das Modell zu integrieren. Durch diese Erweiterung wird diese Lücke gefüllt. Die Erweiterung schlägt nicht nur sechs neue Symbole vor, sondern eine ganze Vorgehensweise, wie die

Datenqualität in einem Geschäftsprozess auf den verschiedenen Ebenen, also vom Geschäftsanalysten bis hin zum Systemanalysten, modelliert werden soll. Die Erweiterung ist zum einen für den Geschäftsanalysten gedacht, für die High Level Modellierung, also auch für Datenqualitätsexperten, die für die Erstellung für das Low Level Modell zuständig sind.

2.4.6 A Extension of BPMN Meta-Model for Evaluation of Business Processes

2.4.6.1 Metadaten

Autoren	Azeem Lodhi, Veit Köppen, Gunter Saake
Veröffentlichungsjahr	2011
Veröffentlichungsform	Article
Universität/Unternehmen	Universität Magdeburg
Link	Verlag Walter de Gruyter, [17]
Veröffentlichung	Scientific Journal of Riga Technical University
Anwendungsdomäne	Businessdomain
Klasse	Analysis
Anzahl neuer Symbole	-
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Beispiel hilft beim Verständnis der Erweiterung. - Metamodell weder ausführlich dargestellt noch gut erklärt. - Verwirrende Symbole die nicht eindeutig erklärt werden. - Ein Beispiel, das nicht alles abdeckt was erweitert wird.
Unterstütze Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Nein
BPMN Version	2.0
Keywords	analysis, improvement, post execution analysis, symbol extension, color extension, evaluation, cost, time, quantitative measurement, qualitative measurement

2.4.6.2 Beschreibung

Diese Arbeit beschäftigt sich mit dem Thema Post Execution Analysis. Dies fällt in die Kategorie Evaluierung bzw. Analyse eines Prozess Modells nach der Ausführung. Im Gegensatz zur normalen Analyse, die sich hauptsächlich mit der Ausführungsreihenfolge beschäftigt, beschäftigt sich die Post Execution Analysis mit der Performanz eines Geschäftsobjekts und Prozesses und bringt diese in Zusammenhang. Dies soll nun durch die Erweiterung in BPMN mit z.B. Farben visualisiert werden. Durch die Farbe grün soll z.B. dargestellt werden, dass ein Service weniger Geld kostet als ein anderer Service, der mit der Farbe rot hinterlegt ist. Es kann auch die Zeit oder jede andere Anforderung farblich dargestellt werden, je nach gewünschten Nutzen. Die visualisierten Daten werden aus Log-Dateien die das System auswirft, abgeleitet und modelliert. Eine Automatisierung des Vorgangs wird in einer zukünftigen Arbeit weiter behandelt.

2.4.7 BPMN for REST

2.4.7.1 Metadaten

Autoren	Cesare Pautasso
Veröffentlichungsjahr	2011
Veröffentlichungsform	Incollection
Universität/Unternehmen	University of Lugano
Link	Springer Verlag, [26]
Veröffentlichung	Third International Workshop, BPMN 2011, Lucerne, Switzerland
Anwendungsdomäne	Information domain
Klasse	Webservice
Anzahl neuer Symbole	1
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Verschiedene Beispiele. + Problem präzise beschrieben. - Kein Metamodell vorhanden.
Unterstützte Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Nein
BPMN Version	2.0
Keywords	REST, Symbol Extension, Webservices, RESTfull Services, Service-Oriented-Architecture, SOA, Resource-Oriented-Architecture, Distributed System

2.4.7.2 Beschreibung

Diese Erweiterung beschäftigt sich mit einer besseren Visualisierung für den Webservice REST. Damit ist es möglich schon in der Design Phase des BPMN Modells explizit die verwendeten Ressourcen darzustellen. Es soll mit der neuen minimalistischen Notation möglich sein, die folgenden Fragen in einem BPMN Model visuell zu beantworten:

- ◇ Von welchen Ressourcen hängt der Prozess ab, bevor er erfolgreich ausgeführt werden kann?
- ◇ Welche Ressourcen sind von der Ausführung des Prozesses betroffen?
- ◇ Welche Tasks eines Prozesses sind dafür Verantwortlich, das die Clienten Zugriff auf die Ressourcen besitzen?
- ◇ Was sind die möglichen Anfragen, die zu einer Ressource gesendet werden können, dessen Verhalten in einem Prozess spezifiziert sind?

Damit sollen die Low Level Informationen, die für den Technikspezialisten von Interesse sind, abstrahiert werden und eine High Level Modellierung, ohne technisch spezifischen Deklarationen, ermöglicht werden.

2.4.8 A Business Process Modeling Notation Extension for Risk Handling

2.4.8.1 Metadaten

Autoren	Bartosz Marcinkowski, Michal Kuciapski
Veröffentlichungsjahr	2012
Veröffentlichungsform	Incollection
Universität/Unternehmen	University of Gdansk
Link	Springer Verlag, [19]
Veröffentlichung	IFIP International Federation for Information Processing 2012
Anwendungsdomäne	General
Klasse	Risk
Anzahl neuer Symbole	[12]
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Ausführliches Metamodell. + Ausführliches Beispiel. + Intuitive Anwendung der Erweiterung. - Neue Symbole grafisch nicht dargestellt.
Unterstützte Modellierungsprogramme	ADONIS ⁷
BPMN 2.0 Regelkonform	Nein
BPMN Version	2.0
Keywords	Risk Management, Symbol Extension, Risk Handling, Risk Factor, Risk Types, Occurrence Probability, Risk Handling, Risk Ranking, Human Risk, Physical Risk

⁷ <http://www.adonis.eu/>

2.4.8.2 Beschreibung

Die Identifizierung von Bedrohungen oder Risiken und die richtige Behandlung von diesen ist in manchen Geschäftsprozessen ein wichtiger und essentieller Bestandteil. Jedoch ist es in BPMN 2.0 nicht so ohne weiteres möglich, formell richtig eine identifizierte Bedrohung zu spezifizieren, sodass sofort ersichtlich ist, welche Maßnahmen ergriffen werden sollen oder müssen um, dieses Risiko zu minimieren oder beim Eintreten richtig zu behandeln. Durch die Verwendung dieser Erweiterung können Annotationen vorgenommen werden, mit denen sofort ersichtlich ist, welche Auftrittswahrscheinlichkeit und welche Auswirkung

das Risiko besitzt und welche Gegenmaßnahmen getroffen werden können. Es ist ebenfalls sofort einsehbar, welche Aktivität ein Risikofaktor besitzt.

2.4.9 BPMN4CP Design and Implementation of a BPMN Extension for Clinical Pathways

2.4.9.1 Metadaten

Autoren	Richard Braun, Hannes Schlieter, Martin Burwitz, Werner Esswein
Veröffentlichungsjahr	2014
Veröffentlichungsform	Conference
Universität/Unternehmen	Technische Universität Dresden
Link	IEEE Xplore Digital Library, [5]
Veröffentlichung	2014 IEEE International Conference on Bioinformatics and Biomedicine
Anwendungsdomäne	Healthcaredomain
Klasse	Medicine
Anzahl neuer Symbole	7
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Konkrete Syntax dargestellt. + Ausführliches Beispiel. + Ausführliche Analyse der Vorhandenen BPMN Elemente im Bezug auf die domänenspezifische Erweiterung.
Unterstützte Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Ja
BPMN Version	2.0
Keywords	Symbol Extension, Clinical Pathways, Healthcare, Patient Treatment, Workflow

2.4.9.2 Beschreibung

Diese Erweiterung beschäftigt sich mit einer besseren Integration von Clinical Pathways, zu deutsch klinische Behandlungspfade. Das bedeutet, es kann durch diese Erweiterung klinische Behandlungspfade wie z.B. eine Weisheitszahn Behandlung oder jede andere medizinische Behandlung visualisiert und verbessert werden.

2.4.10 BPMN Security Extensions for Healthcare Process

2.4.10.1 Metadaten

Autoren	Koh Song Sang, Bo Zhou
Veröffentlichungsjahr	2015
Veröffentlichungsform	Conference
Universität/Unternehmen	Liverpool John Moores University
Link	IEEE Xplore Digital Library, [31]
Veröffentlichung	2015 IEEE International Conference on Computer and Information Technology
Anwendungsdomäne	Healthcaredomain
Klasse	Security
Anzahl neuer Symbole	17
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Neue BPMN Symbole ausführlich dargestellt. + Ein verständliches Beispiel. - Viele neue Symbole.
Unterstützte Modellierungsprogramme	Activiti ⁸
BPMN 2.0 Regelkonform	Nein
BPMN Version	2.0
Keywords	Security, Requirement, Healthcare, Internet of Things, Symbol Extension, Integrity, Availability, Confidentiality, Security Indicators

⁸ <http://activiti.org/>

2.4.10.2 Beschreibung

Diese Erweiterung spezialisiert sich auf die Sicherheit im Gesundheitswesen. Es ist ein komplexes System, indem nicht nur die Daten von Patienten lagern, sondern auch die der Ärzte, als auch von verschiedenen Institutionen im Gesundheits- und Versicherungswesen. In der heutigen Zeit, in der alles online schnell zugänglich gemacht werden kann und muss, werden die Sicherheitsanforderungen meistens zu einem späteren Zeitpunkt der Entwicklung erstellt. Dies ist im Bezug auf das Gesundheitswesen, indem die Privatsphäre einer elektronischen Krankenakte hohe Priorität hat, katastrophal. Deswegen ist es wichtig, diese Anforderungen schon auf der Entwurfsebene zu treffen und diese ins BPMN Modell zu integrieren, anstelle dies erst danach von Sicherheitsspezialisten hinzuzufügen zu lassen.

Mit dieser Erweiterung können diese Sicherheitsanforderungen erhoben werden. Mögliche Sicherheitsanforderungen, die erhoben werden können, sind z.B. Integrität, Verfügbarkeit und Vertraulichkeit.

2.4.11 CF4BPMN: a BPMN extension for controlled flexibility in business

2.4.11.1 Metadaten

Autoren	Ricardo Martinho ^{1, 2} , Dulce Domingos ³ , Joao Varajao ⁴
Veröffentlichungsjahr	2015
Veröffentlichungsform	Article
Universität/Unternehmen	Polytecnic Institute of Leiria ¹ , Center for Health Technology and Services Research ² , Universidade de Lisboa ³ , University of Minho ⁴
Link	ScienceDirect, [20]
Veröffentlichung	Conference on ENTERprise Information Systems/International Conference on Project MANagement/-Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN
Anwendungsdomäne	General
Klasse	Flexibility
Anzahl neuer Symbole	-
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Beispiel hilft beim Verständnis. - Constraint of Change Notation kompliziert erklärt. - Kein konkretes BPMN Metamodell vorhanden.
Unterstützte Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Ja
BPMN Version	2.0
Keywords	Process Flexibility, Flexibility, Execution, Color Extension, Controlled Flexibility, Constraint of Change, COfChange

2.4.11.2 Beschreibung

Kontrollierte Flexibilität wird immer wichtiger für einen Prozess Designer. Unter kontrollierte Flexibilität versteht man, dass gewisse Prozessabschnitte flexibel gestaltet, aber dennoch gerichtet sein sollen. Laut den Autoren der Erweiterung möchten viele nicht unbedingt selbst flexibel Arbeiten. Sie wollen eher durch den Arbeitsprozess durchgeleitet werden. Deswegen muss diese Flexibilität durch die Prozess Designer vordefiniert werden. Diese definieren dann, wo, wie und wer eine bestimmte Aufgabe im Arbeitsprozess ändern darf. Das wird durch diese BPMN Extension dadurch ermöglicht, indem flexible Strukturen die Farbe grün erhalten. Dies ist mit jedem grafischen BPMN Symbol möglich z.B. mit einer Pool Lane, Gateway oder mit einem Sub-Process. Dazu wird eine Text Annotation hinzugefügt, welches genauere Information zu den Fragen wo, wie und wer besitzt. Das Modell basiert auf der Constraint of Change Notation, welche in der Arbeit der Autoren erklärt wird.

2.4.12 Extending the BPMN Syntax for Requirements Management

2.4.12.1 Metadaten

Autoren	Sascha Goldner ¹ , Alf Papproth ²
Veröffentlichungsjahr	2011
Veröffentlichungsform	Incollection
Universität/Unternehmen	Cassidian ¹ , Fraunhofer Application Centre for Logistics System Planning and Information Systems ALI ²
Link	Springer Verlag, [13]
Veröffentlichung	Third International Workshop, BPMN 2011, Lucerne, Switzerland
Anwendungsdomäne	General
Klasse	Security
Anzahl neuer Symbole	2
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Symbole ausführlich beschrieben. - Enthält kein Metamodell.
Unterstützte Modellierungsprogramme	ORYX ⁹
BPMN 2.0 Regelkonform	Nein
BPMN Version	2.0
Keywords	Regulation, Legal Constraints, Requirements Management, Symbol Extension, Security, Law

⁹ <http://workflow-tool.de/oryx-webasierter-bpmn-editor/>

2.4.12.2 Beschreibung

Durch die Verwendung dieser Erweiterung können rechtliche Einschränkungen berücksichtigt werden. Diese werden in Form von Artefakten erweitert. Diese können mit Abstract Tasks, Manual Tasks, Gateways, Sequence Flows, Message Flows, Start- und Endevents verbunden werden. Die Artefakte enthalten dann die rechtlichen Grundlagen, also den genauen Artikel, Paragraph und den Rechtstext selbst, die für eine bestimmte Ausführung einer Aufgabe wichtig sind. Damit kann gewährleistet werden, dass schon in der frühen Planungsphase alle wichtigen rechtlichen Anforderungen berücksichtigt werden können.

2.4.13 BPM4TOSCA

2.4.13.1 Metadaten

Autoren	Oliver Kopp, Tobias Binz, Uwe Breitenbücher, Frank Leymann
Veröffentlichungsjahr	2012
Veröffentlichungsform	Incollection
Universität/Unternehmen	Universität Stuttgart
Link	Springer Verlag, [15]
Veröffentlichung	4th International Workshop, BPMN 2012, Vienna, Austria
Anwendungsdomäne	Information domain
Klasse	Cloud Service
Anzahl neuer Symbole	4
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Symbole ausführlich erklärt. + Transformation in das Standard BPMN gut erklärt. + TOSCA für Laien gut erklärt. - Kein Metamodell vorhanden. - Keine Beispiele vorhanden.
Unterstützte Modellierungsprogramme	Valesca ¹⁰
BPMN 2.0 Regelkonform	Nein
BPMN Version	2.0
Keywords	Symbol Extension, Cloud Computing, Service Management, Management Plans, TOSCA, Cloud

¹⁰ <http://www.cloudcycle.org/valesca/>

2.4.13.2 Beschreibung

Mit dieser Erweiterung ist es möglich, mit dem TOSCA Standard von Oasis leichter zu arbeiten. TOSCA ist die Abkürzung für Topology and Orchestration Specification for Cloud Applications und wird von BPMN 2.0 Standardmäßig für die sogenannten Management Plans unterstützt. Mit dieser Erweiterung jedoch wird die

- ◇ Komplexität verringert,

- ◇ die Möglichkeit gegeben, dass die Management Plans Zugriff auf das TOSCA Topologie Modell bekommen,
- ◇ die Auswahl von Management Operationen erleichtert wird und
- ◇ die Ausführung von Scripts auf Nodes einfacher und komfortabler gestaltet.

Des Weiteren findet sich in der Erweiterung ein Vorgehen, um nach der Verwendung der neu erweiterten Symbole, diese wieder zurückzuführen auf die Standard BPMN Elemente. Diese werden, wie erwähnt, von TOSCA Standardmäßig unterstützt und so kann die Ausführung der Modelle gewährleistet werden.

2.4.14 Entwicklung einer BPMN-Extension für ressourcen-intensive Prozesse im Maschinenbau

2.4.14.1 Metadaten

Autoren	Richard Braun, Werner Esswein
Veröffentlichungsjahr	2014
Veröffentlichungsform	Conference
Universität/Unternehmen	Technische Universität Dresden
Link	-[28]
Veröffentlichung	Tagungsband Multikonferenz Wirtschaftsinformatik 2014
Anwendungsdomäne	Manufacturing domain
Klasse	Optimization
Anzahl neuer Symbole	7
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Neue Symbole verständlich eingeführt. + Ausführliches und gut erklärtes Metamodell. + Verständliches Beispiel. - Domänenspezifisches Wissen vorausgesetzt.
Unterstützte Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Ja
BPMN Version	2.0
Keywords	Machine Engineering, Optimization, Symbol Extension, Heat Flow Optimization, Compensation Procedure, Correction Procedure, Ressources, Reduce Thermal Production Fault

2.4.14.2 Beschreibung

In dieser Erweiterung wird ein Problem im Herstellungsprozess adressiert, nämlich dass in verschiedenen Herstellungs- oder Bearbeitungsprozessen bei Werkzeugmaschinen ein Teil der Energie in Hitze umgewandelt wird. Diese kann zum einen die Produktivität verringern, als auch zum anderen Mangel am erzeugten Produkt verursachen. Um dies zu verhindern existieren Optimierungsverfahren, wie z.B. das sogenannte Kompensationsverfahren oder das Korrekturverfahren. Diese sehr domänenspezifischen Verfahren können durch die Anwendung dieser Erweiterung in BPMN modelliert werden. Die Dokumentation und die

Kommunikation profitieren durch die explizite Darstellung der nicht trivialen Verfahren. Es werden dafür neue Ressourcenobjekte hinzugefügt, die für die Darstellung der Verfahren wichtig sind.

2.4.15 A BPMN Extension for Modeling Non Functional Properties of Business Processes

2.4.15.1 Metadaten

Autoren	Paolo Bocciarelli, Andrea D'Ambrogio
Veröffentlichungsjahr	2011
Veröffentlichungsform	Proceeding
Universität/Unternehmen	University of Rome
Link	ACM Digital Library, [3]
Veröffentlichung	TMS-DEVS '11 Proceedings of the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium
Anwendungsdomäne	General
Klasse	Requirements(Functional/non functional)
Anzahl neuer Symbole	-
Ausführbar	Ja
Lernschwierigkeit	<ul style="list-style-type: none"> + Die einzelnen Metamodell Klassen sind gut erklärt. - Beispiel nicht ausführlich genug.
Unterstützte Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Nein
BPMN Version	2.0
Keywords	Requirements, Functional, Non Functional, Reliability, Performance, Prediction, Model Driven Architecture, Transformation, MARTE

2.4.15.2 Beschreibung

Dies ist eine Erweiterung, die sich mit der Charakterisierung von funktionalen und nicht-funktionalen Anforderungen beschäftigt. Sie nennt sich kurz PyBPMN, und konzentriert sich auf die Performance und die Zuverlässigkeit eines Geschäftsprozesses. Im Wesentlichen ist diese Erweiterung dazu gedacht, verschiedene Alternativen eines Geschäftsprozesses

oder verschiedenen Services schon in der Design Phase, mit Hilfe von einer Vorhersage zu vergleichen. Dies geschieht so, dass ein BPMN Modell durch die Verwendung von der PyBPMN Erweiterung erstellt wird, welches dann ausgeführt wird.

2.4.16 Aspect-Oriented Business Process Modeling with AO4BPMN

2.4.16.1 Metadaten

Autoren	Anis Charfi ¹ , Heiko Müller ¹ , Mira Mezini ²
Veröffentlichungsjahr	2010
Veröffentlichungsform	Conference
Universität/Unternehmen	SAP Research CEC Darmstadt ¹ , Darmstadt University of Technology ²
Link	Springer Verlag, [8]
Veröffentlichung	6th European Conference, ECMFA 2010, Paris, France, June 15-18, 2010. Proceedings
Anwendungsdomäne	General
Klasse	Visualization(Structuring)
Anzahl neuer Symbole	[3]
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Symbole werden gut erklärt. + Das Aspekt-orientierte Problem ist gut erklärt. + Minimale Erweiterung. - Kein Metamodell, Zusammenhänge zum BPMN Metamodell nicht ersichtlich.
Unterstütze Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Nein
BPMN Version	1.2
Keywords	Aspect-Oriented Modeling, Aspects, Join Points, Maintainability, Cross-Cutting-Concerns

2.4.16.2 Beschreibung

Aspekt-orientiertes Modellieren beschreibt in diesem Zusammenhang nichts Anderes, als das bekannte Aspekt-orientierte Programmieren. Zur Erinnerung, Aspekt-orientiertes Programmieren ist dazu gedacht, Funktionalitäten die an mehreren unterschiedlichen Stellen

der Anwendung verwendet werden, aber immer wieder die selbe Aufgabe erledigen zusammenzufassen, sprich zu modularisieren. Damit sind die nichtfunktionalen Anforderungen gemeint, wie z.B. Logging, Tracing oder Prüfungen. Mit AO4BPMN ist es möglich in BPMN diesen Aspekt orientierten Ansatz zu verfolgen und somit die sogenannten Cross-Cutting-Concerns zu modularisieren. Die Erweiterung bietet zwei Varianten an. Die **light-weight** Variante verwendet bestehende BPMN Elemente. Die **heavy-weight** Variante präsentiert eine komplett neue grafische Repräsentation, welche mit Eclipse GMF realisiert wurde. Jedoch existiert kein Verweis auf diesen Editor.

2.4.17 Extending BPMN for Business Activity Monitoring

2.4.17.1 Metadaten

Autoren	Jan-Philipp Friedstab ¹ , Christian Janiesch ² , Martin Matzner ¹ , Oliver Müller ³
Veröffentlichungsjahr	2010
Veröffentlichungsform	Conference
Universität/Unternehmen	University of Muenster ¹ , Karlsruhe Institute of Technology ² , University of Liechtenstein ³
Link	IEEE Xplore Digital Library, [10]
Veröffentlichung	System Science (HICSS), 2012 45th Hawaii International Conference on
Anwendungsdomäne	General
Klasse	Monitoring
Anzahl neuer Symbole	31
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Abstrakte Syntax ausführlich erklärt. + Metamodell vorhanden. Erleichtert die Einordnung und den Zusammenhang der neuen Notation. + Beispiel vorhanden. - Große grafische Notation.
Unterstützte Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Nein
BPMN Version	2.0
Keywords	Business Activity Monitoring, BAM, Key Performance Indicators, Real-Time Access, Monitoring, Duration, Frequency, Aggregated Measure, Actions, Target definition, Symbol Extension

2.4.17.2 Beschreibung

Mit Business-Activity-Monitoring kann in Echtzeit auf Schlüsselindikatoren zugegriffen werden. Damit kann der Status von verschiedenen Operationen, Prozessen oder Transaktionen in Echtzeit visualisiert werden daraufhin können bessere Geschäftsentscheidungen getroffen werden. Das Entwerfen und Warten solcher Systeme gestaltet sich jedoch als

schwierig, da mehrere verschiedene Personen aus unterschiedlichen Anwendungsgebieten involviert sind und diese auf unterschiedlichen Abstraktionsebenen kommunizieren. Genau diese Lücke soll diese Erweiterung schließen, in dem es das Core BPMN um Business-Activity-Monitoring Konzepte erweitert und so eine einheitliche Ansicht auf die komplette Business-Activity-Monitoring Anwendung ermöglicht.

2.4.18 Extending BPMN 2.0 for Modelling the Combination of Activities

2.4.18.1 Metadaten

Autoren	Luisa Parody, Teresa Gomez-Lopez, Rafael M. Gasca
Veröffentlichungsjahr	2012
Veröffentlichungsform	Proceeding
Universität/Unternehmen	University of Sevilla
Link	Springer Verlag[25]
Veröffentlichung	4th International Workshop, BPMN 2012, Vienna, Austria
Anwendungsdomäne	General
Klasse	Optimization
Anzahl neuer Symbole	1
Ausführbar	Ja
Lernschwierigkeit	<ul style="list-style-type: none"> + Neues Symbol gut erklärt. + Beispiel vorhanden. - Beispiel beim mehrmaligen durchlesen erst verständlich. - Viele Formale Definitionen. - Kein Metamodell vorhanden.
Unterstützte Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Nein
BPMN Version	2.0
Keywords	Combination of Activities, Data Input, Data Output, Data Constraint, Common Business Goal, Optimization, Symbol Extension, Execution, Data Range

2.4.18.2 Beschreibung

Heutzutage existieren viele verschiedene Webseiten, auf denen Flüge und Hotels gebucht werden können. Mit dieser Erweiterung ist es nun möglich, solche Aufgaben zu modellieren und auszuführen. Das eben genannte Beispiel greift die Erweiterung auf und beschreibt, dass dies mit den herkömmlichen BPMN Elementen in manchen Szenarien nur unzureichend zu erstellen ist. Die Suche mit verschiedenen Eingabeparametern, insbesondere mit einem Eingabeparameterbereich, in diesem Fall z.B. eine Auswahl von mehreren Ankunftsorten und Abflugorten, sowie eine variable Zeitspanne für die Dauer des Aufenthalts. Dies sind viele unterschiedliche Variablen, insbesondere die Zeitspannen, die beachtet werden müssen, bei der Optimierung nach dem z.B. günstigsten Preis.

Das Problem ist hierbei, dass die Ausführung der Aktivitäten, Flug- und Hotelsuche, zwar unabhängig sind, die Daten jedoch, voneinander abhängen, um das globale optimale Ergebnis zu erreichen. Zusammengefasst, der Hauptzweck dieser Erweiterung ist es, nach einer speziellen Zielfunktion zu optimieren, wobei es eine Menge von Eingabedaten gibt, für die die Aktivitäten miteinander zusammenhängen. Mit Hilfe von Einschränkungen soll dann die Kombination der Aktivitäten ermittelt werden. Es wird ein Symbol hinzugefügt, das dies beschreibt. Die Ausführung basiert auf einem Distributed Constraint Satisfaction Problem Ansatz. Ein Vergleich wäre das schon bekannte BPMN Element Ad-Hoc.

2.4.19 SecureBPMN: Modeling and Enforcing Access Control Requirements in Business Processes

2.4.19.1 Metadaten

Autoren	Achim D. Brucker, Isabelle Hang, Raj Ruparel, Gero Lückemeyer
Veröffentlichungsjahr	2012
Veröffentlichungsform	Proceeding
Universität/Unternehmen	SAP Researche, Hochschule für Technik Stuttgart
Link	ACM Digital Library, [6]
Veröffentlichung	SACMAT '12 Proceedings of the 17th ACM symposium on Access Control Models and Technologies
Anwendungsdomäne	General
Klasse	Security
Anzahl neuer Symbole	-
Ausführbar	Ja
Lernschwierigkeit	<ul style="list-style-type: none"> + Sicherheitsanforderungen erklärt. - Neue Symbole nicht dargestellt.
Unterstützte Modellierungsprogramme	Activiti ¹¹
BPMN 2.0 Regelkonform	Ja
BPMN Version	2.0
Keywords	Process Security, Access Control, Separation of Duty, Tool Support, Business Process Driven Security, Business Process Driven, Binding of Duty, Need to Know

¹¹ <http://activiti.org/>

2.4.19.2 Beschreibung

Diese Arbeit handelt über die Sicherheit von Geschäftsprozessen. Allerdings ist es hier erlaubt, dass die Erweiterung sowohl Sicherheitsanforderungen in der Modellierungsphase erhebt, als auch in die Ausführung des Geschäftsprozesses erzwingt. Die berücksichtigten Sicherheitsanforderungen sind Access Control, also die Einschränkung auf den Zugriff von Ressourcen und Aktionen, Separation of Duty also, dass mehr als eine Person benötigt wird um ein Prozess zum Abschluss zu bringen. Des Weiteren werden als besonders wichtig folgende Eigenschaften gesehen. Binding of Duty, die selbe Person muss verschiedene Aufgaben bearbeiten und Need to Know, eine Person soll nur auf die Informationen Zugriff

besitzen, die für die Vervollständigung des Prozesses benötigt werden. Diese Erweiterung existiert, als Prototyp, fertig integriert in der Activiti BPMN Plattform. Genauer ist der Activiti Eclipse Designer und die Activiti Process Engine erweitert worden.

2.4.20 Extending BPMN for Modeling Complex Choreographies

2.4.20.1 Metadaten

Autoren	Gero Decker, Frank Puhlmann
Veröffentlichungsjahr	2007
Veröffentlichungsform	Conference
Universität/Unternehmen	University of Potsdam
Link	Springer Verlag, [9]
Veröffentlichung	OTM Confederated International Conferences CoopIS, DOA, ODBASE, GADA, and IS 2007, Vilamoura, Portugal
Anwendungsdomäne	General
Klasse	Visualization (Choreographies)
Anzahl neuer Symbole	[3]
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Erweiterung ausführlich erklärt. + Gutes Beispiel das, dass Problem aufzeigt. - Kein Metamodell vorhanden.
Unterstütze Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Nein
BPMN Version	1.0
Keywords	Language Extension, Symbol Extension, Choreographie, SOA, Service-Oriented-Architecture, Service Interaction Patterns, Service Choreography

2.4.20.2 Beschreibung

Diese Erweiterung beschäftigt sich mit den begrenzten Möglichkeiten in BPMN 1.0 komplexe Choreographien darzustellen. Die Erweiterung stützt sich dabei auf das Service Interaction Pattern, welche wiederkehrende Choreographie Szenarien beschreibt. Laut der Erweiterung sind mit dem Core BPMN, nur fünf der zwölf Szenarien umsetzbar. Um dieser Limitierung entgegenzusetzen, wurden neue grafische Symbole hinzugefügt. Dazu gehört das Participant

Set, welches dafür sorgen soll, dass mehr als eine Person beteiligt ist. Ein Beispiel dafür wäre ein Auktionshaus, in dem ein Participant der Bieter ist. Jetzt existieren mehrere Bieter und durch Participant Sets ist dies darstellbar. Des Weiteren wird es durch neue Referenzen erweitert, in Form von Datenobjekten, welche dafür sorgen sollen zwischen den Participant Sets die verschiedenen Participants zu unterscheiden. Ein Beispiel dafür wäre, dass am Ende einer Auktion einer der Bieter gewinnt. So erhält nur der Gewinner eine Nachricht, dass er gewonnen hat und die Verlierer, dass sie verloren haben. Zuletzt wird BPMN durch ein Reference Passing erweitert. Es ist anzumerken, dass diese Erweiterung in BPMN 2.0 schon integriert ist.

2.4.21 A Notation for Supporting Social Business Process Modeling

2.4.21.1 Metadaten

Autoren	Marco Brambilla, Piero Fraternali, Carmen Vaca
Veröffentlichungsjahr	2011
Veröffentlichungsform	Incollection
Universität/Unternehmen	Politecnico di Milano
Link	Springer Verlag, [4]
Veröffentlichung	Third International Workshop, BPMN 2011, Lucerne, Switzerland
Anwendungsdomäne	Information domain
Klasse	Social Networking
Anzahl neuer Symbole	22
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Neue Symbole sind gut erklärt. + Drei verschiedene Beispiele. - Viele neue Symbole zu erlernen.
Unterstützte Modellierungsprogramme	WebRatio BPM Editor ¹²
BPMN 2.0 Regelkonform	Nein
BPMN Version	2.0
Keywords	Social Software, Social Network, Enterprise 2.0, Code Generation, Social

¹² <http://www.webratio.com/>

2.4.21.2 Beschreibung

Diese Erweiterung wird auch Social BPMN genannt, da sich diese Erweiterung um die bessere Darstellung von sozialen Anforderungen beschäftigt. Das bedeutet, es wird hier möglich sein soziale Interaktionen zu modellieren. Die Motivation dafür ist, dass dadurch die organisatorische Effektivität verbessert werden kann. Ein Beispiel dafür könnte eine Terminplanung für ein Meeting sein, dass mit einer Webseite verknüpft ist, wie zum Beispiel die Doodle Plattform ⁶. Mit den neuen Symbolen für die sogenannten Social Tasks, können z.B. Login, Einladungen in ein Netzwerk oder das Kommentieren einer Aktivität visualisiert werden. Es wird eine Vielzahl von Core BPMN Elementen erweitert, wie z.B. die Pool Lane, Gateways oder Ad-Hoc Tasks. Des Weiteren wurde eine Model-Driven Web Anwendung in WebRatio implementiert. Diese erlaubt es, dass BPMN Modell automatisch in eine ausführbare Java EE Applikation zu transformieren.

⁶<http://www.doodle.com/>

2.4.22 Enhancing BPMN 2.0 Informational Perspective to Support Interoperability for Cross-Organizational Business Processes

2.4.22.1 Metadaten

Autoren	Marija Jancovic, Miroslav Ljubivic, Nenad Anicic, Zoran Marjanovic
Veröffentlichungsjahr	2015
Veröffentlichungsform	Article
Universität/Unternehmen	University of Belgrade
Link	National Library of Serbia, [14]
Veröffentlichung	Computer Science and Information Systems
Anwendungsdomäne	General
Klasse	Requirements(Information)
Anzahl neuer Symbole	-
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Beispiel hilft ein wenig. - Die Arbeit ist an vielen Stellen schwer zu verstehen.
Unterstützte Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Ja
BPMN Version	2.0
Keywords	UML, Interoperability, View, CBP, Cross-Organizational, Information Persepctive, Ontology

2.4.22.2 Beschreibung

Ein weiterer Mangel, den diese Erweiterung in BPMN 2.0 zu schließen versucht ist, dass BPMN für das Erstellen von organisationsübergreifende Geschäftsprozesse keine gelungene Möglichkeit anbietet, die Informationsansicht besser darzustellen. Ohne diese Informationsansicht ist das Modellieren des organisationsübergreifenden Geschäftsprozesses laut den Autoren nicht vollständig möglich. Es wird ein UML View Profile eingeführt. Mit diesen sollen die Informationsanforderungen spezifiziert werden als Sicht über die gemeinsame Referenz Ontologie. BPMN wird dahingehend erweitert, dass diese verschiedenen UML Sichten mit den dazugehörigen Prozessaktivitäten verbunden werden. UML wird benötigt um die Ontologie abzubilden, denn BPMN 2.0 kann die Informationsanforderungen nicht feiner Auflösen, wie es nötig wäre. Möglich wäre dies nur, wenn man das XML Schema dafür

benutzen würde. Nachteil daran wäre, dass dies zu schwer zu verstehen und zu erstellen wäre.

2.4.23 BPMN Extension for Business Process Monitoring

2.4.23.1 Metadaten

Autoren	Anne Baumgrass, Nico Herber, Andreas Meyer, Matthias Weske
Veröffentlichungsjahr	2015
Veröffentlichungsform	Conference
Universität/Unternehmen	University of Potsdam
Link	EMISA, [2]
Veröffentlichung	EMISA Webseite
Anwendungsdomäne	General
Klasse	Requirements(Information)
Anzahl neuer Symbole	-
Ausführbar	Ja
Lernschwierigkeit	+ XML Schema erklärt.
Unterstützte Modellierungsprogramme	Event Processing Plattform ¹³
BPMN 2.0 Regelkonform	Ja
BPMN Version	2.0
Keywords	UML, Interoperability, View, CBP, Cross-Organizational, Information Perspective, Process Event Monitoring Points, PEMPS, Complex Event Processing, CEP

¹³ <http://bpt.hpi.uni-potsdam.de/Public/EPP>

2.4.23.2 Beschreibung

Diese Erweiterung beschäftigt sich ebenfalls mit dem Thema Monitoring. Bei der Ausführung eines Geschäftsprozesses, werden viele Daten generiert, die im weiteren Verlauf Events genannt werden, die man für die Prozessüberwachung, also Monitoring und Analyse, verwenden könnte. Dies ist mit normalen Business Process Management Systemen nicht umsetzbar, da diese das Standardmäßig nicht unterstützen. BPMN 2.0 wird deswegen erweitert, damit sogenannte Process Event Monitoring Points (PEMPs) gesetzt werden können. Diese Punkte definieren einen bestimmten Bereich, indem solche Events erwartet werden. Zusammengefasst erweitert diese Erweiterung BPMN 2.0 dahingehend, dass diese

Events mit dem Prozess Modell verbunden werden können, sodass während der Laufzeit beim Auftreten eines Events dieser sofort mit dem dazugehörigen Prozess verbunden werden kann.

2.4.24 Deontic BPMN

2.4.24.1 Metadaten

Autoren	Christine Natschläger
Veröffentlichungsjahr	2011
Veröffentlichungsform	Conference
Universität/Unternehmen	Software Competence Center Hagenberg
Link	Springer Verlag, [24]
Veröffentlichung	Database and Expert Systems Applications
Anwendungsdomäne	General
Klasse	Visualization (Deontic Logic)
Anzahl neuer Symbole	-
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Erweiterung verständlich erklärt. - Einlesen in Deontische Logik notwendig. - Kein Metamodell.
Unterstützte Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Nein
BPMN Version	2.0
Keywords	Deontic logic, Graph transformation, Color Extension, Modality, Readability, Improve Readability, Reduce Complexity

2.4.24.2 Beschreibung

In großen Projekten kann ein BPMN Modell schnell unübersichtlich werden. Wenn dazu noch mehrere alternative Flüsse existieren, dann wird es schwer verständlicher und es besteht ein größerer Aufwand, sich ein umfassendes Wissen des Prozesses anzueignen. Eine alternative dazu bietet diese Erweiterung. Sie fügt Deontische Logik¹⁴ hinzu, die dafür sorgen soll, dass

¹⁴<http://plato.stanford.edu/entries/mally-deontic/>

in einem BPMN Diagramm, zur besseren Lesbarkeit die strukturelle Komplexität reduziert wird. Das bedeutet, wenn ein Fluss aufgeteilt wird, z.B. durch ein Gateway, soll dies durch deontische Logik ersetzt werden, um schnell und direkt ablesen zu können, welcher Fluss verpflichtend, optional oder eine Alternative ist. Dies geschieht durch die Transformation von BPMN zu Deontic BPMN.

2.4.25 Erweiterung der BPMN zur Spezifikation von Prozessmodellen für Cross-Device Anwendungen

2.4.25.1 Metadaten

Autoren	Markus Kuhn
Veröffentlichungsjahr	2014
Veröffentlichungsform	Bachelorthesis
Universität/Unternehmen	Universität Paderborn
Link	Markus Kuhn (Privat), [16]
Veröffentlichung	-
Anwendungsdomäne	Information domain
Klasse	Cross-Device
Anzahl neuer Symbole	2
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Metamodell ausführlich beschrieben. + Symbole ausführlich beschrieben. + Vier unterschiedliche Anwendungsszenarien.
Unterstützte Modellierungsprogramme	Camunda Modeler ¹⁵
BPMN 2.0 Regelkonform	Ja
BPMN Version	2.0
Keywords	Cross-Device, Symbol Extension, Informationssystem, Information, Computer Science

¹⁵ <https://camunda.org/download/modeler/>

2.4.25.2 Beschreibung

Cross-Device Anwendungen sind nicht mit den Core BPMN Elementen modellierbar. Bei ihnen wird ein Prozess auf einem anderen Gerät für seine Ausführung übertragen. Das bedeutet, dass das Gerät kurz gewechselt wird. Ein Beispiel könnte hierfür ein Ticketautomat der Bahn sein, welcher bar oder EC-Karten Bezahlung akzeptiert. Wenn der Kunde weder

eine EC-Karte noch genügend Bargeld zur Hand hat, so kann er Wahlweise auch mit dem Smartphone bezahlen. Hierfür muss der Zahlungsprozess auf das Smartphone ausgelagert werden und nach erfolgreichem Abschließen der Bezahlung muss die Anwendung weiter auf dem lokalen Ticketautomaten fortgeführt werden. Diese Erweiterung führt dafür zwei neue Symbole ein, welche diese Anwendung modellierbar machen.

2.4.26 BPMNt: A BPMN extension for specifying software process tailoring

2.4.26.1 Metadaten

Autoren	Raquel M. Pillat ¹ , Toacy C. Oliveira ¹ , Paulo S.C. Alencar ² , Donald D. Cowan ²
Veröffentlichungsform	Article
Veröffentlichungsjahr	2014
Universität/Unternehmen	Federal University of Rio de Janeiro ¹ , University of Waterloo ²
Link	Science Direct, [27]
Veröffentlichung	Information and Software Technology
Anwendungsdomäne	Information domain
Klasse	Process Tailoring
Anzahl neuer Symbole	-
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Gut erklärtes Beispiel + Relation von SPEM 2.0 und BPMN 2.0 dargestellt - Verständnis des SPEM 2.0 Tailoring Mechanismus nötig. - Nicht ausführbar, kann in großen Projekten unübersichtlich werden.
Unterstützte Modellierungsprogramme	MDT/BPMN2 Project ¹⁶
BPMN 2.0 Regelkonform	Ja
BPMN Version	2.0
Keywords	Software Process Tailoring, Tailoring, SPEM Alternative, SPEM, specialization, SDP, Software Development Processes

¹⁶ <https://wiki.eclipse.org/MDT/BPMN2>

2.4.26.2 Beschreibung

Diese Erweiterung ermöglicht das Process Tailoring in BPMN. Dafür wird die schon bekannte Process Tailoring Methode aus SPEM 2.0, als eine Erweiterung in BPMN 2.0 implementiert. Bei einem Process Tailoring wird ein schon bekanntes Vorgehensmodell in Form eines BPMN Modell hergenommen. Dieses wird dann durch bestimmte Tailoring Mechanismen für ein spezielles Problem oder Team zugeschnitten. Mit Vorgehensmodell ist hier ein Prozess gemeint, wie bei einem Projekt vorgegangen werden soll. Der Vorteil liegt ganz klar in der Wiederverwendbarkeit, denn sonst müsste man für zwei verschiedene Teams, z.B. ein großes Team, das eine hierarchische Struktur benötigt und ein kleines Team das auf einen agilen Ansatz setzt, ein Vorgehensmodell komplett neu entwickeln.

2.4.27 BPDMMN: A Conservative Extension of BPMN with Enhanced Data Representation Capabilities

2.4.27.1 Metadaten

Autoren	Mateo Magnani, Danilo Montesi
Veröffentlichungsform	Article
Veröffentlichungsjahr	2009
Universität/Unternehmen	University of Bologna
Link	arXiv.org, [18]
Veröffentlichung	arXiv.org
Anwendungsdomäne	Information domain
Klasse	Data
Anzahl neuer Symbole	-
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Abstraktion beschreibt, was die Erweiterung kann und was nicht. + Verschiedene Beispiele vorhanden. + Es wird detailliert beschrieben was erweitert wird. - Kein Metamodell. - Beispiele sehr kurz erklärt.
Unterstützte Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Nein
BPMN Version	1.2
Keywords	Data Store, Symbol Extension, Old, Store, Dynamic, Data Mapping, Symbol Extension

2.4.27.2 Beschreibung

Diese Erweiterung ist für die BPMN Version 1.2 gedacht. Hier wird beschrieben das in dieser Version, BPMN einige Probleme hat, Daten darzustellen. In dieser Version waren Daten und Informationsmodelle nicht Teil der Notation, so konnten die Daten zwar dargestellt werden, jedoch waren die Datenflüsse nicht Teil des Business Process Diagramms, genau so wenig konnten die Datenobjekte den Prozessfluss direkt beeinflussen. Die Autoren der Erweiterung sind da anderer Meinung, denn die Daten sollen direkt im Business Process

dargestellt werden so ist schnell und klar ersichtlich wer die Daten benötigt und wer diese in was transformiert. Es wird um verschiedene neue Symbole erweitert, werden aber nicht konkret benannt. So existieren neue dynamische Objekte, deren Symbolik jeder selbst bestimmen kann wenn er möchte. Die Erstellung dieser neuen Symbolik wird durch zwei konkrete Vorgehensweisen definiert. Des Weiteren werden auch Data Flows, also Datenströme hinzugefügt. Dabei wird unter expliziten, impliziten, so wie Data Mapping Ströme unterschieden.

2.4.28 Modelling Concepts For Process Audits - Empirically Grounded Extension Of BPMN

2.4.28.1 Metadaten

Autoren	Niels Müller-Wickop, Martin Schultz, Markus Nüttgens
Veröffentlichungsform	Inproceeding
Veröffentlichungsjahr	2013
Universität/Unternehmen	Universität Hamburg
Link	AIS Electronic Library, [23]
Veröffentlichung	ECIS 2013 Proceedings
Anwendungsdomäne	Business
Klasse	Financial
Anzahl neuer Symbole	6
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Symbole gut erklärt und aufgelistet. + Metamodell vorhanden. + Gut veranschaulichtes Beispiel.
Unterstützte Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Ja
BPMN Version	2.0
Keywords	Audit, Process Audit, Audit Concepts, Financial, Finance, FSLI, Financial Statement Line Items, Symbol Extension

2.4.28.2 Beschreibung

Die Autoren der Erweiterung intervieweten 17 Wirtschaftsprüfer um einen tieferen Einblick in ihre Arbeit zu bekommen und um festzustellen, was ihnen in ihrer Arbeit fehlt. Dabei stellten die Autoren fest, dass die Wirtschaftsprüfer 6 Prüfkonzpte gerne in BPMN als grafisches Symbol hätten. Dabei sind 5 von den 6 genannten Prüfkonzpten, die nach den befragten Wirtschaftsprüfern grafisch dargestellt werden sollen, schon in Modellierungssprachen existent. Nur das sogenannte *Financial Statement Line Items* (FSLI) Konzept existiert hierbei noch nicht. Diese Lücke schließt nun diese Erweiterung. Das Standard BPMN wird nun also mit neuen Gruppen, einem neuen Datenobjekt und einer neuen Text Annotation Erweitert, die sofort ersichtlich machen sollen, welche Aktivitäten eine finanzielle Auswirkung haben. Außerdem wird durch das Benutzen der Erweiterung schnell das Transaktionsvolumen eines jeden Prozesses ersichtlich, so wie der benutzte Account des Prozesses.

2.4.29 A BPMN 2.0 Extension to Define the Ressource Perspective of Business Process Models

2.4.29.1 Metadaten

Autoren	Luis Jesus Ramon Stroppi ¹ , Omar Chiotti ² , Pablo David Villerreal ¹
Veröffentlichungsform	Conference
Veröffentlichungsjahr	2011
Universität/Unternehmen	Universidad Tecnologica Nacional-FRSF ¹ , INGAR-CONICET ²
Link	Departamento de Ciencia da Computacao, [32]
Veröffentlichung	XIV Iberoamerican Conference on Software Engineering
Anwendungsdomäne	General
Klasse	Requirements(Ressource)
Anzahl neuer Symbole	[4]
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Metamodell ausführlich dargestellt und erklärt. + Conclusion Abschnitt der Erweiterung fasst alles perfekt zusammen und hilft beim Verstehen. - Neue Symbole nicht konkret visualisiert. - Beispiel nicht genau genug erklärt.
Unterstütze Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Ja
BPMN Version	2.0
Keywords	Ressource Perspective, Ressource, Symbol Extension, Requirements, Ressource Structure, Work Distribution, Authorization, Task Privileges

2.4.29.2 Beschreibung

Die Autoren beschreiben, dass die Unterstützung der Modellierung aus Sicht der Ressourcen Anforderungen in BPMN 2.0 dürftig umgesetzt ist. Der Schlüssel zu einem effektiven und effizienten Prozess liegt darin, wie die Arbeit auf die Ressourcen verteilt sind. Daher ist es wichtig, dass die Ressourcen Aspekte aus deren Perspektive besser modelliert werden können. Es werden daher in dieser Erweiterung drei Hauptaspekte integriert, *Resource Structure*,

Work Distribution und *Authorization*. Dabei existieren zwei unterschiedliche Diagramme, das Resource Structure Diagramm, das die Ressourcen Struktur aufstellt und diese charakterisiert und klassifiziert, z.B. welche Person in welcher Gruppe arbeitet und welche Qualifikationen diese besitzt. Das Work Distribution Diagramm, visualisiert dann die Verteilung der Arbeit. In einer zukünftigen Arbeit soll ein Model-Driven Ansatz basierend auf dieser Erweiterung entwickelt werden.

2.4.30 BPMN Plus : a modelling language for unstructured business processes

2.4.30.1 Metadaten

Autoren	Zaharah Allah Bukhsh
Veröffentlichungsform	Masterthesis
Veröffentlichungsjahr	2015
Universität/Unternehmen	University of Twente
Link	University of Twente, [1]
Veröffentlichung	University of Twente
Anwendungsdomäne	General
Klasse	Flexibility
Anzahl neuer Symbole	9
Ausführbar	Nein
Lernschwierigkeit	<ul style="list-style-type: none"> + Erweiterung ausführlich erklärt. + Erklärt anhand von Metamodellen. + Beispiel vorhanden. - Neue Symbole mit Standard Symbolen vermischt, schwer zu filtern.
Unterstützte Modellierungsprogramme	-
BPMN 2.0 Regelkonform	Nein
BPMN Version	2.0
Keywords	Unstructured, Symbol Extensions, Knowledge Worker, Flexibility

2.4.30.2 Beschreibung

Jedes BPMN Modell besitzt eine klare strukturierte Definition die in der Design Phase festgelegt wird. Das bedeutet jeder mögliche Fall und Pfad wird konkret modelliert und

ist somit Vorhersehbar. Doch bei unvorhersehbaren kontextabhängigen Modellen ist es schwer oder gar unmöglich in der Design Phase eine genaue Abfolge der Aktionen zu koordinieren. Mit dieser Erweiterung wird diese Lücke in BPMN 2.0 gefüllt. Die Vorteile dieser Erweiterung sind:

- ◇ Einfache Lesbarkeit.
- ◇ Sowohl unstrukturierte als auch strukturierte Modellierung.
- ◇ Eine User Role Notation.
- ◇ Die Möglichkeiten sogenannte Business Rules zu modellieren für ein Prozess Modell.
- ◇ Modellierung der Kollaboration von Aktivitäten.

Kapitel 3

Die grafische Oberfläche

3.1 Einleitung

Der zweite Teil der Bachelorarbeit besteht darin, eine grafische Oberfläche zu erstellen, mit der es möglich sein soll, die zuvor analysierten Erweiterung darzustellen und zu durchsuchen. Dieser Service soll öffentlich zugänglich sein. Das bedeutet, dass es nicht nur Personen aus dem eigenen Uni-Netzwerk erlaubt sein soll diesen Service zu nutzen, sondern auch Personen außerhalb der Universität. Dazu bietet es sich an diese grafische Oberfläche als eine Webseite zu entwickeln. Damit ist gewährleistet, dass jeder mit jedem Gerät Zugriff darauf hat, ohne irgendwelche Zusatzprogramme installieren zu müssen.

In den folgenden Kapiteln wird das Vorgehen der Planung der Webseite näher verdeutlicht. Zur Planung gehört das Erheben von Anforderungen, welche die Webseite besitzen muss, welche Technologien verwendet werden und ein MockUp.

3.2 Verwendete Technologien

Um eine Webseite nach heutigen Standards zu erstellen, werden viele verschiedene Technologien verwendet. Dazu gehört HTML um eine Webseite zu schreiben, sowie CSS (Cascading Style Sheet). Für eine vereinfachte Positionierung der Elemente und zur besseren Aufteilung der Webseite wird die aktuellste Version des Framework Bootstrap¹⁷ (Version 3.3.6) verwendet. Damit die Webseite auf Nutzer Interaktionen reagieren kann, wird außerdem JavaScript benutzt. Das jQuery Framework¹⁸ (Ver. 2.2.0) wird verwendet um einfache Animationen von Elementen oder das Verändern des Zustandes eines Elementes einfacher zu programmieren. Hinzu kommt noch das jQuery Plugin jQueryUI¹⁹ (Ver. 1.11.4), das z.B. verwendet wird, um einige Eingaben wie ein Autocomplete zu erstellen. Für das Speichern der Nutzereingaben wird das Local Storage verwendet, welches sich in jedem Browser befindet. Das Backend wird in PHP (Ver. 5.3.10) geschrieben. Die Datenbank ist eine MySQL Datenbank in der Version 14.14 und das Backend läuft auf einem Apache Server in der Version 2.2.22.

¹⁷<http://getbootstrap.com/>

¹⁸<http://jquery.com/>

¹⁹<http://jqueryui.com/>

3.3 Browser

Die Webseite wird mit den Browsern Google Chrome (Ver. 50.0.2661.102 m) und Firefox (Ver. 47.0) getestet, wobei der Google Chrome Browser der Entwicklungsbrowser ist und somit auf diesem Browser die volle Funktionsfähigkeit garantiert wird.

3.4 Anforderungen an die Webseite

Bevor die Erstellung der Webseite beginnt, müssen verschiedene Anforderungen erhoben werden, um festzulegen, was am Ende der Arbeit als funktionierend zu gelten hat, sowie was die Webseite am Ende alles leisten soll und was nicht. Das schließt die funktionalen sowie die nichtfunktionalen Anforderungen mit ein.

3.4.1 Funktionale Anforderungen

Die Webseite soll die Funktion besitzen, die BPMN Erweiterungen, die im vorherigem Kapitel dieser Bachelorarbeit analysiert wurden, darzustellen und zu durchsuchen, nach den Kriterien die im Abschnitt 2.1 gesetzt wurden. Damit auf dieser Webseite am Ende nicht nur die von mir vorgefertigten 30 BPMN Erweiterungen platz finden, soll eine weitere Anforderung die Erweiterbarkeit dieser Arbeit sein. Das bedeutet, es soll noch ein Mechanismus erstellt werden mit dem es möglich sein soll, weiteren Inhalt auf dieser Seite hinzuzufügen, nach den von mir erstellten Kriterien. Eine weitere Funktion der Webseite soll das nachträgliche Bearbeiten von Extensions sein. In 3.9 wird durch ein Use Case, dass angestrebte funktionale Verhalten der Webseite dargestellt.

3.4.2 Nichtfunktionale Anforderungen

Die nichtfunktionalen Anforderungen sind vielschichtig, deswegen werden diese hier nun aufgelistet und genauer erklärt.

Robustheit

Die Webseite soll robust gegenüber jeder Eingabe und Aktion des Nutzers sein. Das bedeutet, egal was der Nutzer auf der Webseite macht, die Webseite soll richtig und nachvollziehbar für den Nutzer reagieren und keine Fehler oder gar Abstürze verursachen.

Einfach Bedienbar

Da die Seite für jeden zugänglich ist, soll diese einfach und intuitiv für den Nutzer sein. Das bedeutet, alle Buttons oder Eingabefelder sollen für den Nutzer selbsterklärend sein. Es ist wichtig, dass der Nutzer bei keiner Aktion lange überlegen muss, was diese bewirkt. Um dem teilweise entgegenzuwirken, sollen an kritischen Stellen, wie die verschiedenen Suchmethoden, kleine Informationskästchen hinzugefügt werden, die dem Nutzer kurz und prägnant erklärt, was diese Suchmethoden bewirken.

Sicherheit

Die Datenbank soll weder von äußerlich falschen noch von mutwillig schädlichen Eingaben verändert oder gar beschädigt werden.

Look and Feel

Der Nutzer soll sich auf der Seite wohlfühlen. Das soll mit Farben und animierten Zustandsveränderungen, sprich einfachen visuellen Effekten geschehen. Das sorgt dafür, dass der Nutzer visuell angesprochen wird und ihm dadurch beim Arbeiten auf der Seite nicht langweilig wird. Jedoch soll darauf geachtet werden, dass die Effekte und Farben nicht den Rahmen sprengen, sodass der Nutzer nicht durch eine kunterbunte und effektvolle Webseite visuell erschlagen wird. Farben sollen aus diesem Grund so eingesetzt werden, dass der Nutzer diese mit bestimmten Aktionen assoziieren kann. Zum Beispiel kann die Farbe grün verwendet werden, um eine der Suchmöglichkeiten besser zu signalisieren und von den anderen abzutrennen. Des Weiteren sollen Animationen schlicht und nicht übermäßig verwendet werden. Sie sollen zum Beispiel dazu verwendet werden, einfach Zustandsänderungen wie das Hinzufügen oder Entfernen von Suchparametern visuell zu verschönern.

Korrektheit

Die Ergebnisse sollen korrekt und fehlerfrei wiedergegeben werden. Der Nutzer soll von den Ergebnissen nicht irritiert oder falsch informiert werden.

Übersichtlichkeit

Die Webseite soll übersichtlich gestaltet sein. Das bedeutet, es soll nicht alles eng zusammen gesetzt werden, sondern klar, strukturiert und übersichtlich sein. Eine Seite soll nie wirklich überladen wirken, sondern aufgeräumt und schlicht. Der Nutzer soll sofort mit einem Blick erkennen können, was auf der Seite passiert.

3.5 Personenkreis

Der Personenkreis, der am Ende diese Webseite nutzen wird, sind erfahrene BPMN Nutzer. Deswegen soll der Inhalt der Seite fachlich gerichtet und kein Tutorial für unerfahrene Nutzer sein. Außerdem ist ein BPMN Modellierer nicht gleich eine einzige Personengruppe. BPMN Modellierer oder auch erfahrene BPMN Nutzer kommen aus den verschiedensten Arbeitskreisen wie z.B. aus der Informatik, Wirtschaftswesen oder auch aus dem Herstellungsgewerbe. Kurz, ein BPMN Modellierer kommt aus verschiedenen Berufszweigen. Damit jeder sich auf dieser Seite wohlfühlt und er zu seiner Suche ein passendes Ergebnis findet, wird hier festgelegt, dass nur Erweiterungen hinzugefügt werden, die für die Arbeit im Berufsleben Verwendung finden wird.

3.6 Grafische Oberfläche

3.6.1 Suchoberfläche

In Abbildung 3.1 ist die finale grafische Oberfläche der Suchseite zu sehen. Abgebildet sind die 3 verschiedenen Suchmöglichkeiten: Search By Name, welches mit einer grünen Farbe hinterlegt ist, Search By Category, welches mit einer roten Farbe hinterlegt ist und Search By Keyword, welches mit einer blauen Farbe hinterlegt ist. Die farbliche Abtrennung soll dabei helfen, die 3 Suchmöglichkeiten einfacher voneinander zu unterscheiden. Unten rechts sind die beiden Buttons „Search“ und „Reset“. Mit „Search“ wird die Suche angestoßen, wohingegen der „Reset“ Button alle Eingaben die getätigt wurden zurücksetzt. Durch die einzelnen „Reset“ Buttons in den 3 Suchmöglichkeiten, können auch die Werte der einzelnen Suchmöglichkeiten zurückgesetzt werden. In der Search By Category Suche existiert rechts neben der Überschrift, die Schaltfläche „Add Categories“. Diese erlaubt es neue Kategorien dynamisch hinzuzufügen. In der Abbildung 3.1 sind die Kategorien „Year of Publication“ und „University“ dynamisch hinzugefügt worden. Die Kategorien „Class“ und „Domain“ sind immer vorhanden und können nicht entfernt werden. Wenn eine Auswahl oder eine Kategorie entfernt werden soll, muss auf das entsprechende Kreuz geklickt werden.

Bei Search By Keyword sind 2 Keywords exemplarisch hinzugefügt worden. Durch die Eingabe in das Textfeld und durch die Auswahl der vorgeschlagenen Keywords, werden diese Keywords ebenfalls dynamisch zur Webseite und zur Suchauswahl hinzugefügt. Durch das Klicken auf das Kreuzsymbol wird das Keyword wieder entfernt.

3.6.2 Hinzufügen/Bearbeiten Oberfläche

Die beiden Oberflächen zum Hinzufügen und Bearbeiten von Extensions sind identisch, aus diesem Grund wird nur die Oberfläche zum Hinzufügen einer Extension vorgestellt. In Abbildung 3.2 ist die dazugehörige grafische Oberfläche zu sehen. Das Hinzufügen und Bearbeiten wurde im Stile eines Wizards erstellt, um die Übersichtlichkeit der verschiedenen Eingaben zu gewährleisten. Die Kreise zeigen an, wie viele Stufen dieser Wizard besitzt.

BPMN Extension Home Extensions Info Contact

BPMN Extensions Search:

Variant 1: Search by Name

Search by Name

Variant 2: Search by Category

▼ Add Categories

Class: *Security

Domain: *Cloud System

Year of Publication: *University of Stuttgart

University/Company:

Variant 3: Search by Keyword

Search by Keyword

Symbol Extension Time

Search Reset

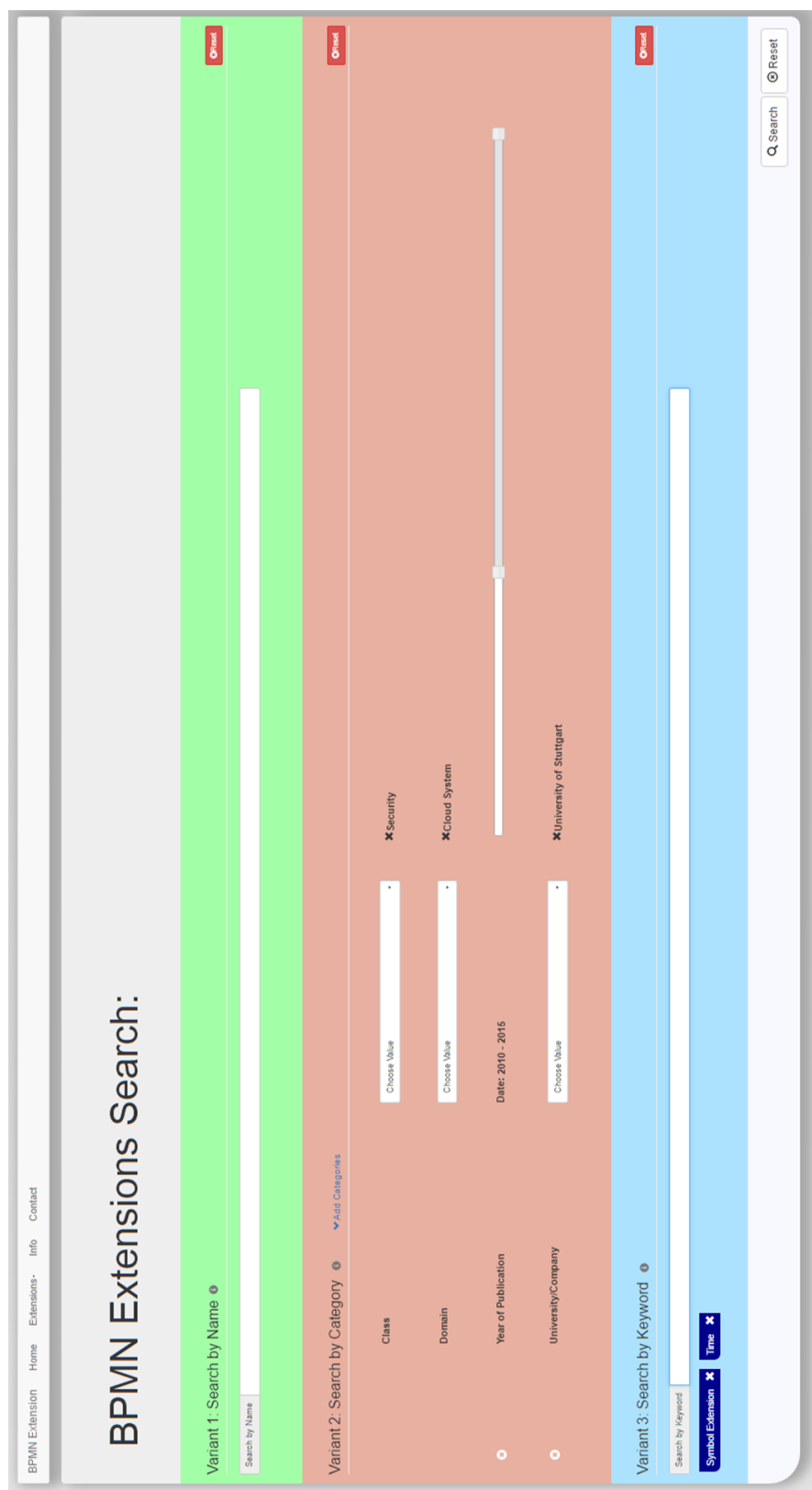


Abbildung 3.1: Suchseite

Außerdem wird durch farbliche Hinterlegungen dem Nutzer dargestellt, an welcher Stelle im Wizard er sich momentan befindet (signalisiert durch die Farbe dunkelgrau) und welche Eingaben schon abgeschlossen wurden (signalisiert durch die Farbe rot). Die Eingaben die in jeder Stufe notwendig sind, sind mit einem (*) Markiert worden. Wenn der Nutzer auf die Schaltfläche „Next“ klickt, wird überprüft ob die Eingaben valide sind. Nur bei einer gültigen Eingabe wird in die nächste Stufe übergegangen. In den folgenden Abbildungen werden die restlichen 6 Eingabeoberflächen vorgestellt.

In Abbildung 3.2 kann der Autor und die dazugehörige Universität bzw. Unternehmen, die an der Extension gearbeitet haben, hinzugefügt werden. Jeder hinzugefügte Eintrag wird unterhalb der Eingabe in der Tabelle hinzugefügt, sodass man im Nachhinein in der Lage ist, diesen wieder zu löschen oder zu bearbeiten. Es muss mindestens ein korrekter Eintrag existieren, bevor es zum nächsten Abschnitt weiter gehen kann.

Wenn die Eingabe korrekt war, geht es in die Abbildung 3.4. Diese zeigt die Eingabe der Klasse und Domäne, mit der dazugehörigen Subklasse und Subdomäne. Auch hier muss mindestens Klasse und Domäne existieren um in den nächsten Abschnitt zu gelangen. Wichtig sei anzumerken, dass die Eingabe eine Autocomplete Funktion besitzt. So ist es für einen Nutzer einfacher schon bestehende Eingaben zu finden.

In Abbildung 3.5 werden die neuen Symbole der Extension hinzugefügt. Nachdem der Upload der Bilddatei erfolgt ist, wird dieses sofort dargestellt. Da mehrere Symbole hinzugefügt werden können ist hier ebenfalls, wie bei der Eingabe des Autors, eine Tabelle, die alle eingegebenen Werte enthält. Das bedeutet die Werte können ebenfalls nochmals bearbeitet oder gelöscht werden.

Das Hinzufügen eines Beispiels ist in Abbildung 3.6 zu sehen. Hier existiert eine Texteingabe, in der das Beispielszenario erklärt wird, sowie die Möglichkeit ein Bild hochzuladen, dass das Beispiel besser beschreiben soll. Der Upload des Bildes ist hierbei Optional.

In Difficulties 3.7 können die sogenannten „Benefits“ und „Difficulties“ hinzugefügt werden. Dies geschieht ebenfalls über Texteingaben, welche im Nachhinein wieder bearbeitet werden können. Hier müssen mindestens ein „Benefit“ wie ein „Difficulty“ hinzugefügt werden.

Als letztes werden die Keywords hinzugefügt. Dieser letzte Zustand ist in 3.8 zu sehen. Wenn mindestens ein Keyword eingegeben wurde, wird durch den nächsten Klick auf „Next“ das Hinzufügen in die Datenbank gestartet. Im Falle von Edit Extension, erscheint noch ein Fenster, dass die Eingabe des Namens, der Email Adresse und den Grund der Überarbeitung von dem Bearbeiter verlangt.

Add new Extension

General Information

University & Author

Class & Domain

Symbol

Example

Difficulties

Keywords

General Information

Title*

Short description* (max 200 Characters)

Form of Publication*

Article

Year of Publication*

Place of Publication*

Link

PDF(max:8MB) ✗ Remove

[Date:unavailable] keine ausgewählt

Executable*

Yes No

BPMN 2.0 Conform*

Yes No


Version*

Number of Symbols*

[← Previous](#)
[Next →](#)

Abbildung 3-2: Hinzufügen einer Extension Oberfläche

Add new Extension



University & Author

Author Information

First name: Surname: URL: E-Mail:

Uni Information

University: URL for University: [+ Add Author and University](#)

Delete/Edit	First name	Surname	University

[← Previous](#) [Next →](#)

Abbildung 3.3: Hinzufügen einer Extension Universität und Unternehmen

Add new Extension

General Information University & Author **Class & Domain** Symbol Example Difficulties Keywords

Class & Domain
Class & Subclass

Class* Subclass

Domain & Subdomain

Domain* Subdomain

← Previous Next →

Abbildung 3-4: Hinzufügen einer Extension Class & Domain

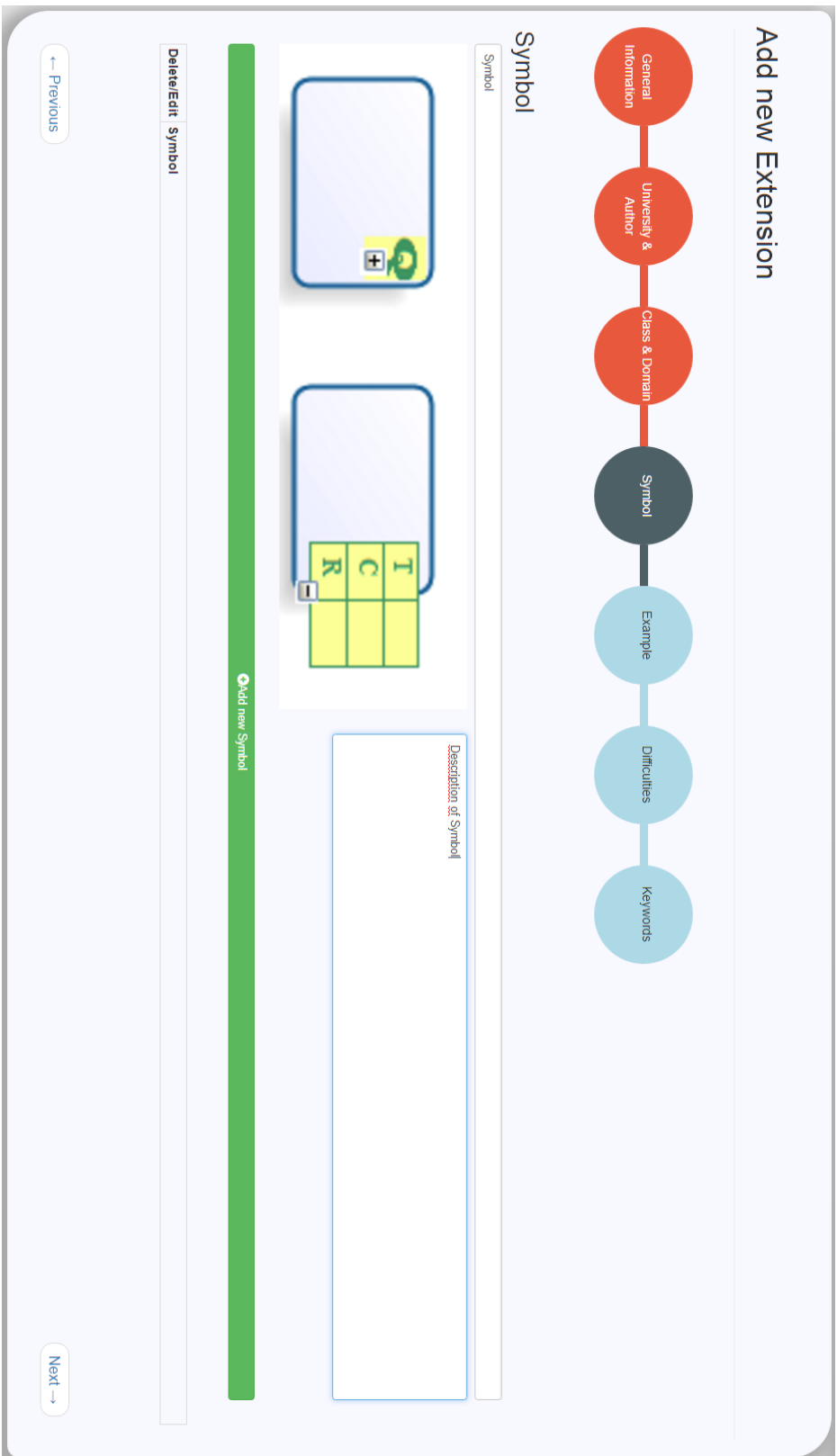


Abbildung 3.5: Hinzufügen einer Extension Symbol

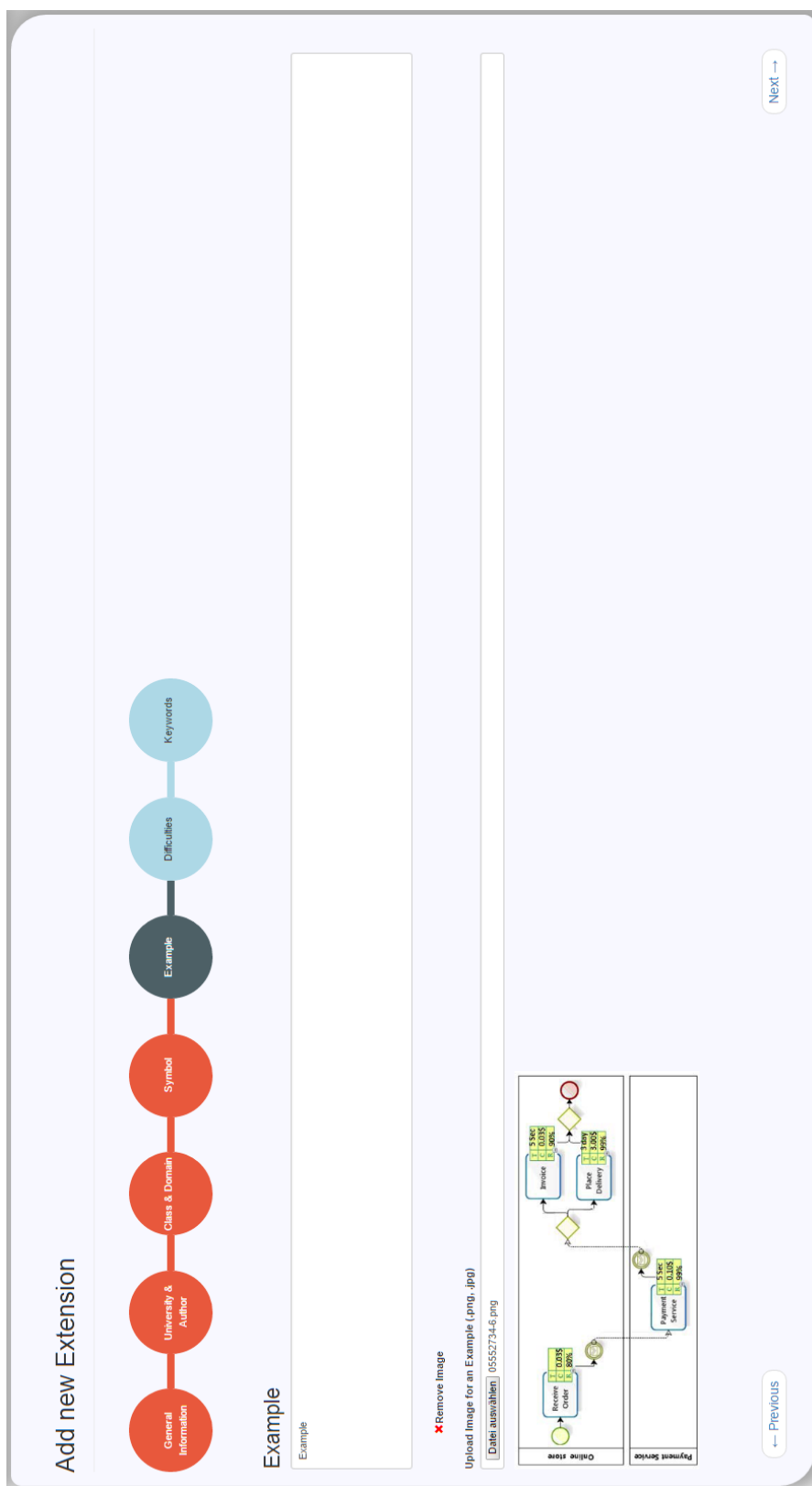


Abbildung 3.6: Hinzufügen einer Extension Example

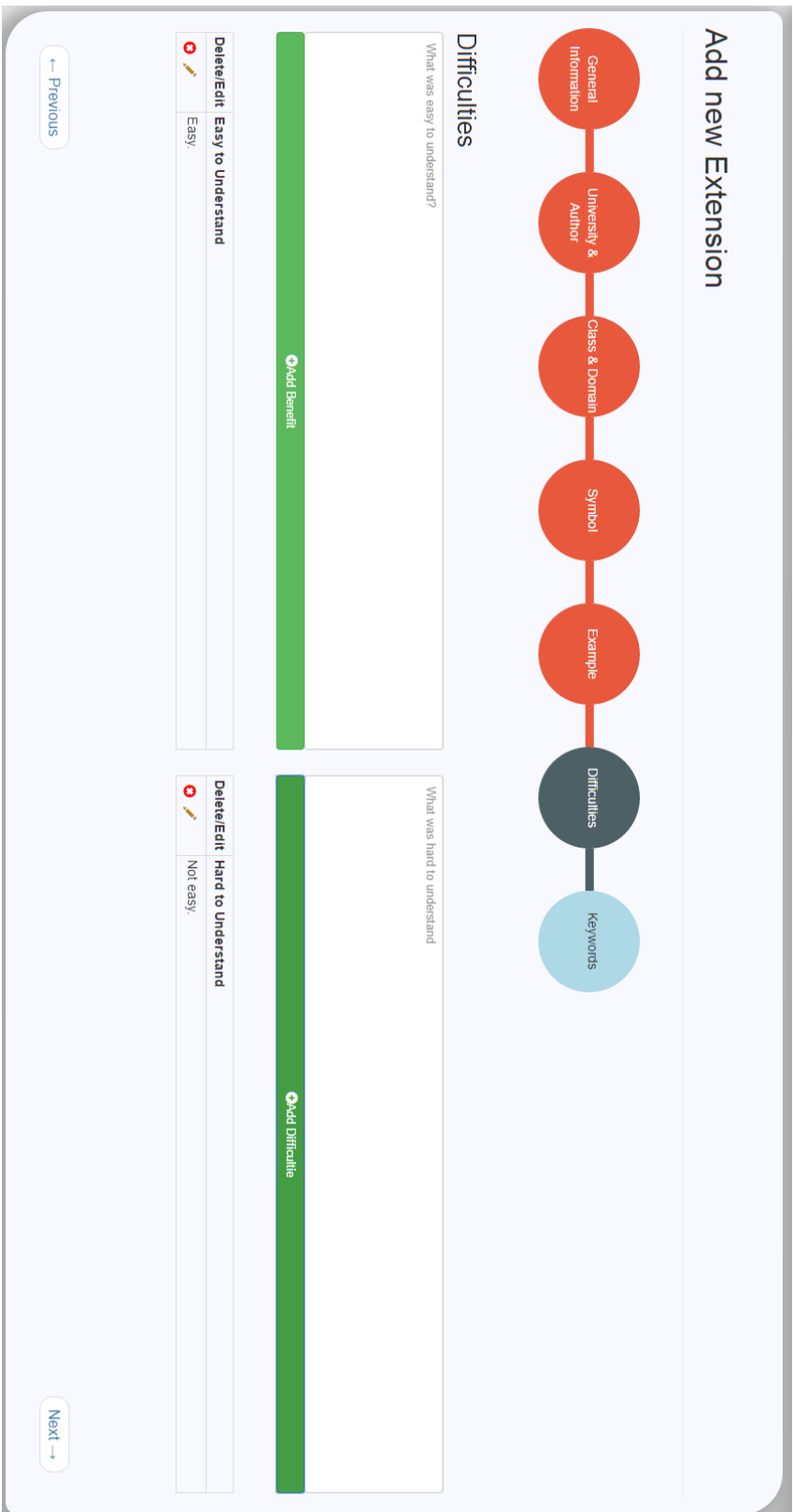


Abbildung 3.7: Hinzufügen einer Extension Difficulties

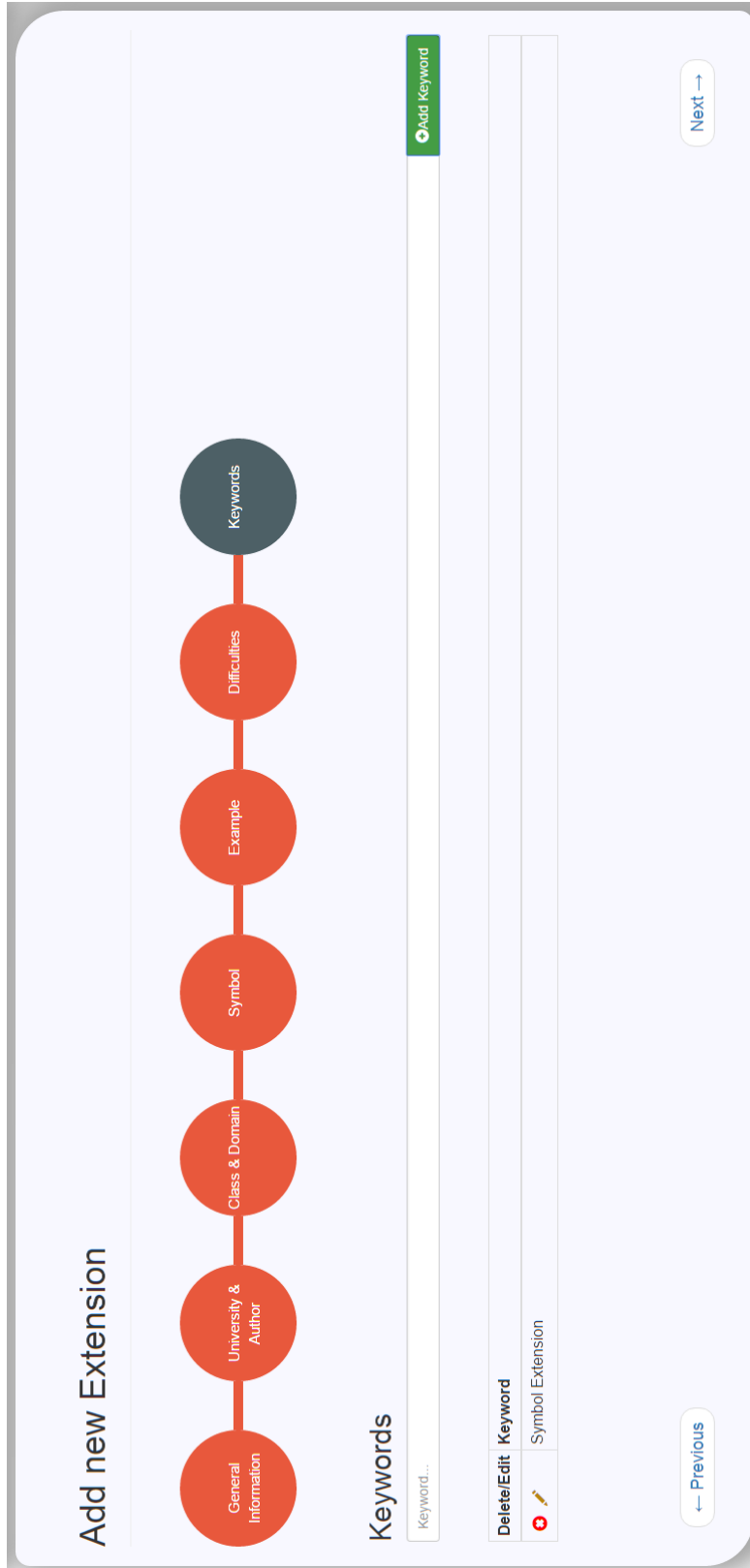


Abbildung 3.8: Hinzufügen einer Extension Keywords

3.7 UseCase

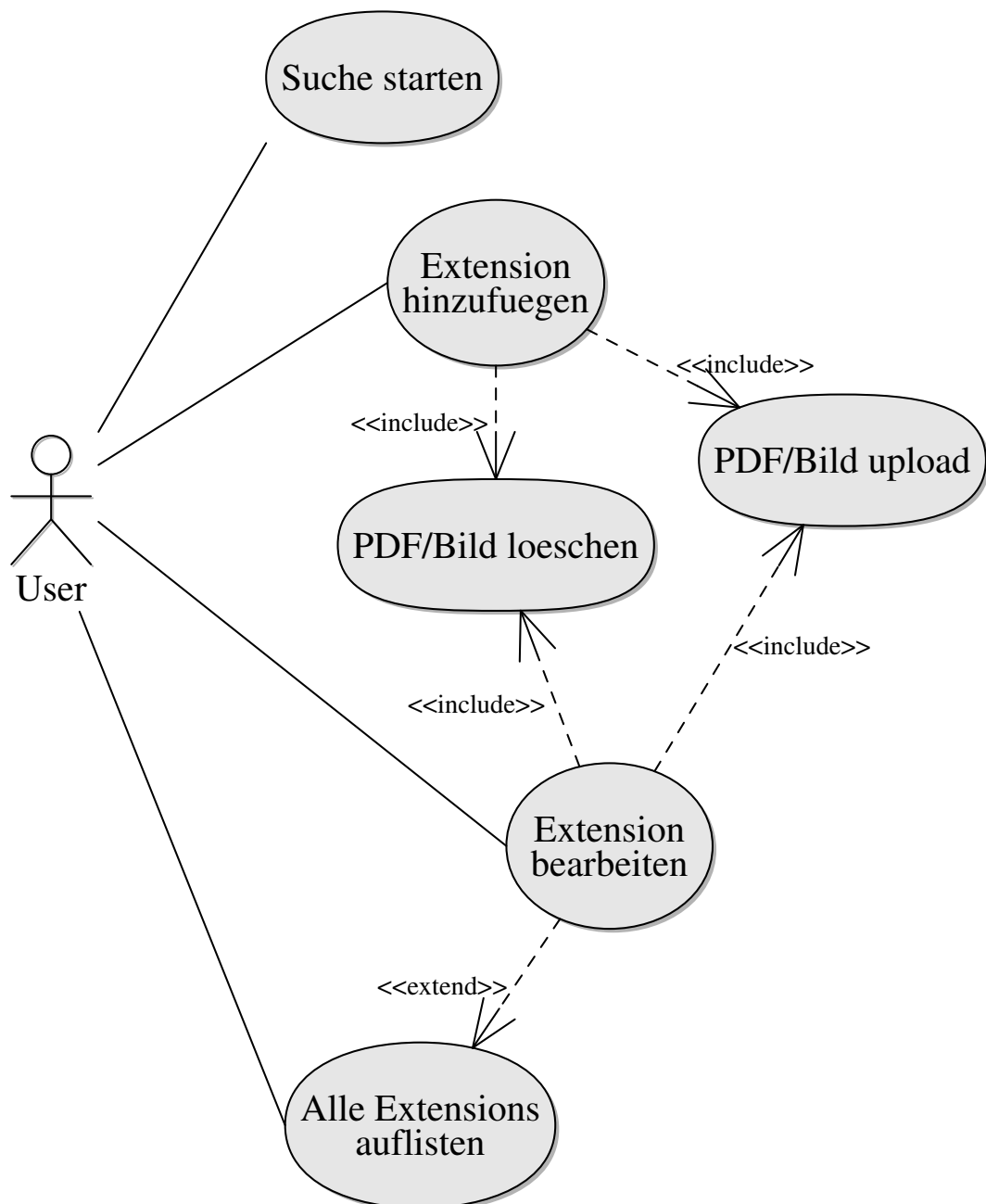


Abbildung 3.9: Use Case der Webseite

Kapitel 4

Datenbank und Backend

4.1 Datenbanksystem

Das Datenbanksystem wird in MySQL²⁰ erstellt. Dabei wird die InnoDB Engine verwendet. Die Datenbank ist in Abbildung 4.1 dargestellt. Danach wird auf die einzelnen Tabellen eingegangen.

4.1.1 EER-Modell

In Abbildung 4.1 ist das EER Diagramm der Datenbank zu sehen, mit allen Tabellen und Beziehungen zueinander. Es gibt insgesamt 14 Tabellen, die dafür sorgen das die Extensions richtig abgebildet werden. Auf die Tabellen und ihre Attribute, sowie diverse Designentscheidungen wird in den Unterkapiteln weiter eingegangen.

²⁰<https://www.mysql.de/>

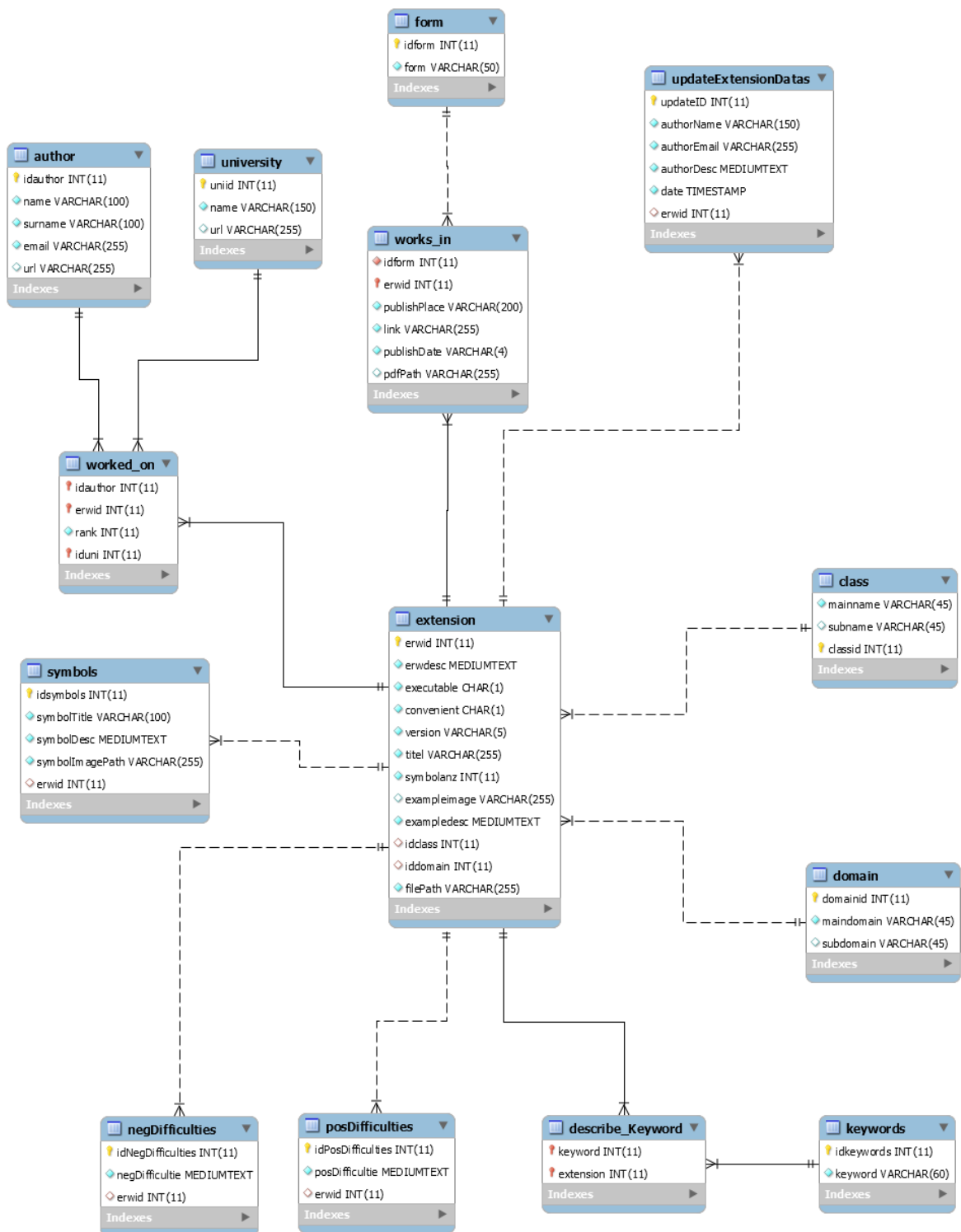


Abbildung 4.1: EER Modell der Datenbank

Extension Tabelle

Die Extension Tabelle ist sozusagen der Dreh- und Angelpunkt des Ganzen. Denn diese Tabelle verbindet alle anderen Tabellen miteinander. In der Extension Tabelle selbst, werden fast alle Metadaten der Extension gespeichert. Dazu gehören der Titel der Extension, die korrespondierende Versionsnummer von BPMN auf die sie entwickelt wurde, sowie ein Beschreibungstext. Alle 3 Attribute sind Strings, wobei *titel* vom Typ VARCHAR ist und einen String von einer maximal Länge von 255 Zeichen speichern kann. *Version* ist ebenfalls ein VARCHAR, aber es sind nur Versionsnummern von maximal 5 Zeichen erlaubt. Jede BPMN Version ist bis jetzt nur maximal 3 Zeichen lang, aber da sich dies in der Zukunft ändern kann, wurden hier 5 Zeichen als maximal Länge festgelegt. Das schließt nämlich 10er Stellen mit ein, z.B. 10.21. Der Beschreibungstext ist vom Datentyp Mediumtext, und kann laut Dokumentation maximal 16.777.215 Zeichen speichern²¹. Da in MySQL kein Boolean Typ mit „True“ oder „False“ existiert, werden bei den einfachen Ja oder Nein Antworten von den Attributen *Conform* und *Executable* ein einziges Zeichen vom Datentyp Char abgespeichert. Dieses besitzt die Werte „y“ für ja oder „n“ für nein.

Die Beispiele werden ebenfalls in die Extension Tabelle hinzugefügt. Da es nur erlaubt ist maximal ein Beispiel hinzuzufügen, wird auf eine separate Tabelle für die Beispiele verzichtet, da es eine 1:1 Beziehung zwischen Example und Extension Tabelle ist. Somit werden die Beispiele in die Extension Tabelle mit hineingezogen. Zu einem Beispiel gehört ein Beschreibungstext, der verpflichtend ist. Dieser Text ist genauso, wie der Beschreibungstext der Extension, vom Datentyp Mediumtext. Das dazugehörige Bild für ein Beispiel ist optional. Von einem Bild wird nur der Pfad gespeichert, wo es sich auf dem Server befindet. Auf Datentypen wie Blob, die komplette Binäre Dateien abspeichern, wird verzichtet.

Das Attribut *filePath* ist ebenfalls dazu da, einen Pfad zu einer Datei abzuspeichern. Diese Datei ist genauer gesagt die HTML Ergebnisseite zu der Extension, die automatisch beim Hinzufügen oder Bearbeiten generiert wird. Zuletzt kommen noch die Fremdschlüssel von Class, mit dem Attribut *idclass* und Domain, mit dem Attribut *iddomain*.

Der Primärschlüssel der Tabelle bildet das Attribut *erwid*. Dieses wird beim Hinzufügen durch AUTO_INCREMENT automatisch gesetzt. Es wird hier AUTO_INCREMENT verwendet, wie bei allen anderen Tabellen auch, da kein anderes eindeutiges Attribut existiert. Der Titel einer Extension war in der Überlegung als Primärschlüssel zu dienen, wurde aber verworfen, da es im späteren Verlauf möglich sein soll den Titel nachträglich zu bearbeiten. Somit ist dieser nicht konsistent.

Class & Domain Tabelle

Die Tabellen Class und Domain unterscheiden sich nur minimal voneinander. Die Tabelle Class besitzt das Attribut *mainname*. Darin wird die Klasse bzw. die Oberklasse abgespeichert. Außerdem kommt noch das Attribut *subname* hinzu, das die Subklasse repräsentieren soll. Beide Attribute sind vom Typ VARCHAR und können maximal 45 Zeichen abspeichern.

²¹http://www.w3schools.com/sql/sql_datatypes.asp

Die Domain Tabelle ist analog zu der Class Tabelle, bis auf den Namensunterschied. Die Hauptdomain wird hier *maindomain* genannt und die Subdomain wird hier *subdomain* genannt. Die Datentypen sind ebenfalls identisch zu der Class Tabelle. Die Class und Domain Tabelle sind in einer 1:n Beziehung zu der Extension Tabelle. Denn eine Extension besitzt genau eine Klasse und eine Domain. Jedoch können mehrere Extensions die selbe Klasse oder Domain besitzen.

Works_in & Form Tabelle

Die nächsten beiden Tabellen speichern wieder Metadaten der Extension ab, genauer gesagt die Veröffentlichungsdaten. Die zu speichernden Daten dafür sind, die PDF zu der Extension, der Link im Internet zu der Extension, sowie Veröffentlichungsjahr und Ort. Die Form Tabelle besitzt schon alle Werte, nämlich alle möglichen Veröffentlichungsformen, wie z.B. Article, Bachelorthesis oder Book. Die Relation zu Form, works_in und Extension ist wie folgt. Die works_in Tabelle ist die Verbindungstabelle zu Extension und Form. Da die Daten der Relation zwischen Extension und Form genauer beschrieben werden müssen, wird die works_in Tabelle gebraucht. Deswegen ist die Relation wie folgt. Ein Form kommt mehrmals, also n-mal in einer works_in Tabelle vor. Mehrere Extension besitzen die gleiche Form und kommen höchstens einmal in der works_in Tabelle vor.

Die Datentypen sind bis auf den Primärschlüssel von der Form Tabelle nur Strings. Genauer sind die Attribute *publishPlace* vom Typ VARCHAR, die es erlauben 200 Zeichen maximal abzuspeichern und *link* das ebenfalls vom Typ VARCHAR ist und erlaubt 255 Zeichen abzuspeichern. Das Jahr wird ebenfalls als String abgespeichert, da es angenehmer ist ein String zu handhaben, als einen Integer. Dieser besteht aus einem VARCHAR, mit einer maximalen Länge von 4 Zeichen. Der Pfad zur PDF wird ebenfalls in einem VARCHAR der maximalen Länge 255 abgespeichert. Aufgrund der Beziehung zu Extension und Form, sind beide Primärschlüssel der Tabellen Extension und Form, Fremdschlüsseln in der Tabelle works_in. Der Extension Primärschlüssel ist zugleich auch noch ein Primärschlüssel in der works_in Tabelle, da dieser in einer n Beziehung zu der Form Tabelle steht.

Symbols Tabelle

In der Symbols Tabelle werden die Symbole einer jeden Extension abgespeichert. Dabei wird der Titel des Symbols, der Beschreibungstext und das dazugehörige Bild, welches das neue Symbol abbildet, abgespeichert. Die Datentypen von den Attributen *symbolTitle* und *symbolImagePath* sind jeweils ein VARCHAR der Länge 255. Der Beschreibungstext wird in dem Attribut *symbolDesc* als Mediumtext abgespeichert. Der Primärschlüssel ist *idsymbols* und ist vom Typ Integer und wird automatisch durch AUTO_INCREMENT gesetzt. Die Symbols Tabelle besitzt nur eine Beziehung zu der Extension Tabelle. Diese ist eine 1:n Beziehung, denn eine Extension besitzt mehrere Symbole.

Keywords & describe_keywords Tabelle

Die nächsten beiden Tabellen sind für die Keywords einer Extension verantwortlich. Die Keywords werden in der Keywords Tabelle abgespeichert, im Attribut *keyword*, vom Typ VARCHAR. Die Länge eines Keywords kann maximal 60 Zeichen betragen. Da die Relation zwischen Keywords Tabelle und Extension eine n:m Beziehung ist, muss eine zwischen Tabelle her, um die n:m Beziehung aufzulösen. In diesem Fall ist es die describe_keywords Tabelle. Diese führt die beiden Tabellen zusammen, indem sie die beiden Primärschlüssel der Tabellen abspeichert.

posDifficulties & negDifficulties Tabelle

Diese beiden Tabellen speichern die Schwierigkeiten und die Erleichterungen beim Erlernen einer Extension ab. Jede Extension besitzt mindestens ein bis n-mal viele Schwierigkeiten und Erleichterungen. Beide Tabellen besitzen ein Attribut, das die Eingabe des Users im Datentyp Mediumtext abspeichert. Der Primärschlüssel von beiden wird auch durch ein AUTO_INCREMENT erhöht.

Author, University & worked_on Tabelle

Um Autor, Universität und Extension in der Datenbank abzuspeichern, wird sich für eine ternäre Beziehung zwischen den 3 Tabellen entschieden. Der Grund dafür ist wie folgt. Die Beziehung zwischen Autor und Universität ist eine n:m Beziehung. Der Grund dafür ist recht einfach. Mehrere Autoren arbeiten an einer Universität. Doch da ein Autor die Universität auch wechseln kann und es nicht gewollt ist, das die Daten einer anderen Erweiterung sich mit ändern, gibt es eine n:m Beziehung zu Autor und Universität. Die ternäre Beziehung entsteht, da die Erweiterung durch die Universität gefunden werden kann. Damit nicht, um die Universität herauszufinden, erst nachgefragt werden muss, welche Autoren mitgearbeitet haben und an welcher Universität diese arbeiten, ist es einfacher gleich die Universität in eine Beziehung zu der Erweiterung zu setzen. Aus diesem Grund entsteht diese ternäre Beziehung.

Nun zu den einzelnen Tabellen. Die Author Tabelle speichert den Vornamen, Nachnamen, Email Adresse als auch die Webseite eines Autors ab. Vorname und Nachname werden von den Attributen *Forname* und *Surname* repräsentiert. Diese beiden Attribute besitzen den Datentyp VARCHAR und können mit jeweils 100 Zeichen befüllt werden. Die URL kann mit 255 Zeichen befüllt werden, genau wie die Email Adresse.

Eine Universität besitzt sowohl einen Namen als auch eine Webseite. Der Name wird durch *name* repräsentiert, die Webseite durch *url*. Beide Attribute besitzen den Datentyp VARCHAR, wobei der Name nur 150 Zeichen abspeichern kann im Gegensatz zur *url*, diese kann 255 Zeichen abspeichern.

In der worked_on Tabelle werden die 3 Tabellen Extension, Author und University zusammengefügt. Jede Verknüpfung dieser 3 Tabellen besitzt noch das zusätzliche Attribut *rank*. Dieses beschreibt die Reihenfolge der Autoren wie sie in der Arbeit genannt wurden, da

dies bei wissenschaftlichen Ausarbeitungen nicht willkürlich ist. Dies ist für die spätere Darstellung von Bedeutung. Der Datentyp von *rank* ist ein Integer. Die Zählung beginnt bei 0, wobei ein Autor bei einem *rank* von 0 an erster Stelle steht. Je höher die Zahl desto weiter hinten steht ein Autor.

updateExtensionDatas Tabelle

In dieser Tabelle wird ein Changelog einer Extension gespeichert. Ein Changelog wird angelegt, sobald eine Extension einmal bearbeitet wurde. Die `updateExtensionDatas` Tabelle besitzt die Attribute `authorName`, `authorEmail` und `authorDesc`. Wobei `authorName` vom Typ VARCHAR ist und maximal 150 Zeichen abspeichern kann. Das Attribut `authorEmail` ist ebenfalls vom Typ VARCHAR und kann 255 Zeichen maximal abspeichern. Die Beschreibung für den Changelog wird im Attribut `authorDesc` gespeichert, welches vom Typ Mediumtext ist. Durch das Attribut `date` wird der Zeitpunkt der Bearbeitung gespeichert. Dieses wird genauer als `TIMESTAMP` abgespeichert. Das setzen dieses Attributes geschieht automatisch durch MySQL.

4.1.2 Views

Es wird in manchen Datenbankabfragen die sogenannten Views von MySQL verwendet. Views sind, nach dem MySQL Referenz Handbuch²², abgespeicherte Abfragen, die wenn man sie ausführt eine Virtuelle Tabelle zurückliefern. Views zu benutzen gibt in diesem Projekt 2 Vorteile. Der erste Vorteil ist die bessere Übersichtlichkeit bei SQL Abfragen, vor allem bei verschachtelten Abfragen. Der zweite Vorteil liegt in der Wiederverwertbarkeit. Denn es werden z.B. manche Abfragen öfters an mehreren Stellen verwendet. Insgesamt existieren 15 Views.

4.2 JavaScript

JavaScript wird verwendet, um clientseitige Aktionen auszuführen. Dazu gehört z.B. das Validieren von Eingaben, das Abspeichern in den LocalStorage, sowie visuelle Effekte. Um dies umzusetzen wird für jede HTML Seite mindestens eine JavaScript Datei erstellt, die dort als Controller für diese Seite fungieren soll. Folgend sind die zusammengehörigen JavaScript Dateien aufgelistet, so wie die HTML Datei, in der sie eingebunden sind.

- ◇ search.html
 - sbc-controller.js
 - sbn-controller.js

²²<http://dev.mysql.com/doc/refman/5.7/en/views.html>

- sbk-controller.js
- serachSideController.js
- ◇ addExtension.html
 - addExtensionController.js
 - addExtensionGeneral.js
 - addExtensionUniAut.js
 - addExtensionClassDomainController.js
 - addExtensionSymbolController.js
 - addExtensionExampleController.js
 - addExtensionDifficultiesController.js
 - addExtensionKeywordController.js
- ◇ editExtension.html
 - editExtensionController.js
 - editExtensionGeneral.js
 - editExtensionUniAut.js
 - editExtensionClassDomainController.js
 - editExtensionSymbolController.js
 - editExtensionExampleController.js
 - editExtensionDifficultiesController.js
 - editExtensionKeywordController.js

Es wird hier nicht tiefer auf die einzelnen Dateien eingegangen. Jedoch möchte ich auf die allgemeine Grundstruktur eines jeden Controllers kurz eingehen sowie erklären, wann genau ein Controller verwendet wird. Ein Controller wird verwendet, falls auf der Webseite clientseitige Aktionen verwaltet werden müssen. Dabei ist ein Controller für eine Seite nicht ausreichend, da dieser ansonsten viel Code besitzen würde und übersichtlich werden würde. Daher gibt es genau einen Hauptcontroller für jede Seite, der die einzelnen Controller, die sich mit einzelnen Abschnitten auf der Webseite beschäftigen, verwaltet. Als Beispiel wird der Controller `editExtensionController.js`, der auf der Webseite `editExtension.html` verwendet wird, betrachtet. Dieser verwaltet die komplette Seite indem es überprüft, in welchem Status der User sich momentan im Wizard befindet und koordiniert, welcher Controller für die Validierung zuständig ist. Jeder einzelne Controller kümmert sich dann selbst um die Validierung der Eingaben, sowie um die Darstellung und Manipulation der Elemente, für die es Zuständig ist. Jeder Controller hat die folgende Struktur:

Listing 4.1: JavaScript Controller Struktur

```
1 controller = function () {  
2     return {  
3         init: function (page) {  
4             }  
5     }  
6 } ();
```

Der Controller wird von außerhalb mit der Methode `init` aufgerufen und somit instanziiert. In `init` befindet sich der Code, der für die Manipulierungen und Verwaltungen auf der Webseite verantwortlich ist. Funktionen, die nicht von außen aufgerufen werden sollen, werden vor dem Beginn des `return` Statements platziert. Funktionen die von außerhalb aufgerufen werden sollen werden nach der geschlossenen geschweiften Klammer von `return` platziert. Die Verwaltung des `LocalStorage` wird über den Controller `task-webstorage.js` übernommen. Dessen Implementierung ist aus [7] entnommen.

4.2.1 Validierung

Wie schon erwähnt kümmert sich JavaScript um die komplette Validierung der Eingaben. Validierungen die JavaScript tätigt sind z.B. Überprüfungen von Jahreseingaben, sowie ob der Nutzer Werte in notwendige Feldern eingegeben hat. Die Eingabe einer URL, Email oder der Versionsnummer von BPMN werden ebenfalls auf Korrektheit validiert. Die meisten Eingaben werden mit Regular Expressions überprüft.

4.3 Erstellung der Kategorien der Suchseite

Die Inhalte der Suchkategorien werden dynamisch vom Server erstellt. Dies wird durch die Datei `fillCategoryController.php` bewerkstelligt. Dieses PHP Script wird vom User aufgerufen, sobald dieser auf der `search.html` Seite an der Kategorienauswahl etwas ändert. Dies geschieht, wenn der Nutzer entweder;

- ◇ eine neue Kategorie zur Auswahl hinzugefügt,
- ◇ eine bestehende Kategorie verändert (Hinzufügen oder Entfernen eines Items)
- ◇ oder die Seite neu lädt oder aufruft.

Es werden dann Comboboxen, welche alle Werte einer Kategorie enthalten, generiert, sowie die Liste aller schon vorhandenen Items. Diese werden dann zur Suchseite in der Sparte `Seach By Category` hinzugefügt. Bei der Kategorie „Year of Publication“ wird jedoch keine Combobox generiert, sondern ein Slider. Da es möglich sein soll Zeitspannen einzugeben. Die Generierung der Comboboxen und des Sliders sind trivial, sodass lediglich ein Beispiel einer Kategorie vorgestellt wird. Bevor es zu sehr ins Detail geht, wird die Struktur der Generierung erklärt.

Das PHP Script `fillCategoryController.php` ist ein Controller. Dieser überprüft welche Kategorien vom User auf der Webseite hinzugefügt wurden, zu den schon fest integrierten Kategorien Class und Domain. Dabei werden von der Webseite der Name der Kategorie, sowie schon alle ausgewählten Elemente dieser Kategorie zum Server geschickt. Wenn die Kategorien von dem Controller PHP Script identifiziert wurden, werden nach und nach die jeweiligen spezifischen PHP Scripts aufgerufen, die für die Befüllung einer spezifischen Kategorie zuständig sind. Die einzelnen Scripts die ausgeführt werden bauen die Comboboxen mit den Inhalten, wobei darauf geachtet wird, das die schon ausgewählten Werte nicht nochmals in der Combobox auftreten. Die Aufteilung der Aufgaben in einzelne Scripts erlaubt eine besser Übersichtlichkeit, sowie eine Erweiterbarkeit, falls sich in der Zukunft mal entschieden wird, eine neue Kategorie zur Auswahl hinzuzufügen. Die Identifizierung läuft über eine Switch-Case Anweisung, bei der die Namen einer Kategorie überprüft werden. Bei einem Treffer, wird das entsprechende PHP Script für die Kategorie inkludiert. Aufgerufen wird der Controller über die JavaScript Datei `sbc-controller`. Ein Beispiel für eine spezifische Befüllung einer Kategorie, wird anhand der Kategorie Universität gezeigt.

Listing 4.2: `universityCategory.php`

```

1 <?php
2
3 $itemList = $array['values'];
4
5 $list = "<div class='col-md-7'>";
6 $whereClause = "";
7 foreach ($itemList as $item)
8 {
9     $list .= generateList($item);
10    if (strlen($whereClause) == 0){
11        $whereClause .= "WHERE id != ". $item['id'];
12    } else
13    {
14        $whereClause .= " AND id != ".$item['id'];
15    }
16
17 }
18
19 $sql_query = "SELECT * FROM final.uniCategory ".$whereClause;
20
21 $queryResult = $conn->query($sql_query);
22
23 $htmlResult .= dynamicEntry($array['id'], "University/Company",
24     $queryResult).$list."</div></div>";
25 mysqli_free_result($queryResult);
26
27 ?>

```

In 4.2 ist der Code zur Generierung der Universität Combobox zu sehen. Dieser Code soll exemplarisch die Mechanik der generellen Befüllung näher bringen. In Zeile 1 werden die

gesamten Werte, die der Nutzer auf der Webseite ausgewählt hat, der Variable *\$itemList* zugewiesen. Die darauffolgende Variable *\$list* beinhaltet am Ende einen String mit dem HTML Code der für die Auflistung der schon bekannten Werte zuständig ist. In der `foreach` Schleife wird die Variable *\$itemList* durchlaufen und dabei wird auch gleichzeitig die *\$list* Variable befüllt, sowie die Variable *\$whereClause*, welche eine SQL Abfrage generiert. Diese SQL Abfrage ist wichtig, da diese die schon vorhandenen Werte ausfiltert. In den Zeilen 19 und 21 wird die Datenbank kontaktiert. Durch die Funktion `dynamicEntry` wird dann letztendlich die Combobox generiert und dem String *\$htmlResult* angehängt, sowie die *\$list* Variable, die in der `foreach` Schleife durch die Methode `generateList` generiert wird. Die Funktion `dynamicEntry` generiert einen kompletten Eintrag in HTML Code für eine Kategorie. Die Funktion `generateList` generiert eine Liste aus der Variable *\$itemList*.

4.4 Suchalgorithmus

Um die Extensions dem Nutzer im späteren Verlauf zur Verfügung zu stellen, muss er diese durchsuchen können. Eine simple Auflistung von allen Extensions wäre dabei nicht Zielführend. Aus diesem Grund wird für das Auffinden von Extensions ein Suchalgorithmus benötigt. Dieser wird in diesem Kapitel erklärt, welches folgend strukturiert ist. Zuerst werden die verwendeten Suchparameter und die unterschiedlichen Suchvarianten vorgestellt. Die Grundlegende Idee des Algorithmus wird im darauffolgenden Kapitel begründet. Danach wird der fertige Algorithmus vorgestellt.

4.4.1 Suchvarianten und Parameter

Es soll zwischen drei Varianten unterschieden werden. Es gibt zum einen die Suchvariante `Search By Name`. Bei dieser Art von Suche wird nach dem Titel einer Extension gesucht. Das bedeutet der Nutzer gibt einen Text oder ein Wort ein, von dem er wissen will, kommt dieser in irgendeiner Form im Titel einer Extension vor. Diese Suchvariante wurde gewählt, da diese Nützlich ist, wenn der Titel einer Extension schon bekannt ist. So kann die Extension direkt und schnell gefunden werden.

Eine andere Suchvariante ist die sogenannte `Search By Category` Suche. Bei dieser Suchvariante kann der Nutzer verschiedene Kategorien hinzufügen, von denen er will, dass die Extensions danach gefiltert werden sollen.

Nun zu den Kategorien aus dieser Suchvariante. Anfangs dieser Arbeit wurde die Tabelle 2.1 gezeigt, welche die Metadaten einer Erweiterung beinhaltet. Aus diesen Daten sind die Kategorien für diese Suchvariante abgeleitet worden. Welche sich davon als Suchparameter geeignet haben sind die folgenden:

Class: Die Oberklasse, in der die Extension angesiedelt ist.

Domain: Die Domain, in der die Extension genutzt werden kann.

Subclass: Die Unterklasse der Extension. Beschreibt die Oberklasse genauer und ergänzt somit Class.

Subdomain: Die Subdomain der Extension. Präzisiert das Anwendungsgebiet und ergänzt somit Domain.

Year of Publication: Das Jahr, in dem die Erweiterung publiziert wurde.

Form of Publication: In welcher Publizierungsart wurde die Erweiterung hinzugefügt.

University/Company: Welche Universitäten oder Unternehmen haben an einer Extension mitgewirkt.

Symbol Size: Beschreibt, wie groß der Umfang der Extension anhand der Anzahl neu hinzugefügter Symbole ist.

Execution: Einfache Ja oder Nein Antworten, ob die Extension für eine Ausführung gedacht ist.

Conform: Einfache Ja oder Nein Antworten, ob die Extension mit den BPMN 2.0 Extension Mechanismus entwickelt wurde.

BPMN Version: Die BPMN Version, für die die Extension entwickelt wurde.

Diese Daten eignen sich am Besten, um nach einer Extension zu suchen. Denn die Kategorien Class, Subclass oder Domain und Subdomain, fassen mehrere Extensions zusammen, die in ähnlichen Bereichen Anwendung finden und man kann somit schnell eine Erweiterung finden, die zu einem Problem zugeschnitten ist. Mit der Kategorie Year of Publication kann man nach der Aktualität einer Extension suchen, sowie mit der Kategorie Version, mit der nach der BPMN Version gefiltert werden kann. Die Form of Publication ist ebenfalls ein wichtiger Suchparameter, denn die meisten Extensions werden, wie schon erwähnt, in wissenschaftlichen Papern publiziert. So lässt sich am Besten unterscheiden, ob z.B. die Arbeit in einem Buch, in einer Konferenz oder als eine Abschlussarbeit veröffentlicht wurde. Nach der Universität bzw. nach einem Unternehmen, die an einer Extension gearbeitet hat, kann ebenfalls gesucht werden. Diese Kategorie wird verwendet, da man vielleicht mehr von der Arbeit einer bestimmten Universität oder Unternehmen sehen möchte. Die Symbol Size Kategorie soll zeigen, wie umfangreich eine Erweiterung ist und ist somit eine hilfreiche Kategorie, wenn man eine Extension haben möchte, die mal mehr oder weniger umfangreich sein soll. Mit Execution soll danach gesucht werden können, ob eine Extension für eine Ausführung gedacht ist und mit Conform soll danach gefiltert werden, ob die Erweiterung mit der von der OMG beschlossenen Erweiterbarkeitsmechanismus in BPMN 2.0, entwickelt wurde oder nicht und ist somit auch eine wichtige Kategorie.

Jede Kategorie besitzt unterschiedlich viele Werte. Der Nutzer wählt dann davon die Werte aus, die zu seiner Suche passen. Alle Werte die ausgewählt werden können, sind aus der Datenbank. Im Datenbankmodell hat man gesehen das Class und Subclass, sowie Domain und Subdomain, eigentlich zusammen gehören. Jedoch werden diese bei der Suche voneinander getrennt. Das hat den Vorteil, das diese bei der Suche nicht voneinander abhängen und das man die Möglichkeit hat, nur nach den Spezialisierungen suchen zu können, unabhängig von der Oberklasse oder Oberdomain.

Die letzte Suchvariante ist die Search By Keyword Suche. Bei dieser Suche gibt der Nutzer Keywords ein. Dabei wird darauf geachtet, dass nur Keywords eingegeben werden können, die schon in der Datenbank existieren. Damit wird gewährleistet, dass der Nutzer garantierte Treffer in der Suche enthält.

Wichtig ist zu erwähnen, dass alle drei Suchvarianten am Ende miteinander kombiniert werden können.

4.4.2 Idee

Da nun die Parameter und die Suchvarianten erklärt wurden, geht es an den Suchalgorithmus selbst. Dieser soll einfach gehalten werden. Das bedeutet der Algorithmus soll die Eingaben des Nutzers betrachten und vergleicht diese mit den schon bestehenden Parameter für jede Extension in der Datenbank. Die erste grundlegende Idee für so einen Algorithmus, war die selbst ausgedachte „Drei Punkte Suche“.

Die Idee der „Drei Punkte Such“ war, alle drei Varianten einzeln zu betrachten. Das bedeutet, es gibt eine Suche für Search By Name, Search By Category und Search By Keyword. Wenn in irgendeiner dieser Varianten ein Suchtreffer für eine Extension existiert, so bekommt diese Extension für diese Variante den Wert „1“. Das würde bedeuten, wenn eine Erweiterung in alle drei Suchvarianten gefunden wird, bekommt diese den Wert „3“ und hätte somit die höchste Relevanz für diese Suche. Jedoch hat diese Suchmethode einige Probleme. Wenn man nur in einer Variante sucht, so haben alle Erweiterungen bei einem Treffer den Wert „1“. Somit wären alle Erweiterungen gleichgestellt und man müsste diese anders voneinander unterscheiden. Bei der Search By Keyword Suche wäre das weit aus schlimmere Problem, das Extensions die nur auf ein Keyword zutreffen, gleichgestellt werden mit einer Extension die vielleicht fünf oder mehr Treffer auf Keywords hätten und somit einen höheren Stellenwert besitzen sollten.

Deswegen ist diese Idee schnell verworfen worden. Aber darauf basierend ist die letztendliche Idee für den trivialen Suchalgorithmus gebildet worden. Anstatt jede einzelne Variante in sich geschlossen zu betrachten, werden alle Eingaben des Nutzers, die einen Treffer erzielen, zusammengezählt. Das gleicht den Nachteil der „Drei Punkte Suche“ aus, indem jeder Treffer einen Punkt gibt. So kann am Ende auch eine Suchrelevanz für die Suche angegeben werden, indem die Anzahl der gesetzten Suchparameter mit den eigentlichen Treffern je Extension ins Verhältnis gesetzt werden. Die komplette Suche soll auch vollständig in einer Datenbank Abfrage erfolgen. Die Abfrage selbst soll je nach eingegebenen Parametern, serverseitig generiert werden.

4.4.3 Algorithmus

Nun, da die Idee des Algorithmus erklärt wurde, geht es hier in die konkrete Implementierung. Die eigentliche Suche soll in einer SQL Abfrage getätigt werden. Die Generierung der Abfrage soll über PHP geschehen. Die komplette Abfrage kann aus Platz gründen nicht

vollständig dargestellt werden. Aus diesem Grund werden wesentlichen Abschnitte, die Wichtig für das Verständnis sind, gezeigt.

Der Grundaufbau der SQL Abfrage ist wie folgt:

Listing 4.3: Suchanfrage

```
1 SELECT ext.titel, ext.erwdesc, class.mainname, class.subname, domain.
   maindomain, domain.subdomain, ext.version, erw AS ErweiterungsID, count
   (erw) AS AnzPerID, ext.filePath FROM (**) AS search, final.extension AS
   ext, final.class AS class, final.domain AS domain WHERE search.erw =
   ext.erwid and ext.iddomain = domain.domainid and ext.idclass = class.
   classid GROUP BY search.erw ORDER BY AnzPerID DESC;
```

Diese Abfrage wird Schritt für Schritt durchgegangen.

Im SELECT Statement werden die Spalten ausgewählt, die von der Datenbank zurückgegeben werden. In diesem Fall soll die Datenbank, sofern diese einen Suchtreffer enthält, den Titel der Erweiterung (`ext.titel`), den Beschreibungstext der Erweiterung (`ext.erwdesc`), den Klassen sowie den Subklassen Namen (`class.mainname`, `class.subname`), den Domain und den Subdomain Namen (`domain.maindomain`, `domain.subdomain`) zurückgeben. Dazu kommt noch die Version und ganz wichtig, am Ende wird die ID der Erweiterung selbst zurückgegeben, die in ErweiterungsID umbenannt wird, so wie die Anzahl der gesamten vorkommen einer ID. Das bedeutet der wichtigste Teil der Abfrage ist `AnzPerID`, da diese beinhaltet wie viele Treffer eine Erweiterung in der Suche insgesamt hatte. Mit `ext.filePath` wird der Link für die Ergebnisseite der Erweiterung zurückgegeben. Diese gesamten Daten werden zurückgegeben, da diese im späteren Verlauf wichtig für die Generierung der Auflistung der Suchergebnisse sind.

Im nächsten Schritt werden im FROM Statement die verwendeten Tabellen genannt. Dabei ist das Segment mit `**` der Teil, in dem die Suchanfrage später eingefügt wird. Dazu später mehr. Die Tabelle die von dieser Suchanfrage zurückgegeben wird, wird `Search` genannt.

Am Ende der Abfrage wird durch den GROUP BY Befehl die Erweiterungen die die gleiche ID haben zusammengefasst und Absteigend sortiert, nach Anzahl der Häufigkeit an Suchtreffern aus `AnzPerID`.

4.4.4 Generierung der Suchanfrage

Die einzelnen Statements werden serverseitig generiert. Dabei wird bei der Generierung zwischen den drei unterschiedlichen Varianten unterschieden. Eine Suchanfrage in der man den Titel einer Erweiterung durchsucht ist anders aufgebaut als die, die nach Keywords sucht. Das Statement für die Suche Search By Name sieht wie folgt aus:

Listing 4.4: Titel Suche SQL Query

```
1 SELECT ext.erwid AS erw FROM final.extension AS ext WHERE ext.titel LIKE "%
   titleToSearch%";
```

Mit diesem Statement wird erreicht, dass von jeder Erweiterung, die in der Datenbank vorhanden ist, der Titel betrachtet wird. Mit dem LIKE Befehl wird nach einem Pattern gesucht, in diesem Fall ist das Pattern `%titleToSearch%`. Das Pattern lautet in einfachen Worten, wenn im Titel einer Erweiterung das vom User eingegebene `titleToSearch` gefunden wird, dann ist dies ein Treffer. Dabei ist es egal, was vor oder hinter der Eingabe steht, das wird mit den beiden Prozentzeichen signalisiert.

Die nächste Abfrage bezieht sich auf die Keyword Suchvariante. Dieses Statement lautet wie folgt:

Listing 4.5: Keyword SQL Query

```
1 SELECT ext.erwid AS erw FROM final.describe_Keyword AS desKey, final.
   extension AS ext WHERE ext.erwid = desKey.extension and desKey.keyword
   in (-1, ...);
```

Auch hier steht der wichtige Teil am Ende, nämlich `desKey.keyword in (-1, ...)`, denn das ist die eigentliche Abfrage. In kurz bedeutet es, wenn ein Keyword in der Tabelle gefunden wird, dann gib die ID der Erweiterung zurück, zu der es gehört. Die Punkte bedeuten, dass hier die vom User ausgewählten IDs noch hinzugefügt werden. Die vorangestellte „-1“ besitzt keine Bedeutung für die Abfrage. Diese ist nur da, um später die Erstellung der Abfrage zu erleichtern. Die Abfrage für den hinteren Teil wird wie folgt generiert.

Listing 4.6: Generiere Suchanfrage für Keyword

```
1 if (count($keyword) > 0)
2 {
3
4     if ($sqlFormattedQuery != "")
5     {
6         $sqlFormattedQuery .= " UNION ALL ";
7     }
8     $sqlFormattedQuery .= "SELECT ext.erwid AS erw FROM final.
        describe_Keyword AS desKey, final.extension AS ext WHERE ext.erwid
        = desKey.extension AND desKey.keyword in (-1";
9     foreach ($keyword as $key)
10    {
11        $sqlFormattedQuery .= ", ".$key['key'];
12        $anzSelections++;
13    }
14    $sqlFormattedQuery .= ")";
15 }
```

Auch dieser Codeabschnitt wird kurz Schritt für Schritt erklärt. In der ersten Zeile wird festgestellt, ob das Array, welches die Keywords enthält, überhaupt Keywords

beinhaltet. Falls ja, wird mit der Ausführung der Generierung begonnen. In die Variable *\$sqlFormattedQuery* werden alle SQL Abfragen, die von den einzelnen Suchvarianten und Parametern generiert werden, hinzugefügt und später in die gesamte Abfrage von 4.3 eingefügt. Falls der String *\$sqlFormattedQuery* schon Werte besitzt, dann soll vor der Generierung das Statement UNION ALL hinzugefügt werden. Dieses MySQL Statement ist für die Abfrage sehr wichtig. Denn es führt dazu, dass die Ergebnisse aller einzelnen SQL Abfragen, also alle vom Server generierten SQL Abfragen, später zusammengefasst werden. Mit UNION ALL wird zugleich sichergestellt, dass Ergebnisse auch mehrfach hinzugefügt werden können. Am Ende wird in einer Schleife das Array durchgelaufen und jedes Vorkommen eines Keywords zur Abfrage hinzugefügt.

Die letzte Variante ist die Kategoriensuche. Dessen Generierung und Aufbau ist analog zu der Keywordsuche, mit dem einzigen Unterschied, dass in der Kategoriensuche erst unterschieden werden muss, welche Kategorien ausgewählt sind. Dies wird mit einer Switch-Case Anweisung gelöst.

Bevor die generierte SQL Abfrage ausgeführt wird, wird erst einmal überprüft, ob überhaupt ein Suchparameter ausgewählt ist. Falls keine Suchparameter vom Nutzer ausgewählt sind, soll abgebrochen werden, indem die Verbindung zur Datenbank beendet wird und die Meldung „No Values Typed in“ dem Nutzer angezeigt wird. Andernfalls soll die Suchanfrage um den Schlussteil der SQL Abfrage ergänzt werden. Danach wird die Suchanfrage an die Datenbank gestellt. Bevor aber das Ergebnis genauer betrachtet wird, wird zuerst überprüft, ob die Datenbank auf die Suchanfrage überhaupt einen Treffer zurückgeliefert hat. Falls es keine Treffer mit den ausgewählten Suchparametern gibt, soll dem Nutzer dieses mal die Meldung „No Results“ angezeigt werden. Wenn Ergebnisse existieren, dann soll ein Ergebniseintrag erstellt und dem Nutzer angezeigt werden.

4.5 Erstellen und Bearbeiten von Extensions

In diesem Kapitel wird gezeigt, wie das Erstellen und Bearbeiten serverseitig gehandhabt wird. Wie bei der grafischen Oberfläche teilen sich auch hier das Erstellen und Bearbeiten einer Erweiterung viele der Operationen und sind daher, bis auf ein paar wenige Unterschiede, nahezu identisch. Deswegen werden beide Vorgehen in diesem Kapitel zusammengefasst. Es gibt 3 wichtige Punkte beim Erstellen und Bearbeiten von Extensions. Der erste Punkt ist die Datenstruktur der Dateien, wie Bilder oder PDF's auf dem Server. Dem Nutzer ist es erlaubt Bilder und PDF's hochzuladen, die dann mit auf die Webseite eingebunden werden. Der zweite Punkt sind die Datenbankoperationen. Diese ermöglichen es einen neuen Eintrag in die Datenbank hinzuzufügen oder zu bearbeiten. Und als letzter Punkt steht die Generierung einer Ergebniseite auf der Agenda. Dieses Vorgehen wird in den folgenden Unterkapiteln näher erklärt.

4.5.1 Datenstruktur

Damit der Nutzer Bilder oder PDF's hochladen kann, wird eine Datenstruktur auf dem Server benötigt, die dies dem Nutzer ermöglicht. Für das Hochladen von Dateien gibt es einen separaten Ordner auf dem Server, dieser heißt `tmp`. In diesem hat jeder Nutzer volle Schreib und Lese Rechte. Dieser Ordner wird sowohl für das Erstellen als auch für das nachträgliche Bearbeiten von Extensions verwendet. Dieser `tmp` Ordner wird benötigt, damit der Nutzer in der Lage ist Dateien Hochzuladen und diese auch wieder zu löschen. Wenn eine Extension letztendlich in die Datenbank hinzugefügt wird, egal ob neu Erstellung oder nachträgliche Bearbeitung, wird ein neuer Ordner im Ordner `extensionFiles` erstellt. Dieser neu erstellte Ordner trägt den Namen des Primärschlüssels der Extension Tabelle aus der Datenbank, also den Wert aus der Spalte `erwid`. Alle Dateien aus dem `tmp` Ordner, natürlich nur die, die zur Extension gehören, werden in diesen neuen Ordner kopiert und aus dem `tmp` Ordner gelöscht. Der neu erstellte Ordner in `extensionFiles` besitzt nur Lese Rechte für nicht Administratoren. Damit wird gewährleistet, dass im Nachhinein der Inhalt nicht manipuliert werden kann.

Im bearbeiten Modus wird der komplette Ordner, in der sich die Dateien befinden, zurück in den `tmp` Ordner kopiert, sodass diese wieder vom Nutzer manipuliert werden können. Beim Bearbeiten einer Extension wird der vorhanden Ordner der Extension in `extensionFiles` erst gelöscht, wenn der Nutzer die Bearbeitung erfolgreich abschließt und somit die Veränderungen in die Datenbank setzt. Damit wird gewährleistet, dass die Daten nicht von dem Server verschwinden, wenn der Nutzer den bearbeiten Modus vorzeitig abbricht.

Die Funktionalität, Bilder und PDF's hochzuladen wird über die PHP Dateien von `fileUpload.php` und `pdfUpload.php` gewährleistet. Um Dateien zu löschen wird das Script `deleteImage.php` aufgerufen. Beim Hochladen jeder Datei wird spezifisch darauf geachtet, das in den Feldern die korrekte Datei hochgeladen wird. Das bedeutet bei Bildern sollen nur die Formate JPG, PNG, JPEG und GIF erlaubt sein mit einer maximalen Größe von 2 MB. Bei dem Hochladen einer PDF dürfen ausschließlich nur PDF Dateien mit einer maximal Größe von 8 MB hochgeladen werden.

4.5.2 Datenbankoperationen

Die Datenbankoperationen für das Bearbeiten einer Extension unterscheiden sich nur minimal von denen der Hinzufügen einer Extension. Eine generelle Reihenfolge der Ausführung ist nicht zu beachten, außer, das zuerst in die `extension` Tabelle geschrieben werden muss, da man den Primärschlüssel der Extension benötigt. Dieser Schlüssel wird, wie jeder Schlüssel, automatisch durch `AUTO_INCREMENT` generiert. Ein Beispiel zum Hinzufügen eines neuen Eintrages wird anhand der Kategorie Class gezeigt. In Listing 4.7 ist der Code Abschnitt zum Hinzufügen einer neuen Class oder Subclass.

Listing 4.7: Hinzufügen einer Klasse mit Subklasse in die Datenbank

```

1 //-----class subclass
2
3
4 //---class
5 $stpt = $conn->prepare("INSERT INTO final.class (mainname) SELECT ? FROM
      DUAL WHERE NOT EXISTS (SELECT mainname FROM final.class WHERE mainname
      = ? AND subname IS NULL)");
6 $tmp_class = $class;
7 $stpt->bind_param("ss", $class, $tmp_class);
8
9 $stpt->execute();
10
11 $id_class_query = "SELECT classid FROM final.class WHERE mainname = ? AND
      subname IS NULL";
12
13 $stpt->close();
14
15 //----subclass
16 if($subclass != ""){
17     $bothClass = true;
18     $insertClassSubclass = $conn->prepare("INSERT INTO final.class (
      mainname, subname) SELECT ?, ? FROM DUAL WHERE NOT EXISTS (SELECT
      mainname, subname FROM final.class WHERE mainname = ? AND subname =
      ?)");
19     $tmp_subclass = $subclass;
20     $insertClassSubclass->bind_param("ssss", $class, $subclass, $tmp_class,
      $tmp_subclass);
21     $insertClassSubclass->execute();
22
23     $id_class_query = "SELECT classid FROM final.class WHERE mainname = ?
      AND subname = ?";
24
25     $insertClassSubclass->close();
26 }
27
28 $id_class_result = $conn->prepare($id_class_query);
29 if($bothClass){
30     $id_class_result->bind_param("ss", $class, $subclass);
31 } else {
32     $id_class_result->bind_param("s", $class);
33 }
34
35 $id_class_result->execute();
36 $id_class_result->bind_result($classID);
37 $id_class_result->fetch();
38
39 $id_class = $classID;
40
41 $id_class_result->close();
42
43 //Update extension table with Class ID
44 $conn->query("UPDATE final.extension SET idclass = $id_class WHERE erwid =
      $id_extension");

```

In diesem Codeausschnitt, sind die Zeilen 5 und 18 von Bedeutung. In diesen wird die Eingabe des Nutzers in die Datenbank hinzugefügt. Wobei bei doppelten vorkommen einer Klasse oder Subklasse, nichts hinzugefügt wird. Um dies in einer einzigen Abfrage zu bewerkstelligen, wird zu dem INSERT INTO Statement der Zusatz `SELECT ? FROM DUAL WHERE NOT EXISTS (SELECT mainname FROM final.class WHERE mainname = ? AND subname IS NULL)` hinzugefügt. Dieser überprüft ob der Wert schon existiert. Falls nicht, dann wird dieser neu hinzugefügt.

Wichtig ist auch anzumerken, dass PHP prepare Statements verwendet werden. Diese prepare Statements kompilieren eine SQL Abfrage schon im Voraus und mit den gesetzten Fragezeichen (?) in den Statements wird PHP signalisiert, dass dort noch Werte erwartet werden. Diese Werte werden durch den Befehl `bind_param`, an die Abfrage gebunden. Durch `execute` wird dann die SQL Abfrage ausgeführt. Diese prepare Statements bringen 2 große Vorteile mit sich.

Der erste Vorteil ist die Ausführungsgeschwindigkeit bei Mehrfachverwendung der selben Abfrage, da durch das Prepare die Abfrage schon kompiliert ist. Dies ist z.B. nützlich beim Hinzufügen von Keywords, denn dort wird mehrfach der gleiche Befehl verwendet um die Keywords in die Tabelle hinzuzufügen.

Ein weiterer Vorteil ist die Sicherheit vor SQL Injections. Da die SQL Abfrage schon kompiliert ist, ist es egal was der Nutzer eingibt. Die Daten werden immer als Daten behandelt und sind kein Teil der SQL Abfrage selbst.

In der Zeile 11 und 23 wird die Variable `$id_class_query` gesetzt. Diese wird benötigt, um nachdem hinzufügen die ID der soeben hinzugefügten Klasse zurückzubekommen. Die Variable wird 2 mal gesetzt. Zuerst ohne Subklasse und falls eine Subklasse existiert, wird die Variable in Zeile 23 mit der Subklasse neu gesetzt. In der letzten Zeile wird dann die Extension Tabelle mit der ID der Klasse upgedatet. Dieses Vorgehen ist analog zu dem Hinzufügen oder Bearbeiten der Domain.

Beim hinzufügen von Symbolen, Autoren oder positiven und negativen Schwierigkeitsgraden muss nicht darauf geachtet werden, ob ein Wert schon existiert. Diese können einfach ohne bedenken hinzugefügt werden. Bei der Universität und bei den Keywords muss allerdings vor dem hinzufügen nochmals überprüft werden, ob die Werte schon existieren.

Beim Bearbeiten einer Extension werden alle Werte upgedatet, egal ob diese verändert wurden oder nicht. Es ist einfacher alles im Vornherein zu updaten oder zu löschen und neu hinzuzufügen, als zu überprüfen, welche Spalte sich um welchen Wert verändert hat. Einfache update Operationen betreffen die Datenbank Tabellen Extension, University und works_in. Bei den restlichen Tabellen describe_Keyword, posDifficulties, negDifficulites, symbols und worked_on werden die Reihen gelöscht, in denen die Erweiterung die bearbeitet wird enthalten ist und neu hinzugefügt.

4.5.3 Ergebnisseite Generierung

Nachdem die Ergebnisse in die Datenbank erfolgreich eingetragen sind, wird das PHP Script `createExtensionHTML.php` durch einen `include` Befehl ausgeführt. Analog zu dem editieren von Extensions wird das PHP Script `editExtensionHTML.php` ausgeführt. Der Unterschied der beiden Scripte ist minimal, denn der einzige Unterschied liegt darin, dass bei `editExtensionHTML.php` Werte aus der Tabelle `updateExtensionDatas` mit hinzugefügt werden. Die Generierung ist im Endeffekt simple. Die Seite ist im voraus schon implementiert worden und der HTML Code dafür, ist als Vorlage in einer Variablen abgespeichert. An den entsprechenden Stellen, wo die Werte der Eingabe hinzugefügt werden müssen, ist Platz gelassen worden. Die meisten Daten können direkt eingefügt werden, so wie eine Übersichtstabelle oder eine kurz Beschreibung. An anderen Stellen müssen Tabellen, Auflistungen oder andere Darstellungen generiert werden. Ein Beispiel Codeabschnitt dafür ist die Generierung der negativen Schwierigkeiten Liste.

Listing 4.8: Generierung des HTML Code von negativen Schwierigkeiten List aus `insertDatabase.php`

```

1 foreach($negDifficulties as $valueTextNegDifficultie){
2     $insertNegDifficulties->execute();
3     $difficultieNegPrint .= '<li style="margin-bottom:25px;"><span class="
        glyphicon glyphicon-minus" style="margin-right:10px;"></span>'.
        $valueTextNegDifficultie.'</li>';
4 }
```

In 4.8 ist das Array `$negDifficulties` zu sehen, das alle Eingaben des Nutzers bezüglich der negativen Schwierigkeit enthält. Dieses wird ausgelesen und in Zeile 2 wird der Wert in die Datenbank hinzugefügt. In der darauf folgenden Zeile wird die Variable `$difficultieNegPrint` mit dem passenden HTML Code befüllt. Dieser wird dann in 4.9 in der Zeile 10 hinzugefügt.

Listing 4.9: Einfügen des generierten HTML Code in die input Variable

```

1 $input = '<html> ....
2 <div style="margin:0 auto; margin-top:20px;">
3     <div class="col-md-6" style="text-align:justify; padding:0px;">
4         <ul>
5             '.$difficultieProPrint.'
6         </ul>
7     </div>
8     <div class="col-md-6" style="text-align:justify; padding:0px;">
9         <ul>
10            '.$difficultieNegPrint.'
11        </ul>
12    </div>
13 </div>
14 ...
15 </html>'
```

Eine extra Generierung für die Oberfläche ist ebenfalls für die positiven Schwierigkeiten, Universität, Autor und Symbol notwendig. Bei `editExtensionHTML.php` kommt noch der Changelog hinzu. Hier wird eine Tabelle generiert, welche alle Veränderungen in der Laufzeit einer Extension auf der Datenbank enthält.

Wenn die Variable `$input` korrekt befüllt ist, wird eine HTML Datei geschrieben, dessen Namen der Titel der Extension ist und wird ebenfalls in den Ordner hinzugefügt, wo sich die Bilder und PDF Dateien befinden. Diese wird ebenfalls keine öffentlichen Schreibrechte besitzen. Beim Überarbeiten der Extension wird, genau wie bei Bildern oder PDF's, die HTML Datei gelöscht und am Ende neu generiert.

Kapitel 5

Zusammenfassung und Ausblick

In dieser Bachelorarbeit wurden insgesamt 30 BPMN Extensions analysiert und kategorisiert. Anschließend wurde eine Webseite erstellt, die diese Extensions beinhalten soll. Die Webseite besitzt die Funktion, Extensions hinzuzufügen, sowie diese im Nachhinein auch wieder zu bearbeiten. Außerdem wurde auch ein Suchalgorithmus implementiert, der anhand der Kategorisierungen der Extensions nach diesen suchen kann.

Ausblick

Die in dieser Bachelorarbeit entwickelte Webseite macht die Auffindung von BPMN Extensions um einiges einfacher als davor. Es existiert nun eine Anlaufstelle sowohl für BPMN Nutzer, um sich über BPMN Extensions zu erkundigen, als auch für BPMN Extensions Entwickler, die hier nun ein Portal haben, um diese Extensions auch verbreiten zu können. So liegen diese Extension nicht nur auf den Servern der jeweiligen Universität, Unternehmen oder Verlegern, sondern sind einheitlich auf einer Webseite kategorisiert zu finden. Jedoch ist die Arbeit noch lange nicht getan. Aus Mangel an Zeit, konnten viele geplante Features hier nicht umgesetzt werden. Dazugehören zum Beispiel eine Visualisierung der Daten, in Form von Balken oder Kreisdiagrammen, um zu zeigen, wie viele unterschiedliche Extensions bzw. wie viele gleichartige Extensions hinzugefügt wurden. Ein weiteres Feature, wäre gewesen aus diesen Diagrammen eine Art interaktive Suche zu gestalten. Ein letztes geplantes Feature war, dass die Eingabe der Keywords durch eine stochastische Textanalyse erweitert wird. So hätte der Nutzer einen Text, z.B. den Abstract oder die Einleitung der Extension hinzugefügt, und daraus hätten die Keywords automatisch gebildet werden können. Dies sind nur einige Features die noch geplant waren. Das bedeutet, die Webseite kann bzw. sollte stetig weiter verbessert werden um das Suchen weiter zu erleichtern oder zu verbessern. Es können natürlich auch neue Kategorisierungspunkte hinzugefügt werden um eine Extensions noch klarer von anderen zu unterscheiden. Wie man sieht, ist durch diese Bachelorarbeit ein Grundstein gelegt worden, den man aber weiterhin verbessern und erweitern kann.

Literatur

- [1] Zaharah Allah Bukhsh.
„BPMN Plus: a modelling language for unstructured business processes“. In: (2015).
URL: <http://essay.utwente.nl/67945/> (besucht am 09.06.2016) (siehe S. 57).
- [2] Anne Baumgrass u. a. „BPMN extension for business process monitoring.“ In: *EMISA*. 2014, S. 85–98. URL: <http://cs.emis.de/LNI/Proceedings/Proceedings234/85.pdf> (besucht am 09.06.2016) (siehe S. 48).
- [3] Paolo Bocciarelli und Andrea D’Ambrogio.
„A BPMN extension for modeling non functional properties of business processes“. In: *Proceedings of the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*. Society for Computer Simulation International. 2011, S. 160–168.
URL: <http://dl.acm.org/citation.cfm?id=2048497> (besucht am 09.06.2016) (siehe S. 37).
- [4] Marco Brambilla, Piero Fraternali und Carmen Vaca.
„A notation for supporting social business process modeling“. In: *Business Process Model and Notation*. Springer, 2011, S. 88–102.
URL: http://link.springer.com/chapter/10.1007/978-3-642-25160-3_7 (besucht am 09.06.2016) (siehe S. 45).
- [5] Roland Braun u. a.
„Bpmn4cp: Design and implementation of a bpmn extension for clinical pathways“. In: *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*. IEEE. 2014, S. 9–16. URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6999261&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6999261 (besucht am 09.06.2016) (siehe S. 12, 29).
- [6] Achim D Brucker u. a. „SecureBPMN: Modeling and enforcing access control requirements in business processes“. In: *Proceedings of the 17th ACM symposium on Access Control Models and Technologies*. ACM. 2012, S. 123–126.
URL: <http://dl.acm.org/citation.cfm?id=2295160> (besucht am 09.06.2016) (siehe S. 12, 43).
- [7] Dane Cameron. *HTML5, JavaScript und jQuery*. dpunkt Verlag, 2015. ISBN: 978-3-86490-268-0 (siehe S. 80).

- [8] Anis Charfi, Heiko Müller und Mira Mezini.
 „Aspect-oriented business process modeling with AO4BPMN“.
 In: *Modelling Foundations and Applications*. Springer, 2010, S. 48–61.
 URL: http://link.springer.com/chapter/10.1007/978-3-642-13595-8_6 (besucht am 09.06.2016) (siehe S. 38).
- [9] Gero Decker und Frank Puhlmann.
 „Extending BPMN for modeling complex choreographies“. In: *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*. Springer, 2007, S. 24–40.
 URL: http://link.springer.com/chapter/10.1007/978-3-540-76848-7_4 (besucht am 09.06.2016) (siehe S. 44).
- [10] Jan-Philipp Friedenstab u. a. „Extending BPMN for business activity monitoring“.
 In: *System Science (HICSS), 2012 45th Hawaii International Conference on*. IEEE. 2012, S. 4158–4167. URL:
http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6149402&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6149402
 (besucht am 09.06.2016) (siehe S. 40).
- [11] Denis Gagne und André Trudel. „Time-bpmn“.
 In: *Commerce and Enterprise Computing, 2009. CEC'09. IEEE Conference on*. IEEE. 2009, S. 361–367. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5210771&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D5210771 (siehe S. 17).
- [12] Feng Gao u. a.
 „Extending BPMN 2.0 with Sensor and Smart Device Business Functions“.
 In: *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2011 20th IEEE International Workshops on*. IEEE. 2011, S. 297–302 (siehe S. 12).
- [13] Sascha Goldner und Alf Papproth.
 „Extending the BPMN Syntax for Requirements Management“.
 In: *Business Process Model and Notation*. Springer, 2011, S. 142–147.
 URL: http://link.springer.com/chapter/10.1007/978-3-642-25160-3_13 (besucht am 09.06.2016) (siehe S. 32).
- [14] Marija Jankovic u. a. „Enhancing BPMN 2.0 informational perspective to support interoperability for cross-organizational business processes“.
 In: *Computer Science and Information Systems* 12.3 (2015), S. 1101–1120. URL:
http://www.doiserbia.nb.rs/Article.aspx?id=1820-02141500013J#.VsRyJ_LhDIU
 (besucht am 09.06.2016) (siehe S. 47).
- [15] Oliver Kopp u. a. „BPMN4TOSCA: A Domain-Specific Language to Model Management Plans for Composite Applications“.
 In: *Business Process Model and Notation: 4th International Workshop, BPMN 2012, Vienna, Austria, September 12-13, 2012. Proceedings*.
 Hrsg. von Jan Mendling und Matthias Weidlich.
 Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, S. 38–52. ISBN: 978-3-642-33155-8.
 DOI: 10.1007/978-3-642-33155-8_4.

- URL: http://link.springer.com/chapter/10.1007/978-3-642-25160-3_13 (besucht am 09.06.2016) (siehe S. 34).
- [16] Markus Kuhn. „Erweiterung der BPMN zur Spezifikation von Prozessmodellen für Cross-Device Anwendungen“. In: (2014).
URL: <http://www.qn-it.de/ebook/bachelorarbeit.pdf> (besucht am 09.06.2016) (siehe S. 50).
- [17] Azeem Lodhi, Veit Küppen und Gunter Saake.
„An extension of bpmn meta-model for evaluation of business processes“.
In: *Scientific Journal of Riga Technical University. Computer Sciences* 43.1 (2011), S. 27–34.
URL: <http://www.degruyter.com/view/j/acss.2011.43.issue--1/v10143-011-0004-7/v10143-011-0004-7.xml> (besucht am 09.06.2016) (siehe S. 25).
- [18] Matteo Magnani und Danilo Montesi. „BPDMMN: A conservative extension of BPMN with enhanced data representation capabilities“.
In: *arXiv preprint arXiv:0907.1978* (2009).
URL: <http://arxiv.org/abs/0907.1978> (besucht am 09.06.2016) (siehe S. 53).
- [19] Bartosz Marcinkowski und Michal Kuciapski.
„A business process modeling notation extension for risk handling“.
In: *Computer Information Systems and Industrial Management*. Springer, 2012, S. 374–381.
URL: http://link.springer.com/chapter/10.1007/978-3-642-33260-9_32 (besucht am 09.06.2016) (siehe S. 28).
- [20] Ricardo Martinho, Dulce Domingos und João Varajão.
„CF4BPMN: A BPMN Extension for Controlled Flexibility in Business Processes“.
In: *Procedia Computer Science* 64 (2015), S. 1232–1239.
URL: <http://www.sciencedirect.com/science/article/pii/S1877050915026447> (besucht am 09.06.2016) (siehe S. 31).
- [21] Mariusz Momotko und Bartosz Nowicki.
„Visualisation of (distributed) process execution based on extended bpmn“.
In: *Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on*. IEEE, 2003, S. 280–284 (siehe S. 12).
- [22] Hanane Ben-Abdallah Mouna Rekik Khouloud Boukadi.
„BBPMN Meta-Model Extension with Deployment and Security“. In: (2013).
(Besucht am 06.01.2016) (siehe S. 19).
- [23] Niels Mueller-Wickop, Martin Schultz und M Nuettgens.
„Modelling Concepts For Process Audits-Empirically Grounded Extension Of BPMN.“
In: *ECIS*. 2013, S. 194. URL: http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1417&context=ecis2013_cr
(besucht am 09.06.2016) (siehe S. 54).
- [24] Christine Natschläger. „Deontic BPMN“. In: *Database and Expert Systems Applications*. Springer, 2011, S. 264–278.
URL: http://link.springer.com/chapter/10.1007/978-3-642-23091-2_23
(besucht am 09.06.2016) (siehe S. 49).

- [25] Luisa Parody, María Teresa Gómez-López und Rafael M Gasca.
Extending BPMN 2.0 for modelling the combination of activities that involve data constraints. Springer, 2012.
URL: http://link.springer.com/chapter/10.1007/978-3-642-33155-8_6 (besucht am 09.06.2016) (siehe S. 41).
- [26] Cesare Pautasso. „Bpmn for rest“. In: *Business Process Model and Notation*. Springer, 2011, S. 74–87.
URL: http://link.springer.com/chapter/10.1007/978-3-642-25160-3_6 (besucht am 09.06.2016) (siehe S. 26).
- [27] Raquel M Pillat u. a.
„BPMNt: A BPMN extension for specifying software process tailoring“.
In: *Information and Software Technology* 57 (2015), S. 95–115.
URL: <http://www.sciencedirect.com/science/article/pii/S0950584914002031> (besucht am 09.06.2016) (siehe S. 51).
- [28] Werner Esswein Richard Braun. „Entwicklung einer BPMN-Extension für ressourcen-intensive Prozesse im Maschinenbau“.
In: *Tagungsband Multikonferenz Wirtschaftsinformatik 2014* (2014) (siehe S. 36).
- [29] Alfonso Rodríguez u. a.
A BPMN extension for including data quality requirements in business process modeling. Springer, 2012.
URL: http://link.springer.com/chapter/10.1007/978-3-642-33155-8_10 (besucht am 09.06.2016) (siehe S. 24).
- [30] Kawther Saeedi, Liping Zhao und Pedro R Falcone Sampaio.
„Extending BPMN for supporting customer-facing service quality requirements“.
In: *Web Services (ICWS), 2010 IEEE International Conference on*. IEEE. 2010, S. 616–623.
URL:
http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5552734&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D5552734
(besucht am 09.06.2016) (siehe S. 20).
- [31] Koh Song Sang und Bo Zhou. „BPMN Security Extensions for Healthcare Process“.
In: *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*. IEEE. 2015, S. 2340–2345. URL:
http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=7363392&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D7363392
(besucht am 09.06.2016) (siehe S. 30).
- [32] Luis Jesús Ramón Stropi, Omar Chiotti und Pablo David Villarreal.
„A BPMN 2.0 extension to define the resource perspective of business process models“.
In: *XIV Iberoamerican Conference on Software Engineering*. 2011, S. 25–38.
URL: http://www2.dcc.ufmg.br/eventos/CIBSEpapers/2011/cibse_paper02.pdf
(besucht am 09.06.2016) (siehe S. 56).

- [33] Sema Zor, F Leymann und D Schumm.
„A proposal of BPMN extensions for the manufacturing domain“.
In: *Proceedings of 44th CIRP international conference on manufacturing systems*.
Citeseer. 2011. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.362.3033&rep=rep1&type=pdf> (besucht am 09.06.2016) (siehe S. 12, 22).

Alle Fußnoten URLs wurden zuletzt am 2016-06-09 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift