

Institut für Architektur von Anwendungssystemen

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit Nr.110

**Repository zur Lösungserfassung in
der Domäne von Kostümen im Film**

Bernhard Wetzel

Studiengang: Softwaretechnik

Prüfer/in: Prof. Dr. Frank Leymann

Betreuer: Dipl.-Inf. Christoph Fehling
Johanna Barzen, M.A.

begonnen am: 17.01.2014

beendet am: 17.07.2014

CR-Klassifikation:D.1,D.2,E.1,H.2,J.5

Kurzfassung

Gleiche oder sehr ähnliche Probleme treten in Bereichen von Arbeitsprozessen immer wieder auf. Es werden bequeme, einfache und vor allem schnelle Lösungsansätze benötigt. Heutzutage werden so genannte Patterns (dt.: Muster) verwendet, da sie bewährte Lösungen für bestimmte Probleme sind.

Auch in der Filmindustrie möchte man nun diesen Vorteil ergreifen, jedoch ist zunächst einige Vorarbeit notwendig. Es müssen bewährte Lösungen gefunden werden. Zu diesem Zweck werden zunächst Daten gesammelt. Dabei ist ein einfaches und geschicktes Werkzeug von Nöten, damit IT-fremde Nutzer die Möglichkeit haben ohne Schulung damit zu arbeiten. Des Weiteren wird eine Begriffsbasis benötigt, auf deren sich die verschiedenen Nutzer verständigen können, da es sich um subjektive und durch den persönlichen Begriffsschatz des Nutzers beeinflusste Daten handelt.

Ziel der Bachelorarbeit ist die Erweiterung des vorhandenen Werkzeugs zur verbesserten und schnelleren Eingabe der Daten.

Abstract

Same or quite similar problems keep occurring in range of workprocesses. Convenient, easy and particularly quick solutions are in big need. Today patterns are used since they are proven solutions for certain problems. To use this advantage in filmindustry some prework is necessary. Proven solutions have to be found and for this cause data has to be gathered. To gather this data an easy and high usability tool is needed, since the data is gathered by non-IT people. Another point is that a common termdefinition is needed on which the users can communicate, given that the data is influenced by the subjectivity and the personal word pool of the user. The goal of this bachelorthesis is the improvement of an exiting tool for better and faster input of data.

Inhaltsverzeichnis

1	Einleitung	5
2	Grundlagen und verwandte Arbeiten	6
2.1	Pattern	6
2.2	Entwicklung von Patterns	7
2.3	Taxonomien	7
2.4	Ontologien	9
3	Fachliche Anforderungen an MuSE	9
3.1	Zweck von MuSE	9
3.2	Rollen	9
3.3	Datenbankschema (ER-Diagramm)	10
3.3.1	Film	12
3.3.2	Rolle	12
3.3.3	Kostüm	12
3.3.4	Basiselement	12
3.3.5	Teilelement	12
3.3.6	Basiselement Relation	12
3.4	Verwendete Taxonomien	13
3.4.1	Basiselemente	13
3.4.2	Beschaffenheit	13
3.4.3	Operatoren	14
3.4.4	Sonstige	14
3.4.5	Probleme	14
4	Lösungen der Taxonomieprobleme	15
4.1	Konzept des MuSE-Tool-Taxonomy-Parsers	15
4.1.1	Benutzung des Parsers	15
4.1.2	Aufbereitung der Daten	15
4.1.3	Parsing der Daten	16
4.1.4	Abbildung in der Datenbank	16
4.2	Konzept des TreeSelectors	17
4.2.1	Verwendung des Tree-Selectors	18
4.2.2	Implementierung	18
5	Evaluation	19
5.1	Leistungsfähigkeit des MuSE - Tool - Taxonomy - Parsers	19
5.2	Änderbarkeit des MuSE-Tool-Taxonomy-Parsers	20

5.3	Vorteile des TreeSelectors	21
6	Zusammenfassung und Ausblick	21
6.1	Zusammenfassung	21
6.2	Ausblick	22
6.2.1	Erweiterung und Verbesserungen der Eingabe	22
6.2.2	Darstellung zur Auswertung der gesammelten Daten	22
7	Anhang A: Verwendete Werkzeuge	23
8	Anhang B: Literaturverzeichnis	24

Abbildungsverzeichnis

1	Unterbaum der Taxonomie "Mütze" [Bar13]	8
2	ER-Diagramm des Aufbaus der Domäne	11
3	Beispiel Baum der Taxonomie Produktionsorte	18
4	Laufzeit des MuSE - Tool - Taxonomy - Parsers	20

Listings

1	XMLReader.java	16
2	SQLParser.java	16
3	Baumstruktur	19

Abkürzungsverzeichnis

- MuSE - Muster-Suchen-und-Erkennen
- HTML - Hyper Text Markup Language
- MySQL - My Standard Query Language
- UML - Unified Modeling Language
- XML - eXtensible Markup Language
- ER-Diagramm - Entity Relationship Diagramm

1 Einleitung

In der Architektur, vor allem im Städtebau und mittlerweile auch in der Softwareindustrie werden mit großem Erfolg so genannte "Patterns" (dt.: Muster) verwendet. Die Verwendung von Patterns ermöglicht eine schnelle und leicht verständliche Lösung von wiederkehrenden Problemen.

Auch in der Filmindustrie gibt es wiederkehrende Probleme, nämlich die Kostümierung der verschiedenen Filmrollen. Darum hat man sich zum Ziel gesetzt, das System der Patterns auch in der Filmindustrie zu verwenden. Da bereits eine Vielzahl von Filmen existieren, welche das Problem der Kostümierung auf verschiedene oder gleiche Weise gelöst haben, kann man aus diesen eine große Anzahl verschiedenster Patterns herausarbeiten. Diese können anschließend in zukünftigen Inszenierungen verwendet werden. Um jedoch die Patterns entwickeln zu können, müssen erst die Daten der bereits abgedrehten Filme gesammelt werden. Zu diesem Zweck wird das Muster-Suchen-und-Erkennen Tool, kurz MuSE, unter dem Begriff der "Digital Humanities" (dt.: Digitale Geisteswissenschaften) entwickelt.

Digital Humanities bezeichnet die Verwendung Computer gestützter Verfahren in Geistes- und Kulturwissenschaften. Zentrale Aufgaben der Digital Humanities sind zum Beispiel Suchverfahren, Text Mining, fachspezifische Datenbanken, Fachinformation oder Langzeitarchivierung (vgl. [wik14]).

Mit Hilfe des MuSE-Tools können die Filme gesichtet und die Daten der Kostümierung gesammelt werden. In naher Zukunft sollen sich mithilfe des Tools die gesammelten Daten abfragen und die Ergebnisse graphisch darstellen lassen. Auf diese Weise können Gemeinsamkeiten und Unterschiede dargestellt und somit die erfolgreicherer Lösungen von den qualitativ schlechteren getrennt werden.

Das MuSE-Tool verwendet zur einheitlichen Erfassung von Kostümen eine Taxonomiesammlung¹. Diese umfasst mehrere tausend Einträge zu Kostüm-elementen, deren Eigenschaften sowie Trageweisen. Um auf diese Sammlung zugreifen zu können, werden diese in eine Datenbank geladen und mithilfe einer Baumstruktur im Werkzeug dargestellt.

¹[Wah00] gr.: Taxonomie: Einordnung in ein bestimmtes System

Die vorliegende Bachelorarbeit befasst sich mit den folgenden Punkten:

- Grundlagen und verwandte Arbeiten
- fachliche Anforderungen für MuSE
- Lösungen der Taxonomieprobleme
- Evaluation
- Zusammenfassung und Ausblick

2 Grundlagen und verwandte Arbeiten

Dieses Kapitel beschäftigt sich mit Grundlagen und Begriffen, welche für das Verständnis der weiteren Arbeit notwendig sind.

2.1 Pattern

Der Begriff Pattern wurde zunächst nur im Zusammenhang mit der Städtearchitektur verwendet. Christopher Alexander hat mit seiner Aussage "Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without doing the same way twice" [Ale77] die Architektur von Städten enorm verbessert. Daraufhin hat man schnell erkannt, dass sich Patterns auch im Softwaretechnikbereich verwenden lassen. Im Buch "Design Patterns - Elements of Reusable Object-Oriented Software" [GRJV03] wird ausführlich beschrieben, wie solche Patterns aufgebaut sind und wie sie in der objekt-orientierten Softwareentwicklung verwendet werden. Es handelt sich dabei um strukturierte Textdokumente, welche mindestens aus den 4 folgenden Elementen bestehen.

Name

Er beschreibt grob das Problem, die Lösung und die daraus resultierenden Konsequenzen.

Problem

Hier wird beschrieben um welche Probleme es sich handelt, in welchem Kontext sie stehen.

Lösung

Sie beschreibt die einzelnen Module, aus denen das Lösungsdesign aufgebaut ist, sowie dessen Beziehungen, Aufgaben, Zusammenarbeit und warum das Problem auf diese spezielle Weise gelöst werden.

Konsequenzen

Vor- und Nachteile der Verwendung des Patterns, sowie Beschreibung des resultierenden Zustands nach Verwendung des Patterns.

Ziel ist es nun, herauszufinden ob Patterns auch in der Filmindustrie von Nutzen sein können. Zu diesem Zweck wird ein Werkzeug entwickelt, mithilfe dessen zunächst Daten über Filme und insbesondere der Filmkostüme in der Domäne gesammelt und anschließend strukturiert angezeigt und ausgewertet werden können.

2.2 Entwicklung von Patterns

Um das Problem der Filmindustrie, welche Kostümierung für die Filmrolle am geeignetsten ist, lösen zu können, benötigen man zunächst eine möglichst große Menge von Daten. Da bereits abgedrehte Filme dieses Problem, das noch kommende Filme haben werden, gelöst haben, können wir auf eine gigantische Menge von Problemen (Kostümierung der Schauspieler) und Lösungen (tatsächliche Kostümierung) zurückgreifen. Um die Lösungen zu dokumentieren, damit Regisseure in zukünftigen Filmen darauf zugreifen können, muss dieser Datenberg zunächst in eine sinnvolle Umgebung gebracht werden, damit sich dieser analysieren lässt. Das MuSE-Tool wurde entwickelt um, eine Möglichkeit zu bieten, die Daten zu sammeln und soll im nächsten Schritt weiterentwickelt werden, um die gesammelten Daten nach bestimmten Kriterien gefiltert abzufragen.

2.3 Taxonomien

Beim Sammeln dieser Menge von Daten rückt ein Problem in den Vordergrund. Da jeder Mensch die Welt anders erlebt, sieht und beschreibt, wird,

um die verschiedenen Kostümteile sinnvoll vergleichen zu können, eine gemeinsame Begriffsbasis benötigt.

Diese Basis bilden die Taxonomien. Sie sind Klassifikationsschemata, die mithilfe von bestimmten Kriterien Objekte klassifizieren und in Klassen oder Taxa² zu sortieren. Sie erlauben es Taxonomien wie, Basiselemente, Teilelemente usw. in eine hierarchische Ordnung zu bringen. Mithilfe der Taxonomien können später die Gemeinsamkeiten zwischen den Lösungen gezeigt werden, da diese nun durch automatisierte Verfahren verarbeitet werden können (vgl. [Bar13]).

Die folgende Abbildung (1) zeigt am Beispiel "Mütze", wie sich das Basiselement in weitere untergeordnete Basiselemente aufteilen lässt.

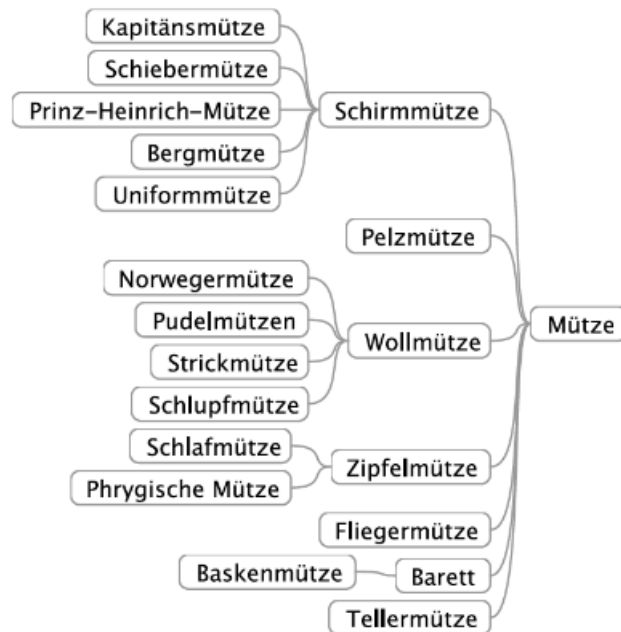


Abbildung 1: Unterbaum der Taxonomie "Mütze" [Bar13]

Die im MuSE-Tool verwendeten Taxonomien wurden bereits in "Taxonomien kostümrelevanter Parameter: Annäherung an eine Ontologisierung der Domäne des Filmkostüms" [Bar13] entwickelt.

²[Wah00] gr.: eig. Taxon: Systematische Kategorie

2.4 Ontologien

Im Gegensatz zu Taxonomien, welche einen hierarchischen Zusammenhang darstellen, sind Ontologien³ ein Informationsnetzwerk mit logischen Relationen. Sie enthalten hauptsächlich formale Darstellungen von Begrifflichkeiten und deren Beziehungen. Ihre Aufgabe ist es, Daten zwischen Anwendungsprogrammen und Diensten zu verschicken um

- bereits bestehende Datenbestände zusammenzufügen
- in bestehenden Datenbeständen zu suchen und diese zu editieren
- aus Typen von Datenbeständen neue Instanzen zu generieren.

Die meisten bekannten Anwendungen von Ontologien kennen keine individuellen Instanzen und beschränken sich auf wissenschaftliche Zwecke zur Systematisierung der Nutzung von Begriffsräumen(vgl. [Gru93]).

3 Fachliche Anforderungen an MuSE

In diesem Kapitel werden die Aufgaben und Anforderungen an das MuSE-Tool, der gewählte Aufbau und die zu lösenden Probleme beschrieben.

3.1 Zweck von MuSE

Zweck des MuSE-Tools ist zunächst die Sammlung von Film-, Schauspieler-, Rollen- und Kostümdateien in einer Domäne. In dieser Domäne können später Regisseure oder andere Experten auf die gesammelten Daten zugreifen und die für ihren Kontext besten Lösungen auswählen.

3.2 Rollen

Im Muse-Tool gibt es vier verschiedene Rollen.

Lösungssammler

Er ist für das Einpflegen der Daten verantwortlich. Seine Aufgabe ist es, die Filme zu sichten und die Kostüme zu dokumentieren.

³[Wah00] gr.: Lehre vom Sein und seinen Prinzipien

Domänenexperte

Er verwaltet die Taxonomien und Ontologien.

Patternautor

Seine Aufgabe ist es, aus den gesammelten Lösungen Muster zu entwickeln.

Administrator

Er ist für Probleme mit dem Tool zuständig, ebenso wie für das Ändern der Eingabemasken oder der Taxonomien auf Wunsch der anderen Rollen.

3.3 Datenbankschema (ER-Diagramm)

Das folgende ER-Diagramm zeigt eine schematische Darstellung des Zusammenhangs der einzelnen Komponenten. Es beschreibt, welche Taxonomien für welche Einträge benötigt werden und wie der hierarchische Aufbau gewählt wurde. Die gelben Elemente bezeichnen hier die Objekte in der Datenbank (Film, Rolle, Kostüm, Basiselement, Teilelement, Relation), die grünen Elemente die begrenzten Eingaberäume der Objektinhalte (Taxonomien). Die Zahlen auf den Verbindungen geben die Kardinalitäten an, d.h. die Anzahl der möglichen verschiedenen Eingaben. Mithilfe der Basiselement - Relation kann der Sammler beschreiben, wie die einzelnen Basiselemente eines Kostüms zusammenhängen (z.B. Schuhe werden über den Socken getragen). Jedes Teilelement kann aus einem oder mehreren Unterteilelementen bestehen (z.B. Gesäßtasche kann einen Knopf haben).

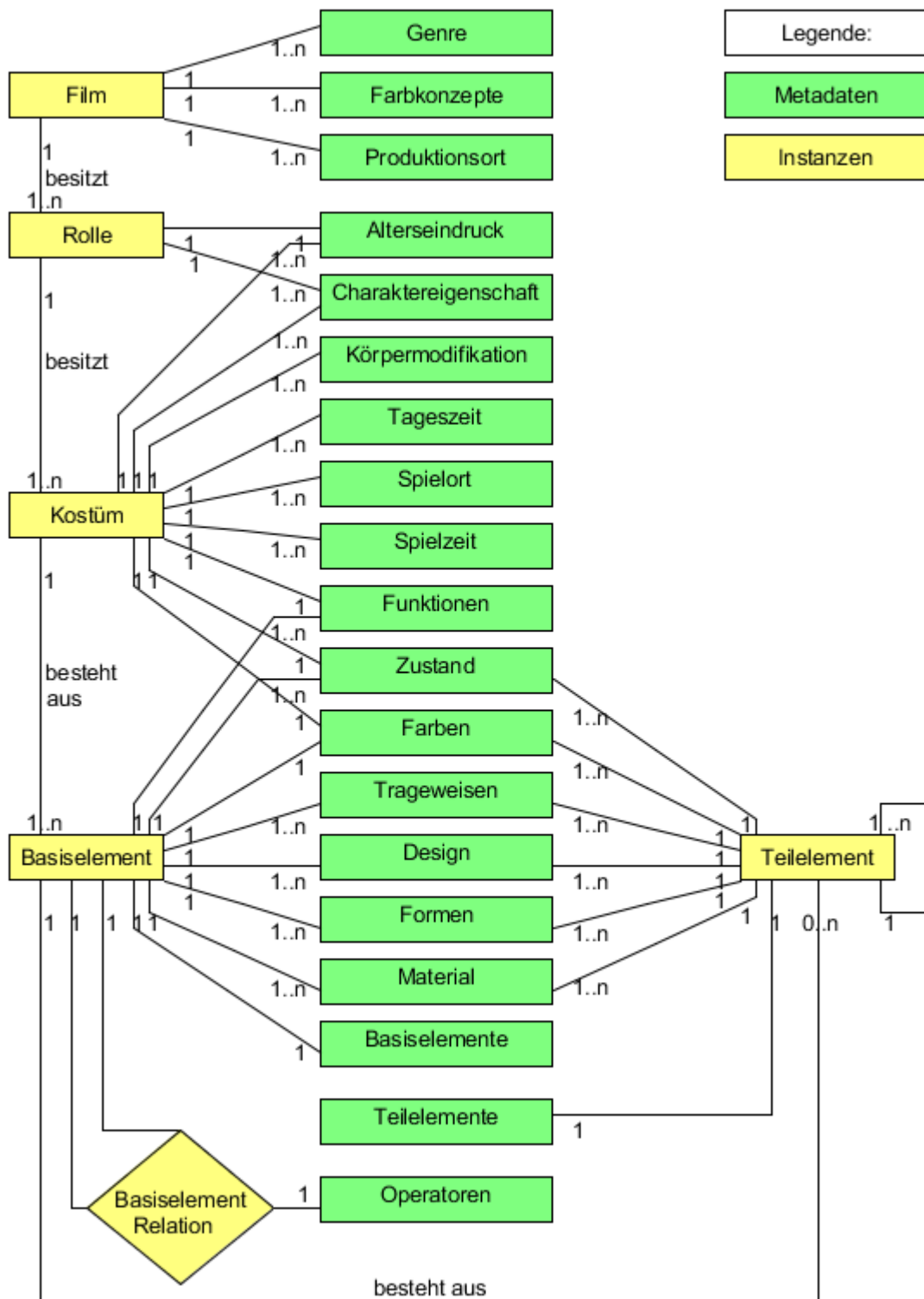


Abbildung 2: ER-Diagramm des Aufbaus der Domäne

3.3.1 Film

Der Film bildet die höchste Hierarchieebene. Er wird näher beschrieben durch die Taxonomien Genre, Farbkonzepte und Produktionsort und besitzt mehrere Rollen.

3.3.2 Rolle

In der Rolle wird die Filmfigur näher spezifiziert. Sie kann eine oder mehrere Kostüme besitzen und verwendet die Taxonomien Alterseindruck und Charaktereigenschaften.

3.3.3 Kostüm

Kostüme bezeichnen die Outfits der Rollen in bestimmten Szenen oder Zeitabschnitten. Es werden der Alterseindruck und die Charaktereigenschaften der Rolle geerbt und es kommen Körpermodifikation, Tageszeit, Spielort, -zeit, Funktionen, Zustand und Farben hinzu.

3.3.4 Basiselement

Die Basiselemente bezeichnen die einzelnen Kleidungsstücke (z.B. "Hose" oder "Mütze"). Sie erben die Funktion, Zustand und Farbe des Kostüms und werden durch die Taxonomien Trageweisen, Design, Formen, Material und Basiselemente ergänzt.

3.3.5 Teilelement

Teilelemente sind die einzelnen Bestandteile der Basiselemente (z.B. Hose) oder anderer Teilelemente (z.B. Knopf oder Reißverschluss). Sie erben die Taxonomien Zustand, Farben, Trageweisen, Design, Formen und Material und werden durch die Taxonomie der Teilelemente vervollständigt.

3.3.6 Basiselement Relation

Eine Basiselement Relation besteht aus zwei verschiedenen Basiselementen und einem Operator. Hier werden zwei Basiselemente angegeben. Der Operator gibt an, wie sie zu einander angeordnet sind. Z.B.: werden Schuhe (1. Basiselement) über (Operator) den Socken(2. Basiselement) getragen.

3.4 Verwendete Taxonomien

In diesem Kapitel werden die im MuSE-Tool zum Einsatz kommenden Taxonomien beschrieben. Sie werden in die Bereiche Basiselemente, Beschaffenheit und Sonstige unterteilt.

3.4.1 Basiselemente

Die Basiselemente sind die Grundbausteine der Kostüme und umfassen alle kostümrelevanten Arten von Bekleidungsstücken und Accessoires (z.B. Mütze, Hose, Schuhe). Die erstellten Taxonomien können aber niemals die Gesamtheit aller möglichen Basiselemente erfassen, da mit jedem neuen Film oder Modeerscheinung die Liste verlängert werden müsste. Die gegenwärtige Taxonomie baut auf einem Bekleidungslexikon⁴, sowie einer lexikalischen Recherche auf den Webseiten Bergfashionlibrary⁵, Amazon⁶, Zalando⁷ und Otto⁸ auf (vgl. [Bar13]).

3.4.2 Beschaffenheit

Um die Basiselemente exakt beschreiben zu können, werden Taxonomien benötigt, welche die einzelnen Basiselemente genauer beschreiben. Unter diesen Taxonomien nehmen die Teilelemente eine besondere Stellung ein. Sie beschreiben die einzelnen Elemente aus denen das Basiselement aufgebaut ist und können jeweils durch weitere Teilelemente beschrieben werden. Z.B.: besitzt eine Hose (Basiselement) eine Gesäßtasche (Teilelement) mit Knopf (Unterteilelement).

Die weiteren Taxonomien sind Form, Material, Farbe, Design, Zustand, Trageweise, Funktion und Körpermodifikation. Ihre Aufgabe ist es, die Basis- und Teilelemente in ihrer jeweiligen Ausprägung genauer zu spezifizieren, damit später Abfragen über die Eigenschaften gemacht werden können und somit Gemeinsamkeiten oder Unterschiede sichtbar werden. Ein mögliches Beispiel ist eine Hose, welche durch die Form kurz, das Material Baumwolle, Farbe blau, Design modern, Zustand abgetragen, Trageweise hochgekrem-pelt, Funktion Sportbekleidung und die Körpermodifikation Muskelbetonung beschrieben wird.

⁴W. Schierbaum (Hg.): Bekleidungslexikon. Mode, Formgestaltung, Schnittkonstruktion, Gradierung, Ausstattung, Zuschnitt, Verarbeitungstechnik, Bügeln, Management und Marketing, Berlin 1993.

⁵www.bergfashionlibrary.com

⁶www.amazon.de

⁷www.zalando.de

⁸www.otto.de

3.4.3 Operatoren

Eine weitere notwendige Taxonomie bilden die Operatoren. Durch sie kann angegeben werden, wie die verschiedenen Basiselemente zu einander angeordnet sind. Durch sie werden aus den einzelnen Basiselementen komplett zusammenhängende Kostüme komponiert. Ein Beispiel ist die Beziehung zwischen T-Shirt und Pullover, der Pullover wird über dem T-Shirt getragen.

3.4.4 Sonstige

Alle weiteren Taxonomien beschäftigen sich nicht mit Basis- oder Teilelementen, sondern beschreiben Spielorte und -zeiten an welchen die Kostüme getragen werden, die Charaktereigenschaften und Alter der Rolle sowie Genre, Produktionsort und Farbkonzept des Films. Verschiedene Möglichkeiten sind z.B. Spielort London, Spielzeit mittags, Charaktereigenschaft böse, Alter jung, Genre Komödie, Produktionsort USA sowie Farbkonzept schwarz/weiß.

3.4.5 Probleme

Bei der Entwicklung des MuSE-Tools zeigten sich zwei Probleme.

Um die erstellten Taxonomien im Werkzeug verwenden zu können, müssen diese zunächst in einer Datenbank abgelegt werden. Die Taxonomien von Hand einzupflegen würde eine sehr große Zeit in Anspruch nehmen (ca. 5400 Einträge). Da sämtliche Taxonomien jedoch bereits in elektronischer Form als Freemind-Dateien gespeichert vorliegen, wurde ein Taxonomy-Parser entwickelt. Dieser extrahiert die Taxonomien und deren Zusammenhänge und erstellt ein MySQL-Script, mithilfe dessen die Taxonomien in einer Sql-Datenbank gespeichert werden können.

Ein weiteres Problem ist die Eingabe. Bei kleineren Taxonomien, wie zum Beispiel den Körperteilen (8 Einträge) kann der Benutzer relativ leicht alle Möglichkeiten vergleichen und den geeignetsten Eintrag auswählen. Bei größeren Taxonomien wie zum Beispiel Basiselemente mit über 1000 Einträgen ist dies jedoch nicht mehr möglich und eine Auswahl über eine Liste sehr ineffizient. Auch eine Suchfunktion würde nur geringe Vorteile bieten. Der Benutzer kann zwar die gewünschten Einträge leichter finden, jedoch kann er nicht alle Einträge mit einander vergleichen. Dies führt dazu, dass der Benutzer auf bereits verwendete Einträge zugreift und eventuell zutreffendere Einträge übersieht. Wird zum Beispiel ein Hemd mit Kragen beschrieben, würde ein unwissender Benutzer eventuell den Stehkragen auswählen, statt den präziseren Ausdruck "Kollar" oder "Vatermörder". Aus

diesem Grund wurde ein TreeSelector entwickelt, in welchem die Taxonomien baumartig dargestellt werden und der Benutzer durch Anklicken die einzelnen Unterbäume öffnen, an- oder abwählen kann.

4 Lösungen der Taxonomieprobleme

Hier werden Lösungskonzepte zur Speicherung der Taxonomien und der verbesserten Auswahl und Selektierung vorgestellt.

4.1 Konzept des MuSE-Tool-Taxonomy-Parsers

Der MuSE-Tool-Taxonomy-Parser arbeitet in mehreren Schritten. Zunächst wird die Freemind-Datei mithilfe einer "Sax-Builder"-Bibliothek eingelesen. Anschließend werden die Daten nach den gewünschten Informationen gefiltert und schließlich als MySQL-Script in einer Datei oder in der Zwischenablage ausgegeben.

4.1.1 Benutzung des Parsers

Der Benutzer gibt in einer Eingabemaske die Quelldatei, den Tabellen-, Element- und Überelementname an, sowie die Ausgabedatei. Sollte die Ausgabe in die Zwischenablage des Betriebssystems erfolgen, ist die letzte Eingabe nicht notwendig. Alle anderen Felder müssen ausgefüllt werden, da das Programm sonst entweder die Eingabedatei nicht findet oder das MySQL-Script unvollständig wird. Anschließend wird der Parser mit Knopfdruck gestartet.

4.1.2 Aufbereitung der Daten

Die eingelesenen Daten des Freemind-Tools müssen erst bereinigt werden. Dies bedeutet, dass unnötige Daten entfernt, sowie die restlichen in ein einheitliches Format gebracht werden, damit anschließend die eigentliche Konvertierung stattfinden kann. Der Parser verarbeitet das XML-Dokument linear und prüft die XML-Struktur sowie die Speicherung der Daten. Dabei wird ein Listenbaum mit den Elementen erstellt.

Im folgenden Listing bezeichnet Vater den Wurzelknoten, Kind das Blatt.

Listing 1: XMLReader.java

```
1 MyNode XMLRekursion(Node node) {
2     switch (node.getNodeName()) {
3         case ''unwichtig'':
4             return null;
5         case ''wichtig'':
6             MyNode Vater = new MyNode(node.getText());
7             for (Node Kind : node.getChildNodes()) {
8                 MyNode KindNode = XMLRekursion(Kind)
9                 if (Kind != null) Vater.addKind = KindNode;
10            }
11            return Vater;
12        }
13    }
```

4.1.3 Parsing der Daten

Der Parser verarbeitet den erstellten Listenbaum rekursiv und erstellt dabei die SQL-Skripte. Im folgenden Codeabschnitt wird der rekursive Abstieg abstrakt dargestellt.

Listing 2: SQLParser.java

```
1 BaumRekursion (Vater) {
2     for (Kinder von Vater) {
3         erstelleSQLScript (Vater.Text, Kind.Text);
4         BaumRekursion (Kind);
5     }
6 }
```

4.1.4 Abbildung in der Datenbank

Es wurden verschiedene Möglichkeiten in Erwägung gezogen die Baumstruktur in der Datenbank relational abzubilden.

Bei dem "Nested Sets" Prinzip behilft sich das Modell mit einem einfachen System: Jedes Element des Baumes erhält zwei Werte - einen linken und einen rechten -, die fortlaufend durch die gesamte Struktur bei jedem Schritt um eins erhöht werden. Zur Veranschaulichung kann man sich einen Wurm

vorstellen, der - an der Wurzel beginnend - den gesamten Baum durchwandert. Der Wurm hinterlässt bei jedem Knoten, den er besucht, eine Nummer und erhöht anschließend seinen Zähler um eins. Dabei wird zunächst immer erst der linke Wert gesetzt. Gelangt der Wurm an das Ende eines Astes (Blatt), springt der Wurm von der linken auf die rechte Seite und wandert von dort weiter zur linken Seite des nächsten Knotens. Folgen keine weiteren Blätter oder Äste, geht er wieder zurück zum letzten übergeordneten Knoten, bis er schließlich wieder an der Wurzel des Baumes ankommt (vgl. [Kle]). Das Problem hierbei liegt jedoch in der fehlenden Flexibilität des Baumes. Während das Ändern eines Blattes kein Problem darstellt, gestaltet sich das Hinzufügen oder Entfernen als eine aufwändige Prozedur. Wird ein Knoten hinzugefügt, oder gelöscht, müssen alle nachfolgenden Knoten angepasst werden, da ihre Werte nicht mehr korrekt sind.

Eine weitere Möglichkeit wäre den Baum mithilfe von 2-Wertepaaren zu speichern. Dabei wird jeder Eintrag einzeln und jede Beziehung als Vater-Kind-Tupel gespeichert. Allerdings ist hier mit großer Redundanz zu rechnen, da die meisten Einträge mehrfach vorkommen, sowohl als Knoten und in der Vater-Kind-Beziehung.

Die Redundanz kann jedoch einfach verringert werden, indem anstatt Vater-Kind-Beziehungen zu jedem Kind der Vater gespeichert wird. So fallen Einträge für Werte und für die Beziehungen zusammen und der Speicheraufwand verringert sich um ca. 30% im Vergleich zu den Vater-Kind-Beziehungen.

4.2 Konzept des TreeSelectors

Um dem Benutzer die Spezifikation von Kostümen zu erleichtern und damit die Eingabe zu beschleunigen, wurde ursprünglich die Baumstruktur der Taxonomien entwickelt. Die Baumstruktur erlaubt es, dem Benutzer schnell den gewünschten Taxonomieeintrag zu finden, ohne die komplette Taxonomie durchsuchen zu müssen, in dem er sich an den einzelnen Knoten hinunter orientiert. Allerdings kostet es sehr viel Zeit, jedes mal mit Hilfe des Freemind-Programms die Taxonomien zu durchsuchen. Mit dem TreeSelectors kann der Benutzer im MuSE-Tool die Taxonomien anzeigen und unter Verwendung eines Suchworts den Taxonomiebaum filtern.

4.2.1 Verwendung des Tree-Selectors

Ein Klick auf den  Button öffnet die Baumstruktur.

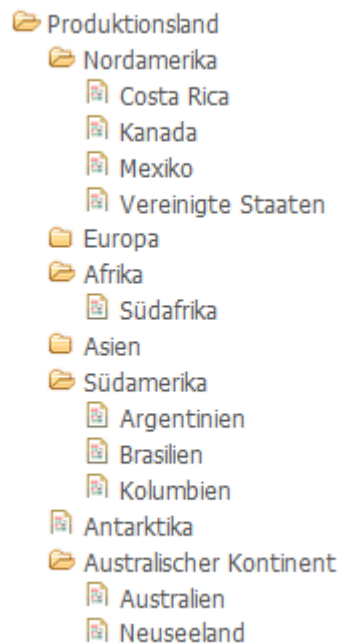


Abbildung 3: Beispiel Baum der Taxonomie Produktionsorte

Hier kann der Benutzer nun durch Anklicken der Ordner den Baum öffnen oder wieder schließen. Bereits eingetragene Werte im Select-Feld werden farbig hervor gehoben. Der Benutzer kann durch Anklicken einen Wert hinzufügen, dabei wird dieser ebenfalls farbig hervorgehoben oder, wenn er bereits eingetragen ist, diesen wieder entfernen.

4.2.2 Implementierung

Der Tree-Selector besteht aus zwei Hauptkomponenten `treeSelect` und `treeModel`.

Das `treeSelect` besteht aus einem Button, welcher die Baumstruktur öffnet und einem mit dem Baum verknüpften Select-Feld.

Das `treeModel` zeigt den eigentlichen Baum.

Listing 3: Baumstruktur

```
1 <ul>
2   <li Wiederhole für jeden Kindknoten der Wurzel>
3     <i>Button zum Öffnen des Unterbaums</i>
4     <i>Knoten</i>
5     <div data-angular-treeview='true'
6         data-tree-id='{{ id }}'
7         data-tree-model='treeModel'></div>
8   </li>
9 </ul>
```

Hier wird im HTML-Format zunächst eine Liste erstellt. Anschließend wird für jeden Kindknoten des Wurzelements ein Eintrag hinzugefügt (vgl. Zeile 2). Jeder Eintrag besteht aus einem Button zum Öffnen des Unterbaumes, dem Textwert des Knotens und dem Unterbaum, welcher ein neues treeModel beinhaltet (vgl. Zeile 3-7).

5 Evaluation

Die Leistungsfähigkeit und der daraus resultierende Nutzen der entwickelten Programme wird hier zusammengefasst.

5.1 Leistungsfähigkeit des MuSE - Tool - Taxonomy - Parsers

Aus dem oben beschriebenen Implementierungsansatz ergibt sich eine lineare Laufzeit, da alle Daten immer genau zwei Mal durchlaufen werden, zunächst um den Listenbaum zu erstellen und anschließend um aus dem Baum die SQL-Skripte zu erstellen. Das Erstellen des Baumes und der SQL-Skripte benötigt einen vernachlässigbaren Zeitaufwand, da diese unabhängig von der Größe der eingelesenen Datei sind.

Es wurden folgende Messungen durchgeführt, um die Leistungsfähigkeit des Parsers festzustellen. Dazu wurden jeweils drei Werte gemessen und der Median verwendet.

10 Knoten	5 ms	0.5 ms/Knoten
100 Knoten	15 ms	0.15 ms/Knoten
1 000 Knoten	150 ms	0.15 ms/Knoten
10 000 Knoten	2 600 ms	0.26 ms/Knoten
100 000 Knoten	427 500 ms	4.275 ms/Knoten

Typische Mindmapgrößen liegen zwischen 10 und 200 Knoten. In extremen Fällen (z.B.: Basiselemente) können Größen von 1000 Knoten erreicht werden.

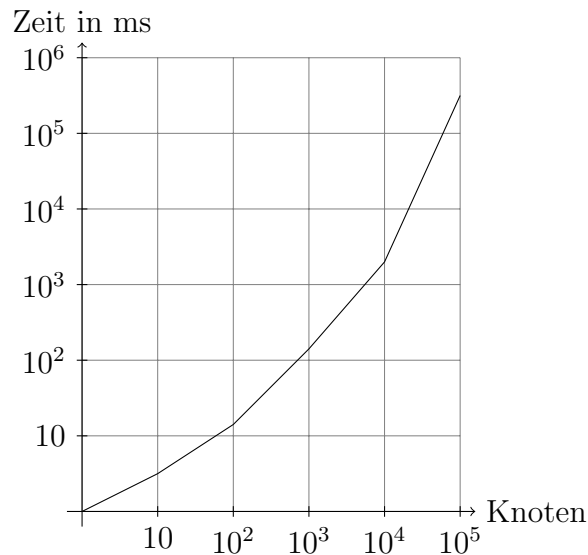


Abbildung 4: Laufzeit des MuSE - Tool - Taxonomy - Parsers

Es zeigt sich, dass die Laufzeit nicht exakt linear verläuft. Dies ist damit zu erklären, dass ab einer bestimmten Baumgröße und -tiefe Teile des laufenden Programms in höhere Caches ausgelagert werden müssen und diese einen größeren Lese- und Schreibaufwand besitzen.

5.2 Änderbarkeit des MuSE-Tool-Taxonomy-Parsers

Bei der Entwicklung des MuSE-Tool-Taxonomy-Parsers wurde darauf geachtet, eine möglichst lose Kopplung zwischen den einzelnen Modulen zu erzielen. Dadurch lassen sich die einzelnen Module leicht ändern oder austauschen. Eine Anpassung auf ein anderes Mindmap-Tool ist somit möglich,

es muss lediglich ein neuer geeigneter Frontend-Reader entwickelt werden, der das Format der einzulesenden Datei versteht und diese in ein für das Programm geeignetes Format umwandeln kann.

5.3 Vorteile des TreeSelectors

Der TreeSelector wurde mithilfe eines Expertentests evaluiert. Dabei haben Experten typische Arbeitsszenarien durchgespielt und die Usability jeweils mit und ohne TreeSelect bewertet. Die Bewertung durch die Experten ergab mehrere Vorteile.

Zum einen wurde die Arbeitsgeschwindigkeit, vor allem für neue Kostümierungen deutlich erhöht, da die gesuchten Werte der Taxonomien wesentlich schneller gefunden werden konnten.

Zum anderen wurde die Bandbreite an verwendeten Taxonomiewerten erhöht, weil die Benutzer alle zur Verfügung stehenden Werte betrachteten und nicht nur die bereits dem Benutzer bekannten Werte verwendeten. Diese Vorgehensweise ergab genauere Lösungen für den Patternautor wie von ihm gefordert.

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

In dieser Bachelorarbeit wurde ein Werkzeug zur Datensammlung von Filmkostümierungen und Patternentwicklung um ein Einlesewerkzeug und eine Eingabehilfe erweitert. Grundlagen zu Patterns, deren Entwicklung, Taxonomien und Ontologien wurden vorgestellt und erklärt, welche für das Verständnis der weiteren Arbeit notwendig sind. Als nächstes wurde genauer auf die fachlichen Anforderungen und den eigentlichen Zweck des MuSE-Tools eingegangen. Anschließend wurden die verschiedenen Benutzertypen des Tools beschrieben, sowie das Datenbankschema erläutert und die verwendeten Taxonomien analysiert. Dabei wurde auf zwei Probleme hingewiesen. Zum Ersten das Eintragen der Taxonomien in die Domäne und zum Zweiten auf die unzureichende Verwendung selbiger. Beide Probleme wurden im nächsten Kapitel analysiert und dazu verschiedene Lösungen vorgestellt. Anschließend wurden die gewählten und umgesetzten Lösungen im Falle des

MuSE-Tool-Taxonomy-Parsers auf seine Leistungsfähigkeit durch verschiedene Messungen und der Nutzen des TreeSelectors von MuSE-Experten mithilfe von Beispieldurchführungen bewertet.

6.2 Ausblick

Im folgenden Abschnitt werden mögliche Erweiterungen zum MuSE-Tool beschrieben, welche die Eingabe erweitern oder verbessern und eine Darstellung zur Auswertung der gesammelten Daten ermöglichen.

6.2.1 Erweiterung und Verbesserungen der Eingabe

Auch wenn durch den TreeSelect bereits eine enorme Zeitersparnis und eine höhere Qualität der Eingaben erreicht wird, kann dieser noch weiter verbessert werden.

Eine Verbesserung der Usability wäre das automatische Öffnen des Taxonomiebaumes bei einer Eingabe im Suchfeld.

Es werden zwar die Sprachen Deutsch und Englisch unterstützt, jedoch wären weitere Sprachen, wenn das Tool international eingesetzt werden soll, von Vorteil.

6.2.2 Darstellung zur Auswertung der gesammelten Daten

Bis jetzt wurde das Tool daraufhin entwickelt, Daten in der Domäne zu sammeln. Im nächsten Schritt sollen die gesammelten Lösungen vom Patternautor analysiert werden. Dazu werden Masken zur Erstellung von Abfragen benötigt. Anschließend müssen die abgefragten Daten in geeigneter Form angezeigt werden. Hier sollte wie beim bereits entwickelte Tool eine einfache und flexible Handhabung im Vordergrund stehen.

Ein letzter Schritt wäre daraufhin eine Eingabehilfe zu schaffen, welche den Patternautor bei dessen Arbeit unterstützt und die Erstellung der Patterns erleichtert und beschleunigt.

7 Anhang A: Verwendete Werkzeuge

Folgende Werkzeuge, die in dem Projekt zur Verfügung stehen, wurden von der Universität Stuttgart bereitgestellt:

- UMLet - UML-Modellierung
- Testsuite-Management(TSM)- Testfallverwaltung

Des weiteren wurden für die Entwicklung weitere folgende Werkzeuge verwendet:

- JetBrains Webstorm - Entwicklungsumgebung für Javascript
- Mozilla Firefox - Test- und Arbeitsumgebung für die Software
- Google Chrome - Test- und Arbeitsumgebung für die Software
- Xampp - Mysql-Datenbank
- Cisco Anyconnect - zur Verbindung zur Onlinedatenbank der Universität Stuttgart
- TexMaker - Erstellen der Tex-Dateien der Dokumente
- mikTex - Compiler für Tex-Dateien
- Tulip - Use-Case-Editor mit Unterstützung für Usability Patterns
- Eclipse - Entwicklungsumgebung für Javaprogramme
- WindowBuilderPro - Plugin für Eclipse für die Erstellung der Benutzeroberfläche

8 Anhang B: Literaturverzeichnis

Literatur

- [Ale77] ALEXANDER, Christopher: *A Pattern Language - Towns, Buildings, Construction*. New York : Oxford University Press, 1977. – ISBN 978-0-195-01919-3
- [Bar13] BARZEN, Johanna: Taxonomien kostümrelevanter Parameter: Annäherung an eine Ontologisierung der Domäne des Filmkostüms / Institut für Architektur von Anwendungssystemen. 2013. – Forschungsbericht
- [GRJV03] GAMMA, Erich ; RICHARD, Helm ; JOHNSON, Ralph E. ; VLISIDES, John: *Design Patterns - Elements of Reusable Object-Oriented Software with Applying Uml and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*. Amsterdam : Addison Wesley, 2003. – ISBN 978-0-582-84442-1
- [Gru93] GRUBER: *A Translation Approach to Portable Ontology specifications*. http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html. Version: Juni 1993
- [Kle] KLEMPERT, Arne: *Nested Sets*. http://www.klempert.de/nested_sets/
- [Wah00] WAHRIG, G.: *Wahrig - Deutsches Wörterbuch*. München : Bertelsman, 2000. – ISBN 3-577-104465
- [wik14] *Digital Humanities*. http://de.wikipedia.org/wiki/Digital_Humanities. Version: Juni 2014

Eidesstattliche Erklärung

Ich erkläre, dass ich meine Bachelor-Arbeit "Repository zur Lösungserfassung in der Domäne von Kostümen im Film" selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe und dass ich alle Stellen, die ich wörtlich oder sinngemäß aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht habe. Die Arbeit hat bisher in gleicher oder ähnlicher Form oder auszugsweise noch keiner Prüfungsbehörde vorgelegen.

Ich versichere, dass die eingereichte schriftliche Fassung der auf dem beigefügten Medium gespeicherten Fassung entspricht.

Stuttgart, den 17.07.2014

(Bernhard Wetzel)