

**Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering**

Konzeption und Implementierung eines Online-Katalogs für Informationsflussmuster

Bachelorarbeit

im Studiengang Informatik

von

Roman Fetsch

**Prüfer: Prof. Dr. Kurt Schneider
Zweitprüfer: Prof. Dr. Rainer Parchmann
Betreuer: M. Sc. Kai Stapel**

Hannover, 10. Juli 2009

Erklärung

Hiermit versichere ich, Roman Fetsch, die vorliegende Bachelorarbeit selbständig, ohne fremde Hilfe und nur unter Verwendung der von mir aufgeführten Quellen und Hilfsmitteln angefertigt zu haben.

Hannover, den 10.07.2009

_____ Unterschrift: Roman Fetsch

Danksagung

Ich möchte mich an dieser Stelle vor allem bei Herrn Kai Stapel herzlich für die ausgezeichnete Betreuung während dieser Arbeit bedanken. Außerdem möchte ich mich natürlich bei allen freiwilligen Testern bedanken, die ihre Zeit geopfert haben, um an dem Usabilitytest teilzunehmen.

Hannover, den 10.07.2009

Inhaltsverzeichnis

1. Einleitung.....	5
1.1. Motivation.....	5
1.2 Aufgabenstellung und Ziel.....	6
1.3 Struktur der Arbeit.....	6
2. Grundlagen.....	7
2.1. FLOW.....	7
2.2. Informationsflussmuster.....	7
2.2.1 Informationsflussmustergruppen.....	8
2.2.2 Informationsflussmusterattribute	8
2.2.2.1 Allgemeine Eigenschaften.....	9
2.2.2.2 Charakteristik.....	10
2.2.2.3 Auswirkung.....	10
2.2.2.4 Diskussion.....	10
2.2.2.5 Klassifikation.....	10
2.3 Informationsflussmusterkatalog.....	11
3. Anforderungen.....	12
3.1 Nachteile des Katalogs in Papierform.....	12
3.2 Vorteil des Online-Katalogs.....	14
3.3 Konkrete Anforderungen an den Online-Katalog.....	15
3.3.1 Usability.....	16
3.4 Technische Anforderungen.....	16
4. Entwurfsentscheidungen.....	17
4.1 Grundlegende Softwarearchitektur.....	17
4.2 Datenhaltung und Logik.....	18
4.3 Präsentation.....	19
4.4 Vollständige Technologieübersicht.....	20
4.4 MDA.....	20
4.5 Benutzer und Rollen.....	22
4.6 Tagcloud.....	23
4.7 Katalogdiagramm.....	24
4.8 Suche.....	24
5. Implementierungsdetails.....	25
5.1 Modell des Informationsflussmusters.....	25
5.2 Modell der Klassifikation.....	26
5.3 Modell der Ansichten für normale Benutzer.....	27
5.4 Modell der Ansichten in der Administration.....	28
5.5 DAO-Pattern.....	30
5.6 Factory-Proxy-Pattern.....	31
5.7 Rich-Text-Editor.....	32
5.8 Farbwähler.....	32
6. Usabilitytest.....	33
6.1 Ziel.....	33
6.2 Aufgabenzusammenstellung.....	33
6.3 Risiko und Gegenmaßnahmen.....	36
6.4 Ergebnis.....	38
6.5 Auswertung.....	38
7. Zusammenfassung und Fazit.....	41
7.1 Ausblick.....	41
Literaturverzeichnis.....	43
Anhang.....	44

1. Einleitung

1.1. Motivation

In der Masterarbeit von Xiaoxuan Ge [XG 2008], die sie 2008 am Fachbereich Software Engineering des Instituts für Praktische Informatik schrieb, beschäftigte sie sich ausführlich mit Informationsflüssen und im speziellen mit Informationsflussmustern.

Im Rahmen dieser Arbeit wurde eine Definition eines Informationsflussmusters und seiner relevanten Eigenschaften getroffen. Basierend auf dieser Definition entstand ein ganzer Katalog mit Informationsflussmustern.

Allerdings liegt dieser nur in seiner Papierform bzw. einer einfachen digitalisierten Form, im PDF-Format, vor. Das bringt entscheidende Nachteile mit sich.

Im einzelnen sind dies:

- **eingeschränkte Navigation durch den Katalog**
 - *man kann nur Vor- und Zurückblättern*
- **eingeschränkte Suche**
 - *man muss zum Teil immer wieder nachschlagen wo sich welches Muster befindet wenn man die Seitenzahlen nicht gerade auswendig gelernt hat*
- **Änderungen und Erweiterungen sind nur mit sehr viel Aufwand verbunden bzw. zum Teil überhaupt nicht durchführbar**
 - *neue Muster könnte man noch im Katalog mit einheften. Aber wenn sich etwas an der Klassifikation ändert oder entfällt, muss dies in allen davon betroffenen Mustern des Katalogs ändern*
 - *Erweiterungen eines Musters sind überhaupt nicht durchführbar wenn man keinen Platz mehr auf der entsprechenden Seite hat*
 - *Im schlimmsten Fall muss der Katalog neu gedruckt werden*
- **Sortierung der Muster kann man nur ein einziges Mal vornehmen**
- **Weitergabe des Katalogs an dritte ist sehr aufwändig**
 - *dies betrifft vor allem den Papier-Katalog, nicht so sehr die digitalisierte Form*

1.2 Aufgabenstellung und Ziel

Ausgehend von dem bestehendem Katalog für Informationsflussmuster soll ein neuer Katalog in digitaler Form erstellt werden.

Die Kernaufgabe dieser Arbeit ist der Entwurf und die Implementierung eines Online-Katalogs für Informationsflussmuster. Ziel dabei ist es die Nachteile des Katalogs in Papierform fest zu stellen und schließlich zu beseitigen.

Hauptaugenmerk liegt dabei auf der Usability [Definition siehe Kapitel 3.3.1] und Erweiterbarkeit des Katalogs.

Um zu überprüfen und nachzuweisen ob das Ziel, also eine Verbesserung der Usability, erreicht wurde, habe ich mich entschlossen zusätzlich ein Usabilitytest mit freiwilligen Testpersonen durchzuführen.

Bei diesem Test wird ein Vergleich zwischen dem einfach, digitalisiertem und dem Katalog online angestrebt

1.3 Struktur der Arbeit

Die Ausarbeitung besteht aus 7 Kapitel. Sie sind aufeinander aufbauend angeordnet.

- **Kapitel 1. Einleitung**

In diesem Kapitel wird die Motivation dieser Arbeit und die damit verbundenen Ziele aufgezeigt

- **Kapitel 2. Grundlagen**

Zu Beginn der Arbeit wird nochmal zusammenfassend geklärt, womit sich FLOW beschäftigt und wie ein Informationsflussmuster aussieht, damit man besser verstehen kann wozu man überhaupt ein Katalog braucht und wie man Muster darin repräsentiert.

- **Kapitel 3. Anforderungen**

Basierend auf der Aufgabenstellung und den theoretischen Grundlagen ergeben sich bestimmte technische und weitere Anforderungen die Usability betreffend an den Online Katalog. Sie alle werden in diesem Kapitel behandelt.

- **Kapitel 4. Entwurf**

Die getroffenen Entwurfsentscheidungen und deren Hintergründe werden in diesem Kapitel aufgezeigt und erklärt.

- **Kapitel 5. Implementierungsdetails**

Ausgehend von den Anforderungen und dem Entwurf wurden spezielle Werkzeuge für die Implementierung ausgewählt. In diesem Kapitel werden alle relevanten, eingesetzten Techniken beschrieben.

- **Kapitel 6. Usabilitytest**

Um die entstandene Anwendung auf die erreichten Ziele hin zu überprüfen und auch nachzuweisen, dass die Anwendung tatsächlich der Aufgabenstellung gerecht wird, wurde eine Reihe von Tests mit dem Online-Katalog durchgeführt. Die Ergebnisse und die Auswertung finden Sie hier.

- **Kapitel 7. Zusammenfassung und Fazit**

Der Inhalt der kompletten Arbeit wird an dieser Stelle nochmal zusammenfassend dargelegt.

2. Grundlagen

2.1. FLOW

FLOW setzt sich allgemein mit der Analyse, Modellierung und Optimierung von Informationsflüssen [XG 2008 Kapitel 2.2.2], die innerhalb von Softwareprojekten „fließen“, auseinander. Ziel ist es Probleme in Softwareentwicklungsprozesse früher zu erkennen und zu verbessern.

Da sich diese Arbeit hauptsächlich mit der Entwicklung eines Katalogs für Informationsflussmuster beschäftigt, möchte ich nicht zu ausführlich in die Theorie von FLOW einsteigen und mich nur auf die Informationsflussmuster beschränken.

2.2. Informationsflussmuster

Die Pfade bzw. Informationsflüsse auf denen sich Informationen in einem Projekt bewegen, weisen bestimmte und wiederkehrende Regelmäßigkeiten, die innerhalb von einander unabhängigen Softwareprojekten auftreten, auf. Diese Regelmäßigkeiten wurden in FLOW als Informationsflussmuster identifiziert. Xiaoxian Ge hat in ihrer Masterarbeit aus dem Jahr 2008, den Informationsflussmustern [XG 2008 Kapitel 4.1] eine einheitliche Struktur geben. Zudem hat sie in ihrer Arbeit einen Katalog mit den bis dahin bekannten Mustern zusammengetragen.

Da diese gefundenen Muster immer mit den selben Probleme dahinter wiederkehren, lassen sich mit ihrer Hilfe die Informationsflüsse innerhalb der Softwareprozesse, von FLOW-Experten wesentlich leichter analysieren und die gefundenen Lösungskonzepte, dieser Problematiken, auch wiederverwenden.

Auf Basis der definierten Informationsflussmusterattribute lassen sich Analyse und Lösungsfindungen zum Teil sogar automatisieren. Im folgenden möchte ich zusammenfassend auf die Attribute von Informationsflussmustern eingehen.

2.2.1 Informationsflussmustergruppen

Alle Informationsflussmuster werden ihrer charakteristischen Struktur nach in gemeinsamen Mustergruppen [XG 2008 Kapitel 5.2.1] zusammengefasst.

Mustergruppen werden wiederum ihrer Struktur nach bestimmten Mustergruppentypen zugewiesen.

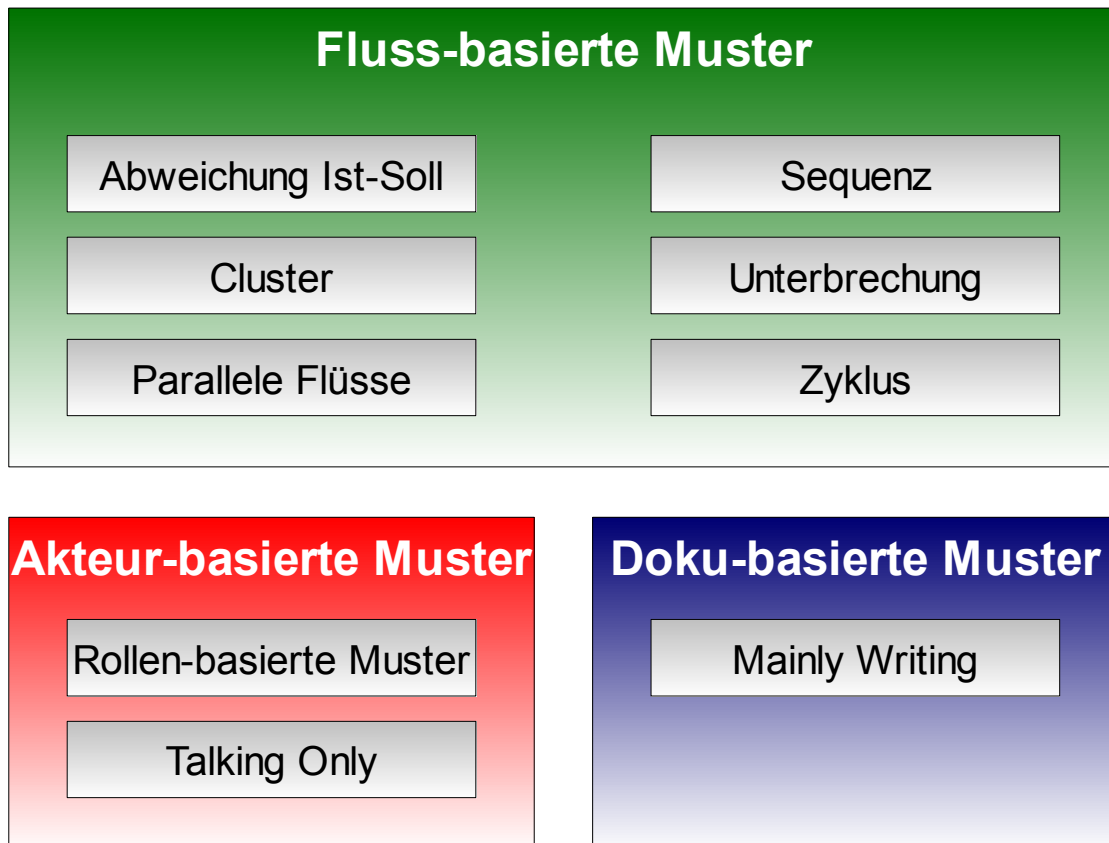


Abb. 1: Diagramm der bisher bekannten FLOW-Mustergruppen und FLOW-Mustergruppentypen

2.2.2 Informationsflussmusterattribute

Alle gefundenen Muster müssen einer festgelegten Schablone nach beschrieben werden. Unter den Attributen [XG 2008 Kapitel 5.2.2] in der Schablone befinden sich solche, die zwingend erforderlich sind, da anderenfalls das gefundene Muster nicht als solches angenommen und in den Katalog aufgenommen werden kann. Diese Pflichtattribute haben sich erst während der Entwicklung dieser Arbeit und bei Gesprächen mit dem Kunden herausgestellt. Es wurden nur Attribute als Mindestanforderung an eine Musterbeschreibung ausgewählt, mit denen man das Muster strukturell erkennt und damit gegen andere Muster abgrenzen kann. Im folgendem möchte ich alle Pflichtattribute mit einem (*) markieren.

Grundsätzlich wird ein Informationsflussmuster über folgende Eigenschaften definiert:

Allgemein	Zu den allgemeinen Eigenschaften zählt unter anderem der Name sowie eine kurze Beschreibung.
Charakteristik	In der Charakteristik soll klar werden wie das Muster konkret aussieht, wie es entsteht und wie man es gegenüber anderen erkennen kann.
Auswirkung	Mit Auswirkung ist die Auswirkung des Musters auf den Informationsfluss als auch auf das Projekt gemeint. Es soll klar werden wie sich das Muster auswirkt und wie stark.
Diskussion	An dieser Stelle kann das Muster nochmal ausführlicher diskutiert werden und man kann auch Beispiele aus der Realität eingehen.

2.2.2.1 Allgemeine Eigenschaften

Mustername (*)	Jedes Muster muss einen möglichst prägnanten Titel haben
Metapher	Ein optionales Attribute, das dazu genutzt werden kann um die Entstehung oder das zentrale Merkmal des Musters bildlich zu beschreiben.
Beschreibung (*)	Das Muster wird zusammenfassend mit der dahinter stehenden Problematik bzw. Lösung beschrieben.
Klassifikation (*)	Das Muster wird mit folgenden Merkmale/Facette formal kategorisiert [XG 2008 Kapitel 5.1]: <ul style="list-style-type: none"> • Form • Struktur • Qualität • Komplexität • Mustergruppe
Verwandtschaftsintensität	Falls das Muster mit anderen, bereits vorhandenen Mustern verwandt ist, kann dies hier zusammen mit der Art der Verwandtschaft angegeben werden. Bisher festgelegte Verwandtschaftsarten sind : <ul style="list-style-type: none"> • sehr verwandt • verwandt • wenig verwandt • nicht verwandt • disjunkt

2.2.2.2 Charakteristik

Struktur (*)	Struktur des Musters wird an dieser Stelle in mit Hilfe der erweiterten FLOW-Notation beschrieben. Dabei soll deutlich hervorgehen warum dieses Muster dominant oder rezessiv ist.
Komplexität	Hier soll die Art der Komplexität und warum das Muster diese aufweist beschrieben und begründet werden.
Mustergruppe	Die Zugehörigkeit zu einer bestimmten Mustergruppe wird an dieser Stelle erläutert.

2.2.2.3 Auswirkung

Qualität	Die Qualität der Auswirkung und die Art der Qualität, die die Auswirkung hat wird an dieser Stelle aufgezeigt.
Form	Die Form der Auswirkung auf ein Projekt wird hier beschrieben.

2.2.2.4 Diskussion

Verweis auf Muster	Ob und wie das Muster in Verwandtschaft zu einem bereits gefundenem anderen Muster steht wird an dieser Stelle erklärt.
Auftreten	Unter welchen Voraussetzungen und wo das Muster auftritt wird hier erläutert.
Beispiel	An dieser Stelle kann man noch ein Beispiel aus der Realität aufzeigen, um das Muster auch in Aktion nachvollziehen zu können.

2.2.2.5 Klassifikation

Jedes Muster wird mit Hilfe einer sogenannten „Facettenklassifikation“ [XG 2008 Kapitel 5.1] formal, zusammenfassend beschrieben. Es gibt bereits festgelegte Facetten, die ebenfalls über einzigartige Symbole erkennbar sind, in denen das Muster nur noch eingeordnet werden muss. Jede Facette spiegelt zusammenfassend ein Merkmal der Charakteristik bzw. der Auswirkung des Musters wieder. Folgende Facetten sind bisher bekannt:

- **Form**
An dieser Facette kann man die Form der Auswirkung des Musters ablesen. Die Auswirkung kann dabei eine zeitlich oder eine quantitative Form [XG 2008 Kapitel 5.1.6] annehmen.

- **Struktur**
Diese Facette gibt die Form der charakteristische Struktur wieder. Die Struktur [XG 2008 Kapitel 5.1.4] kann entweder dominant oder rezessiv sein.
- **Qualität**
Die Qualität [XG 2008 Kapitel 5.1.7] der Auswirkung eines Musters kann negativ, positiv oder kontextabhängig sein.
- **Komplexität**
Die charakteristische Komplexität [XG 2008 Kapitel 5.1.5] kann bisher nur die Form, unitär, linear, mehrdimensional oder kombinatorisch annehmen.
- **Mustergruppe**
Jeder Muster wird aufgrund seiner Eigenschaften bestimmten Mustergruppen zugeordnet. Mustergruppen werden ihrerseits in Kategorien eingeordnet.

Hier noch einmal die Musterschablone mit allen Pflichtattributen:

Allgemein	<ul style="list-style-type: none"> • Mustername • Beschreibung • Klassifikation
Charakteristik	<ul style="list-style-type: none"> • Struktur

2.3 Informationsflussmusterkatalog

In der Praxis hat sich gezeigt, dass mehrere solcher Informationsflussmuster gibt. Damit man aber einen Vorteil aus den gefundenen Mustern ziehen kann, müssen sie entsprechend der „Schablone“ aufgeschrieben und gesammelt werden.

Aus diesem Grund wurde bereits von Xiaoxian Ge ein Katalog mit den bis dahin bekannten Mustern erstellt.

Im Allgemeinen dient dieser Katalog als Nachschlagewerk. Er kann aber dazu genutzt werden um einen Informationsfluss innerhalb eines Softwareprojektes zu analysieren, indem man ein drauf passendes Muster herausucht.

Da ein sich im Katalog befindendes Muster bereits klassifiziert ist, kann man, mit Hilfe des Katalogs, die negativen Aspekte des Musters auf einen Blick erkennen und z.B. ein anderes, „disjunktes“ Muster herausfinden, das vielleicht ein Lösungskonzept zu den Problemen anbietet.

Vorrangig ist also der Katalog vor allem für Projektleiter und Entwickler, die ihre Kommunikation und Dokumentation, mit Hilfe der Informationsflussmuster, auf Probleme analysieren und diese Probleme, mit

Hilfe anderer Muster, verbessern wollen, interessant.

Diese Benutzer, also solchen die nur aus dem Katalog lesen, möchte ich in diesem Zusammenhang als „normal“ definieren.

Neben den „normalen“ Benutzer gibt es nämlich auch „Administratoren“. Administratoren haben ein großes Interesse daran, dass der Katalog weiter entwickelt wird, d.h. das man z.B. neuen Muster oder Mustereigenschaften findet und im Katalog ergänzt bzw. fehlerhafte ändert.

Zu den Administratoren zählen vorrangig alle, die auch am Forschungsprojekt FLOW mitarbeiten.

Allerdings existiert der Katalog bisher nur in seiner Papierform bzw. in einer einfachen digitalisierten (im *.doc und *.pdf Format) Form.

3. Anforderungen

In diesem Kapitel möchte ich zunächst die Nachteile des Katalogs in Papierform aufzeigen. Danach gehe ich auf die daraus resultierenden Anforderungen an den neuen Online-Katalog ein.

3.1 Nachteile des Katalogs in Papierform

Bereits in der Einleitung, im Unterkapitel „Motivation“, bin ich kurz auf die Nachteile des Katalogs in Papierform eingegangen. Zwar hat die einfache digitalisierte Variante, die ebenfalls vorliegt, einige Aspekte stark verbessern, wie z.B. die Suche, jedoch nicht alle. Folgende Nachteile möchte ich in dieser Arbeit verbessern:

- **1. Eingeschränkte Navigation durch den Katalog**

Ein extrem kritischer Punkt hinsichtlich der effizienten Usability [Definition siehe Kapitel 3.3.1] ist die Navigation durch den Katalog. Offensichtlich kann man sich bei dem Papier-Katalog nur in zwei Richtungen bewegen, nämlich vor und zurück.

Das Gleiche gilt auch für den digitalisierten Katalog. Eine Verlinkung zwischen einzelnen Mustern, z.B. basierend auf ihren Verwandtschaften, ist nur mit einigem Aufwand möglich.

- **2. Sortierung der Muster kann man nur einmal vornehmen**

Beim Katalog in seiner Papierform ist die Reihenfolge der Muster nur ein einziges Mal festlegbar. Danach gestaltet sich das Umsortieren sehr schwierig wenn man nicht riskieren möchte den Katalog zu beschädigen.

Diesen Makel kann die digitalisierte Form zwar ausgleichen, allerdings ist dies mit einigem Aufwand, bedingt unter anderem durch die eingeschränkte Navigation, verbunden. Man kann sich z.b. nicht ohne weiteres alle Muster einer bestimmten Mustergruppe mit einer

bestimmten Klassifikation anzeigen lassen, es sei dann bereitet das vorher vor, das wiederum bedeutet, dass man den kompletten Katalog dafür bearbeiten muss. Zudem muss man es allen Katalogbesitzern mitteilen damit alle Kommunikationspartner auf dem selben Stand sind.

- **3. Eingeschränkte Suche**

Eine Suche gestaltet sich mit dem Papier-Katalog als relativ schwierig. Wenn man z.B. nach einem bestimmten neuen Muster sucht, von dem man z.B. nur die Klassifikation kennt, dann muss man im Extremfall den Katalog einmal komplett bis zum letzten Muster durchsuchen.

Vor allem die Suche, wobei hier die Navigation und Einsortierung der Muster wesentlich mit einwirken, ist ein Hauptaugenmerk das sich stark auf die Effizienz der Bedienbarkeit des Katalogs auswirkt.

- **4. Änderungen und Erweiterungen sind nur mit sehr viel Aufwand verbunden**

Erweiterungen und Änderungen im Papier-Katalog sind nicht durchführbar, ohne das der Katalog danach neu gedruckt werden müsste.

Hat man Änderungen vorgenommen müssen erneut alle Personen, mit denen man über Katalogeinträge kommunizieren möchte, erst einen neuen Katalog erhalten damit man die gleiche Basis hat.

Sicherlich hat schon der einfach digitalisierte Katalog in dieser Hinsicht deutlich Vorteile gegenüber dem in der Papierform, da man nicht den Aufwand des Nachdrucks hat. Neue Muster können relativ einfach geändert und neue entsprechend der Schablone ohne weiteres angehängt werden.

Schwieriger wird es wenn sich etwas an einer Facette der Klassifikation ändert oder eine neue Facette hinzukommt. Dann muss man nämlich dies nicht nur an einer Stelle sondern in allen Mustern, die davon betroffen sind, ändern. Erneut wirkt sich dann die Suche und Navigation negativ auf den Aufwand aus.

- **5. Weitergabe des Katalogs an dritte ist sehr aufwändig**

Möchte man den Papier-Katalog weitergeben muss man ihn erst erneut ausdrucken und eventuell binden. Dies ist sehr aufwendig und wird vom digitalisierten Katalog schon wesentlich ausgebessert.

Wenn man aber etwas Inhaltliches im Katalog ändert, müssen alle Katalogbesitzer benachrichtigt werden und man muss ihnen dann die aktuelle Version des Katalogs zukommen lassen.

Noch aufwändiger wird es wenn der Katalog von mehreren Personen geändert wurde, dann müssen alle Änderungen erst synchronisiert werden.

Neue Personen, die potentiell von dem Katalog profitieren oder gar

neue Muster beitragen könnten, muss man erst persönlich kontaktieren um den Katalog dann weiter zureichen.

3.2 Vorteil des Online-Katalogs

Der wesentliche Vorteil des Online-Katalogs ist die Tatsache, dass er von allen Benutzern, mit einem Internetzugang, von überall erreichbar ist. Er bildet damit eine gemeinsame Basis für alle diese Benutzer. Aktualisierungen im Katalog sind sofort, nach deren Umsetzung, von allen abrufbar. Man kann also auf sehr schnellem Weg allen an FLOW interessierten Personen eine Plattform für Informationsflussmuster bereitstellen.

Man geht derzeit davon aus, dass der Katalog mit den bisher entdeckten Mustern noch nicht vollständig ist. Neue Muster werden jedoch vor allem aus gemachten Erfahrungen gewonnen [XG 2008 Kapitel 3.3]. Um also neue Muster überhaupt entdecken zu können muss man erstmal alle alten kennen und aus den Erfahrungen, die man in Projekten gemacht hat, neue Muster erschließen.

Da der Katalog sehr viele außen stehende Personen erreicht, ist die Erfahrungsmenge, die potentiell zur Verfügung steht sehr groß, daher sind auch die Chancen größer, dass jemand ein neues Muster, das noch nicht im Katalog aufgenommen wurde, findet und dann vorschlägt.

Zusammengefasst heißt das :

- **Bequeme Navigation durch den Katalog**
Man kann Muster untereinander durch Links verknüpfen, d.h. falls man mal z.B. von einem unbekanntem Muster, das von einem anderen referenziert wird, liest dann kann man es mit einem Klick erreichen. Im Allgemeinen lassen sich alle Seiten mit einander verknüpfen, sodass jede verwandte Information in kürzester Zeit zu erreichen ist.
- **Sortierung der Muster kann jeder Zeit beliebig geändert werden**
Alle Muster liegen ohne bestimmte Reihenfolge in der Datenbank und können entsprechend der gesuchten Attribute geladen, sortiert und dargestellt werden.
- **Unkomplizierte und erweiterte Suche**
Ein Suche kann nun komplett automatisiert erfolgen. Man kann nicht nur nach einem bestimmten Begriff suchen, sondern auch die Suche durch die Angaben von bestimmten Eigenschaften einschränken.
- **Änderungen und Erweiterungen sind ohne viel Aufwand durchführbar**
Möchte man z.B. eine Klassifikation-Facette ändern ist dies einmal in einer entsprechenden Maske, da der Online-Katalog speziell für solche Fälle konstruiert werden kann, zu machen und die Änderung ist

nach dem Speichern sofort allen zugänglich. Das gleiche gilt für die Erweiterung.

- **Einfache Weitergabe des Katalogs**

Die Weitergabe gestaltet sich sehr einfach, denn man muss nicht den Ganzen Katalog weitergeben sondern nur die URL wo man ihn findet. Allen Nutzern steht die selbe Kommunikationsbasis zur Verfügung.

3.3 Konkrete Anforderungen an den Online-Katalog

Die ersten wesentlichen Anforderungen an den Online-Katalog ergeben sich aus den oben genannten Nachteilen, zusätzlich kommen die Anforderungen, die Kai Stapel als Kunde gestellt hat:

Funktionale Anforderungen :

- Vorhandene Muster müssen sortierbar sein, d.h. der Benutzer[siehe Kapitel 2.3, Absatz 5] muss die Muster nach z.B. der Klassifikation sortieren können [bedingt durch Nachteil 2.]
- Der Benutzer soll ohne großen Aufwand durch alle Muster suchen können, d.h. man muss z.B. durch alle Muster mit Hilfe der Kategorien der Klassifikation suchen können [bedingt durch Nachteil 3.].
- Änderungen und Erweiterungen von Mustern und Mustereigenschaften sollen ohne großen Aufwand durchführbar sein [bedingt durch Nachteil 4.].
- Nicht jeder beliebige Benutzer darf im Katalog Änderungen durchführen.
- Der Katalog soll in eine bereits bestehendes System integriert werden können.
- Ein Benutzer der Katalogs soll neue Muster vorschlagen können. Der Vorschlag geht dann per Mail an einen vordefinierten Empfänger *Mit dieser Anforderung wird versucht den Aufwand, der dafür zuständigen Administratoren, zu minimieren.*
- Ein Muster darf erst angelegt werden wenn alle Pflichtattribute ausgefüllt sind.
- Der Katalog muss über das Internet erreichbar sein, um die Weitergabe zu erleichtern [bedingt durch Nachteil 5.].

Nicht funktionale Anforderungen :

- der Katalog muss gut navigierbar sein [bedingt durch Nachteil 1.]
- der Benutzer soll den Katalog möglichst intuitiv bedienen können [bedingt durch Nachteil 1.]

„intuitiv“ heißt in diesem Zusammenhang, dass der Benutzer z.B. schon aus der Benennung einer Funktionalität auf die Funktion selber schließen kann oder auch das der Benutzer Zusammenhänge der Anwendung erkennt ohne das man es ihm lange erklären bzw. er danach suchen muss.

Sonstige nicht technische Anforderungen :

- Es soll keine neuer Informationsfluss-Editor erstellt werden
Informationsflüsse sollen in einem externen Editor, aufgezeichnet werden und in Online-Katalog nur als Bild integriert werden.

Zusammenfassend wird deutlich das vor allem die Usability[Definition siehe Kapitel 3.3.1] die größte Anforderung an den neuen Online-Katalog darstellt.

3.3.1 Usability

Da die Usability eine der größten Anforderungen darstellt, möchte ich es an dieser Stelle definieren:

Die „Usability“ möchte ich zunächst in dieser Arbeit als „Benutzbarkeit“ nach ISO/IEC 9126 definieren, d.h. die Anwendung ist brauchbar („usable“) wenn der Benutzer ohne größeren Aufwand die Anwendung benutzen kann.

Die Benutzbarkeit ist dann besonders gut wenn der Aufwand der Bedienbarkeit klein ist. Daher möchte ich die Definition um den Begriff der Effizienz erweitern. Effizienz verstehe ich in diesem Zusammenhang als die Zeit, die ein Benutzer, für das Erledigen einer bestimmten Aufgabe mit und in dem Katalog *benötigt*.

Zusammenfassend heißt das, das der Online-Katalog eine bessere Usability gegenüber dem Papier-Katalog besitzt, wenn man für die selbe Aufgabe, die man in beiden Katalogen durchführt, weniger Zeit benötigt.

Die Definition von „Benutzbarkeit“ ist damit noch nicht abgeschlossen, denn eine Anwendung muss auch „attraktiv“ sein und eine gute „Konformität“ aufweisen, d.h. möglichst viele der Anforderungen, die an die Oberfläche gestellt werden, erfüllen. Die zentrale Anforderungen gilt aber der effizienten Bedienbarkeit, deshalb möchte ich sie auch in den Fokus der Usability rücken.

3.4 Technische Anforderungen

Folgende technischen Anforderungen haben sich in den Gesprächen mit dem Kunden ergeben bzw. wurden aus den angegebenen Gründen festgelegt:

Anforderungen mit dem Ziel die Usability und Navigation zu verbessern:

- Kein Javascript im Benutzer-Front-End, ausgenommen in der Administration

jeder Benutzer der Javascript aus Sicherheitsgründen in seinem Browser deaktiviert hat, soll trotzdem den Katalog nutzen können.

- Läuft auf Firefox und IE äquivalent, Darstellung also erfolgt in einem Browser
Damit der Katalog auch von einem möglichst großem Personenkreis erreicht werden kann
- „Tagcloud“ für alle bestehenden Muster, mit der Häufigkeit der Klicks des Musters als Bezugsgröße
Diese Funktionalität soll vor allem die Navigation verbessern, denn so kann man direkt die „beliebtesten“ Muster erkennen und mit einem Klick zu ihnen gelangen.

Anforderungen, die sich aus der funktionalen Anforderung, der Katalog müsse in einem bestehendem System integriert werden können, ergeben:

- MySql als Datenbank
- PHP 5
- Apache
- Ansprechende Optik, angepasst an das SE-Homepage-Layout

Sonstige technische Anforderungen :

- Entwicklung soll Objektorientiert erfolgen
- Passwörter werden verschlüsselt in der Datenbank abgespeichert

4. Entwurfsentscheidungen

4.1 Grundlegende Softwarearchitektur

Ausgehend von der technischen Anforderung, das die Anwendung auf einem Apache laufen und entweder über Firefox oder den Internet Explorer zugänglich sein soll, ergibt sich automatisch der Rückschluss, dass man eine Client-Server-Systemarchitektur verwenden muss.

Da die Logik hinter dem Online-Katalog keines Wegs trivial ist und auch die Oberfläche, ausgehend von denen an sie gestellten Anforderungen, komplex aufgebaut ist, habe ich eine 3-Schichten-Softwarearchitektur bestehend aus einer Datenhaltung-, einer Logik- und eine Präsentationsschicht für den Online-Katalogs ausgewählt.

Der Vorteil der Aufteilung in die drei Schichten ist, dass man die jede einzelne Schicht, ganz nach dem Prinzip „divide et impera“, jeweils separat entwickeln kann. Diese einzelnen Schichten kommunizieren über vordefinierte Schnittstellen miteinander. Im Zentrum der 3-Schichten-Softwarearchitektur steht die Logikschicht, die zwischen den Datenhaltungsschicht und Präsentationsschicht vermittelt.

Die Funktionen jeder einzelnen Schicht ergeben sich schon aus der Benennung, wo jedoch die jeweilige Schicht liegt, also ob auf dem Client oder auf dem Server, wird in den folgenden Kapiteln noch geklärt.

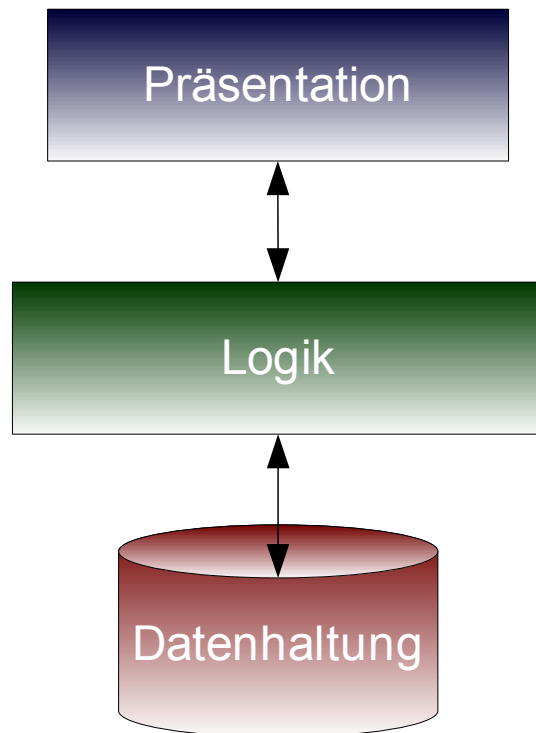


Abb. 2 : Schematische Übersicht über die grundsätzliche Softwarearchitektur

4.2 Datenhaltung und Logik

Für Datenhaltung ist eine MySQL-Datenbank, die serverseitig verwaltet wird, zu wählen.

Für die Logik ist PHP 5 zu verwenden ist. Ausgehend von der gewählten Softwarearchitektur bedeutet das, dass auch alle Create-Read-Uppdate-Delete-Operationen (CRUD) über PHP 5 und damit serverseitig ausgeführt werden.

4.3 Präsentation

Da der Katalog über einen Browser bedient werden soll, muss die Ausgabe in HTML beschrieben werden. Das Entwickeln der Oberfläche in reinem HTML bzw. mit PHP ist relativ aufwändig, zeitraubend und außerdem erfordert es viel Disziplin um die Logik von der Präsentation, die beide dann in PHP geschrieben wären, auseinander zu halten. Allerdings ist die Entwicklungszeit von Beginn dieses Projektes an begrenzt.

Um möglichst schnell und flexible entwickeln zu können, fiel die Entscheidung eine andere Template Engine, die für die Entwicklung von Oberflächen konzipiert wurde, für die Präsentationsschicht zu verwenden. Zusätzlich muss diese Template Engine natürlich eine Schnittstelle zu PHP besitzen, damit die Präsentationsschicht auch unkompliziert mit der Logik kommunizieren kann.

Smarty ist eine solche Template Engine, die eine eigene Syntax in ihren Templates verwendet und mit ihrer Hilfe dann HTML generiert. Zudem existiert eine eigene Smarty-PHP-Klasse, die als Schnittstelle zwischen der Logik in PHP und der Darstellung dient.

Dadurch wird eine Trennung zwischen Präsentation und Logik möglich. Ähnlich wie PHP werden die Templates von Smarty serverseitig von einem Compiler übersetzt und in einem Cache zwischen gespeichert. Die Nähe zu PHP und die bereits vordefinierte Schnittstelle, legen es nahe Smarty für die Präsentation zu verwenden.

Es sei noch erwähnt das Javascript zwar für die Oberfläche der normalen Benutzer, also für Benutzer die sich nur Muster angucken oder nach Mustern suchen, in einer Anforderung verboten wird, es kommt jedoch nach Rücksprache mit dem Kunden in der Administration, also in den Oberflächen wo neue Daten in den Katalog eingegeben bzw. bestehende geändert werden, zum Einsatz, um mit einigen in Javascript geschriebenen Komponenten die Bedienbarkeit zu verbessern.

4.4 Vollständige Technologieübersicht

Zusammenfassend mit den vorhergegangenen Kapiteln zum Entwurf und der Anforderungen einen Apache-Server zu benutzen ergibt sich folgende vollständige Übersicht über alle eingesetzten Technologien :

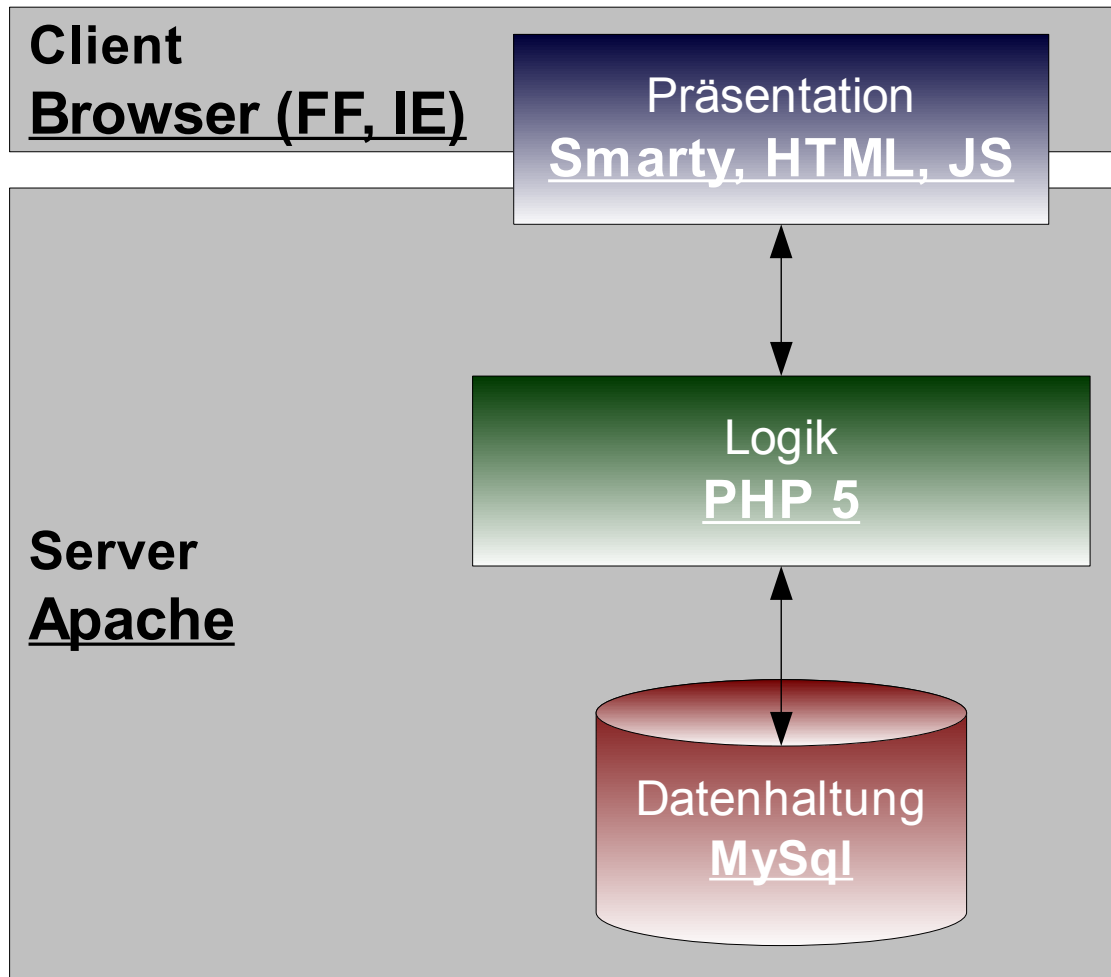


Abb. 3: Vollständige, schematische Übersicht der Softwarearchitektur

In der obigen Abbildung ist die Präsentationsschicht mit Absicht zwischen der Server- und Clientseite positioniert. Denn im Gegensatz zu PHP und Smarty werden die Javascript-Komponenten erst beim Benutzer, vom Browser ausgeführt.

4.4 MDA

Ein großes Problem stellt die Anforderung an die Flexibilität in Bezug auf Erweiterbarkeit der internen Struktur dar. Wie schafft man es also den Katalog nach den bisher Entwurfsentscheidungen möglichst schnell zu entwickeln und dabei trotzdem flexible auf Änderungen, wie z.B. neue Facetten in der Klassifikation, zu reagieren? Eine Antwort darauf ist die modellgetriebene Softwareentwicklung. Man verschiebt dabei den Fokus der

Entwicklung auf ein im UML beschriebenes Modell. Mit Hilfe eines Generators übersetzt man dann das erstellte Modell in Code transformieren. Das bedeutet man abstrahiert und bildet die Anwendung in einem Modell ab. Es soll dabei möglichst viel des notwendigen, dabei möglichst kein Codes generiert werden.

Ein guter Generator sollte alle für die Kommunikation mit der Datenhalten bzw. auch die modellierte Datenschicht erstellen damit sich der tatsächliche Entwicklungsaufwand danach möglichst nur noch auf die Logik konzentriert werden muss. Im speziellen wird also ein Model-Driven-Architecture-Ansatz (MDA) [OMG Kapitel 2.1] gesucht, der aber auch die schon vorher getroffenen Entwurfsentscheidungen berücksichtigt, d.h. das Modell soll mit allen CRUD-Operationen in PHP, die Datenhaltung dabei in MySQL- und die modellierten Oberflächen in Smarty- Syntax Code übersetzt werden.

Tatsächlich existiert bereits ein solcher Ansatz, der auch genau die eben gestellten Anforderungen erfüllt, namens Acceleo [ACC].

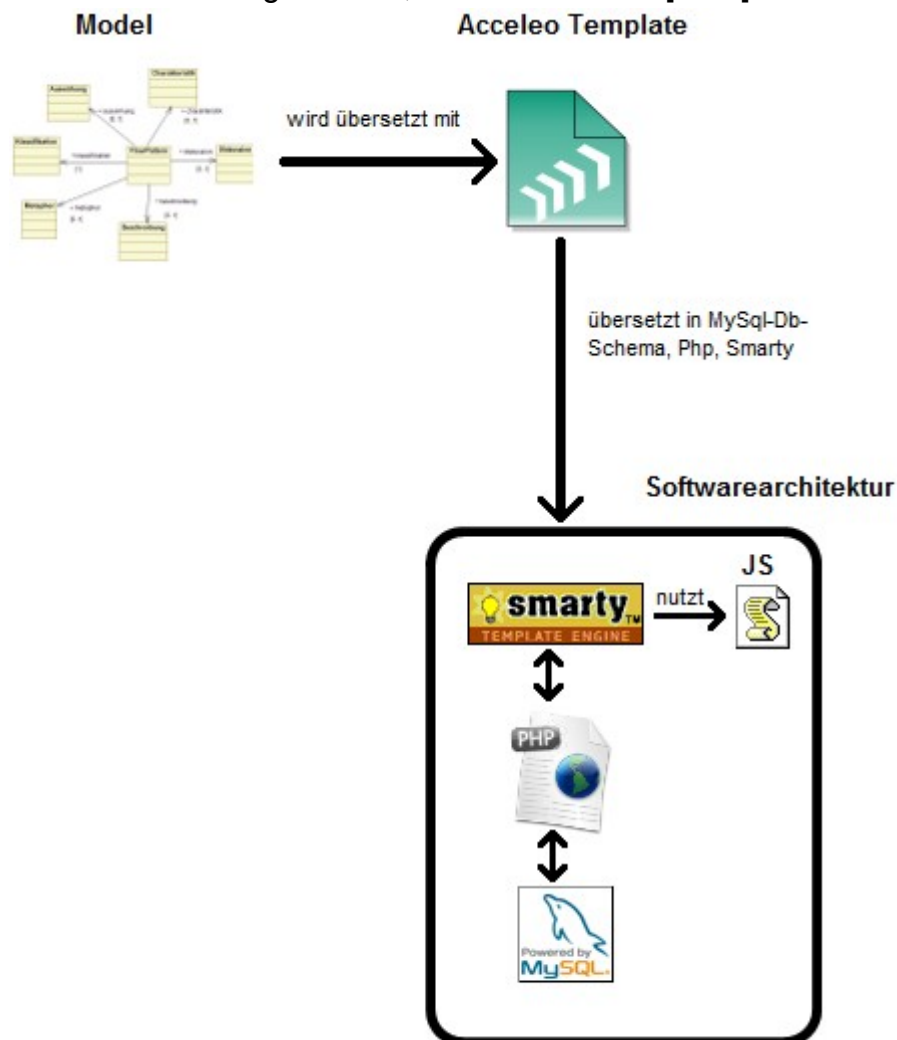


Abb. 4: Einfache Abbildung des Entwicklungspfad vom Modell zum Code

Acceleo transformiert das UML-Modell mit Hilfe von Templates, die in einer eigenen Syntax geschrieben sind und generiert dann so den Programmcode.

Allerdings stellt Acceleo auch ein Risiko dar, denn das PHP Modul befindet sich noch in einer Entwicklungsphase, d.h. im schlimmsten Fall wird ein Teil des Modells nicht richtig übersetzt.

Für diesen Fall stellt Acceleo zusätzliche Werkzeuge zur Verfügung, mit denen man die Acceleo Templates bearbeiten bzw. eigene neu hinzufügen kann. Das heißt die schlimmste Konsequenz aus dem Risiko wäre zusätzlicher Aufwand, der vielleicht das Projekt zu lange verzögern würde, in dem man sich in die Werkzeuge von Acceleo und die Syntax der Templates einarbeiten muss um die gewünschten Funktionalitäten zu ergänzen.

Das Risiko erscheint jedoch vertretbar in Hinblick auf die beschleunigte Entwicklungszeit, da das Genrat genau die geforderte Technologien einhält. Zudem wird Acceleo auch weiter entwickelt, womit die Aussicht darauf besteht, dass eine Funktionalität, die es noch nicht gibt, beim nächsten Update mit dabei ist, wenn man denn diese auch in den Entwicklern meldet.

4.5 Benutzer und Rollen

Die Anwendung wurde in zwei Bereiche aufgeteilt, den Bereich für normale, „anonyme“ Benutzer und den Bereich für die Administratoren [siehe Kapitel 2.3 ab Absatz 5].

Aus einer der Anforderungen geht hervor, dass nicht jeder Benutzer jedes Element online im Katalog ändern darf. Aus dem Grund habe ich die Idee entwickelt Benutzerrollen einzuführen. Ziel dabei ist, ein gewisses Maß an Sicherheit bei der Administration von Mustern, Facetten und anderen Benutzern zu gewährleisten. Jeder der Benutzerrollen besitzt unterschiedlich viele Rechte. Konkret wurden die Rollen hierarchisch, mit immer größer werdenden Befugnissen, angeordnet.

Folgende Rollen wurden dabei festgelegt :

- **Musteradministrator**
Benutzer mit dieser Rolle können neue Informationsflussmuster anlegen und bestehende verändern.
- **Strukturadministrator**
Ein Benutzer mit der Rolle „Strukturadministrator“ kann sowohl alle Muster administrieren als auch neue Strukturelemente anlegen sowie bestehende ändern. Zu den Strukturelemente zählen alle Facetten der Klassifikation, als auch die Arten der Verwandtschaftsintensitäten, d.h. ein Strukturadministrator kann z.B. neue Arten von Formen der Auswirkung in die Klassifikation aufnehmen.
- **Superadministrator**
Dieser Benutzertyp hat die größte Mächtigkeit aller Benutzerrollen. Er kann sowohl Muster, als auch Strukturelemente als auch alle registrierten Benutzer administrieren, d.h. er kann als einzige Benutzerrolle die Rollen, Passwörter, etc. anderer Benutzer ändern.

Zusammengefasst die Rangfolge der Benutzer:

**normale Benutzer < Musteradministrator < Strukturadministrator
< Superadministrator**

Ein Benutzer kann also erst dann neue Muster eintragen oder bestehende ändern, wenn er als Benutzer des Katalogs registriert ist und ihm von einem „Superadministrator“ eine entsprechende Rolle zugewiesen wurde.

Die normalen Benutzer können also nur aus dem Katalog lesen, und die Administratoren entsprechend ihren Befugnissen Elemente löschen, ändern und neue hinzufügen. Innerhalb der Administration gibt es demnach noch einmal eine Unterteilung der Anwendung, d.h. je Benutzerrolle einen eigenen Bereich.

4.6 Tagcloud

Die sogenannte Tagcloud, oder auch „Schlagwortwolke“, hat sich erst während der Entwicklungsphase als eine Anforderung herausgestellt. Sie soll auf der Startseite alle Muster, die bisher angelegt wurden, darstellen und dabei unterschiedlich hervorheben.

Die Bezugsgröße ist dabei die Anzahl der Klicks eines Musters, d.h. je häufiger ein Muster angeklickt und damit vermutlich auch angeschaut wurde desto größer und deutlicher ist der Name dieses Musters zu erkennen.

Die Einträge der Tagcloud sind zudem auch Links, sodass man als neuer Benutzer z.B. direkt zu den vermeidlich bekanntesten Mustern navigieren kann.

Die Formel [Wiki 1] für das Bestimmen der Schriftgröße lautet wie folgt:

$$f_i = \left\lceil \frac{f_{max} * (n_i - n_{min})}{(n_{max} - n_{min})} \right\rceil \quad \text{für } n_i > n_{min}; \text{ sonst } f_i = f_{min}$$

- f_i : Anzuzeigende Schriftgröße
- f_{max} : maximale Schriftgröße
- f_{min} : minimale Schriftgröße
- n_i : Anzahl Klicks
- n_{min} : minimale Anzahl Klicks
- n_{max} : maximale Anzahl Klicks

4.7 Katalogdiagramm

Das Katalogdiagramm wurde mit dem Ziel entworfen die Navigation und Bedeinarkeit zu verbessern. Konkret soll mit dem Diagramm der normale Benutzer mit nur einem Blick die Verwandtschaften zwischen den einzelnen Mustern erkennen. Wie aus der den Grundlagen schon hervorgeht werden alle Muster in bestimmten Mustergruppen zusammengefasst, d.h. Muster, die sich ähnlich sind, sind auch in ähnlichen Mustergruppen. Deshalb muss das Katalogdiagramm also alle vorhandenen Muster und ihre Einordnung in den Mustergruppen aufzeigen, um das gestellten Ziel zu erreichen. Zudem muss man aus dieser Tabelle direkt zu den jeweiligen Mustern bzw. Mustergruppe gelangen können.

Muster \ Eltern-Muster	Sequenz	Mainly Writing	Cluster	Rollenbasierte Muster	Zyklus
Buerokratie	■	■			
Wichtiges Dokument			■		
Write For One		■			
Hierarchie	■			■	
Experienced Expert			■		
Architekt zu Entwickler				■	■
Interaktion					■

Abb.5: Implementiertes Katalogdiagramm

4.8 Suche

Wie aus den Anforderungen bereits hervorgeht stellt die Suche einen extrem wichtigen Punkt der Anwendung dar. Eine automatisierte Suche ist in Papier-Katalog gar nicht und in der einfachen, digitalisierten Form nur eingeschränkt, auf die Suche nach einem bestimmten Begriff, möglich. Die neue Suchfähigkeit des Online-Katalogs soll deshalb mehr können. Neben der Suche nach Begriffen, innerhalb der ganzen Musterschablone, soll man Anhand bestimmter Mustereigenschaften bereits bestimmte Muster, die dem nicht entsprechen aussortieren. Da die Klassifikation eine Abstraktion der Mustereigenschaften darstellt, bietet es sich an mit ihrer Hilfe die Suche zu beschränken. Jedoch kennt der Benutzer nicht immer sofort alle Merkmale eines gesuchten Musters oder er möchte sich z.B. neue Muster einer bestimmten Mustergruppe angucken, daher muss auch die Mustergruppe und die Kategorie der Mustergruppe mit in die Suche einbezogen werden. Um den Benutzer eine gute Bedienbarkeit anzubieten sollen dann alle Muster in der Ergebnismenge auch schon direkt Links zu den Musterbeschreibungen darstellen, sodass sich der Benutzer mit einem Klick zu einem gefundenem Muster navigieren kann.

Grundlegendes	
Mustername oder Suchbegriff:	<input type="text" value="Write For One"/>
Mustergruppe:	<input type="text" value="Mainly Writing"/>
Mustergruppentyp:	<input type="text" value="Dokubasierte Muster"/>
Klassifikation	
Struktur:	<input type="text" value="dominant"/>
Komplexität:	<input type="text" value="mehrdimensional"/>
Form:	<input type="text" value="zeitlich"/>
Qualität:	<input type="text" value="kontextunhabhaengig"/>
<input type="button" value="search"/>	

Abb.6: Implementierte Suchmaske des Online-Katalogs

5. Implementierungsdetails

In diesem Kapitel möchte ich auf einige Implementierungsdetails eingehen, die wichtig sind um zu verstehen wie die Anwendung im Realen funktioniert. Da das Modell auf keine einzelne Seite passt wurde es aufgeteilt. Zunächst beginne ich mit den Modellen für die Datenschicht und danach für das Präsentationsschicht.

5.1 Modell des Informationsflussmusters

Das vorliegende Modell hält sich an die Musterschablone, wie sie in dem Kapitel 2.2 definiert wurde. Zusätzlich ist hinzugekommen:

- die Klasse „User“ mit der dazugehörigen Rollenklasse „Role“, da jedes Muster auch eine Referenz auf deren Autor speichern soll
- ein neue Attribute „headPictureLink“, das eine Verknüpfung zu einem Titelbild jedes Musters speichern soll
- die Klasse „VisibleKomponents“, die zur Steuerung der Ausgabeseite genutzt wird. In „VisibleKomponents“ kann man festlegen welche Komponenten der Musterschablone tatsächlich in der Ausgabeseite sichtbar sind. Die Notwendigkeit dieser Klasse hat sich erst während der Entwicklungsphase in den Gesprächen mit dem Kunden ergeben.

Der Übersicht halber wurden in diesem Bild, die Attribute und Operationen der einzelnen Musterattribute ausgeblendet, zudem wurden sie auch bereits in der Musterschablone beschrieben.

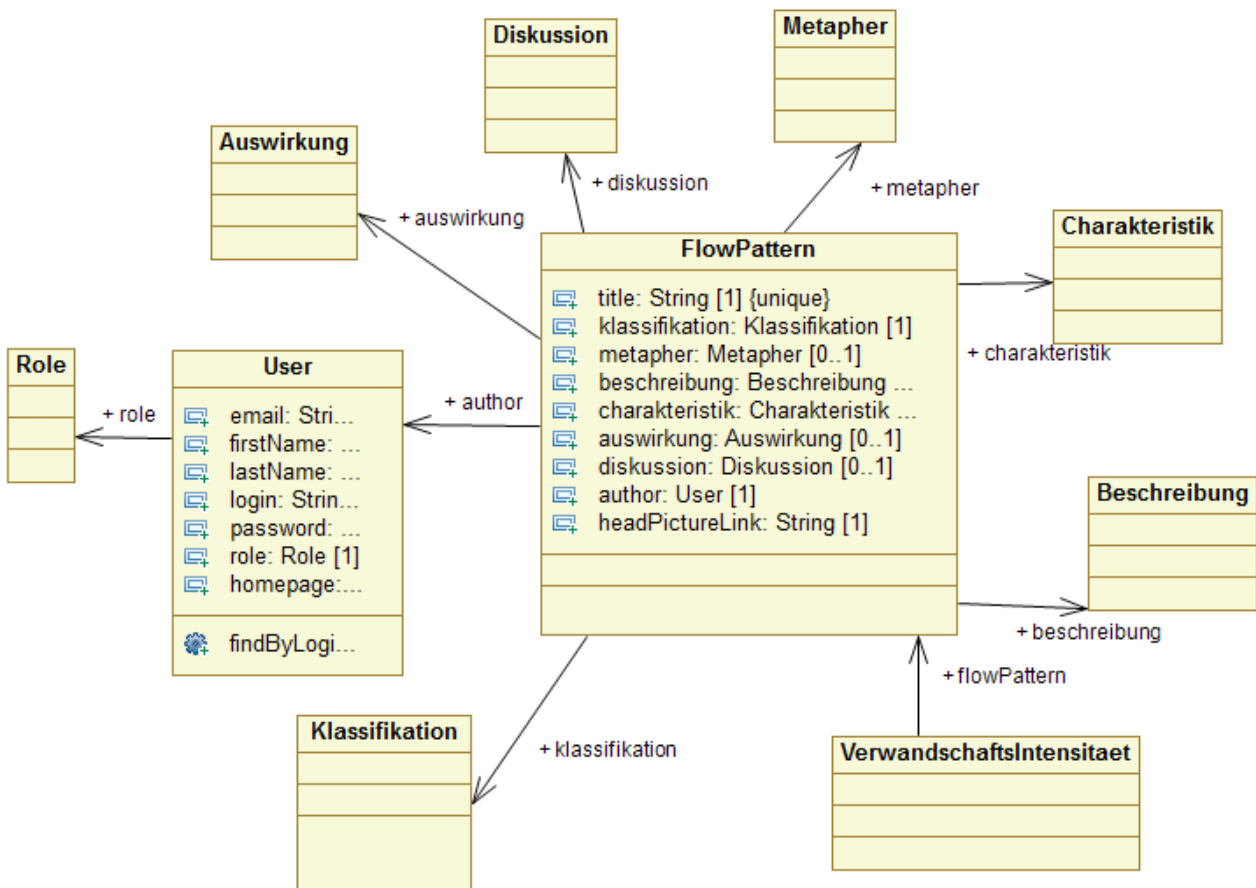


Abb. 7: Klassendiagramm des Informationsflussmusters mit allen seinen relevanten Eigenschaften.

Aus jeder dieser Kästen wird sowohl eine Entität im MySQL-Schema als auch Klasse, die diese Entität repräsentiert, generiert. Im Detail ist sehen Sie oben den Namen, darunter die Attribute, dann die Operationen und schließlich private Klassendefinitionen, der jeweiligen Klasse.

5.2 Modell der Klassifikation

Das folgende Teilmodell bildet die Klassifikation ab. Das wichtigste Detail in dieser Abbildung ist die Verwandtschaftsintensität und die Verknüpfung zu einer Mustergruppe als ein Element der Klassifikation und nicht direkt im Informationsflussmuster modelliert wurde.

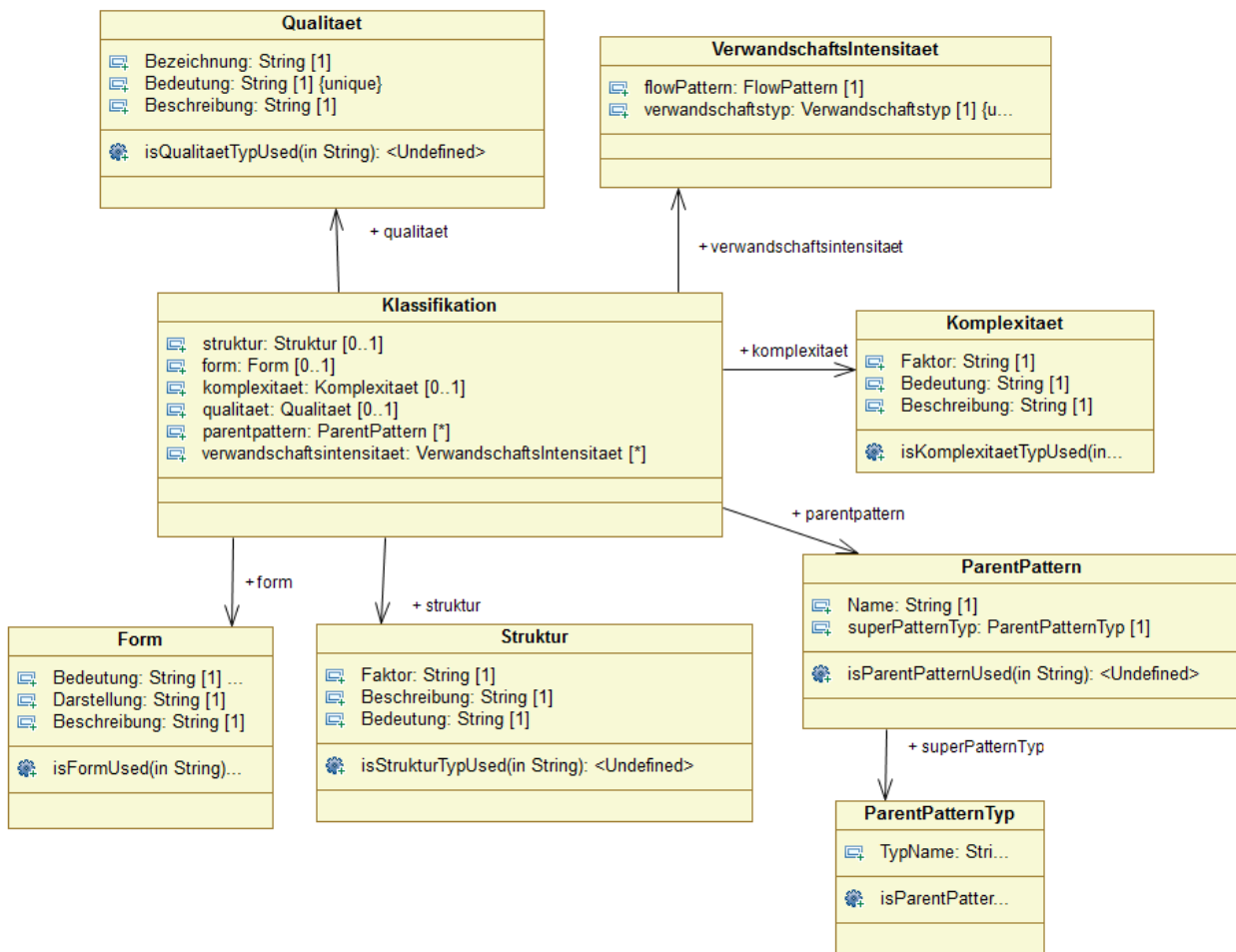


Abb. 8: Das Klassendiagramm der Klassifikation

5.3 Modell der Ansichten für normale Benutzer

An dem folgendem Modell kann man ähnlich wie bei einer „Sitemap“ alle Pfade zwischen den einzelnen Ansichten nachverfolgen, d.h. man kann sehen von wie man sich durch diesen Teilbereich der Anwendung navigieren kann.

Zu jeder modellierten Ansicht (mit dem Stereotyp „screen“), wird ein eigener Controller, mit den in der Ansicht definierten Attributen, generiert. Ziel dieses Modells ist es, jede „View“, bedingt durch die dahinter stehende Funktionalität, die man für das Arbeiten mit dem Informationsflussmuskatalogs benötigt, abzubilden.

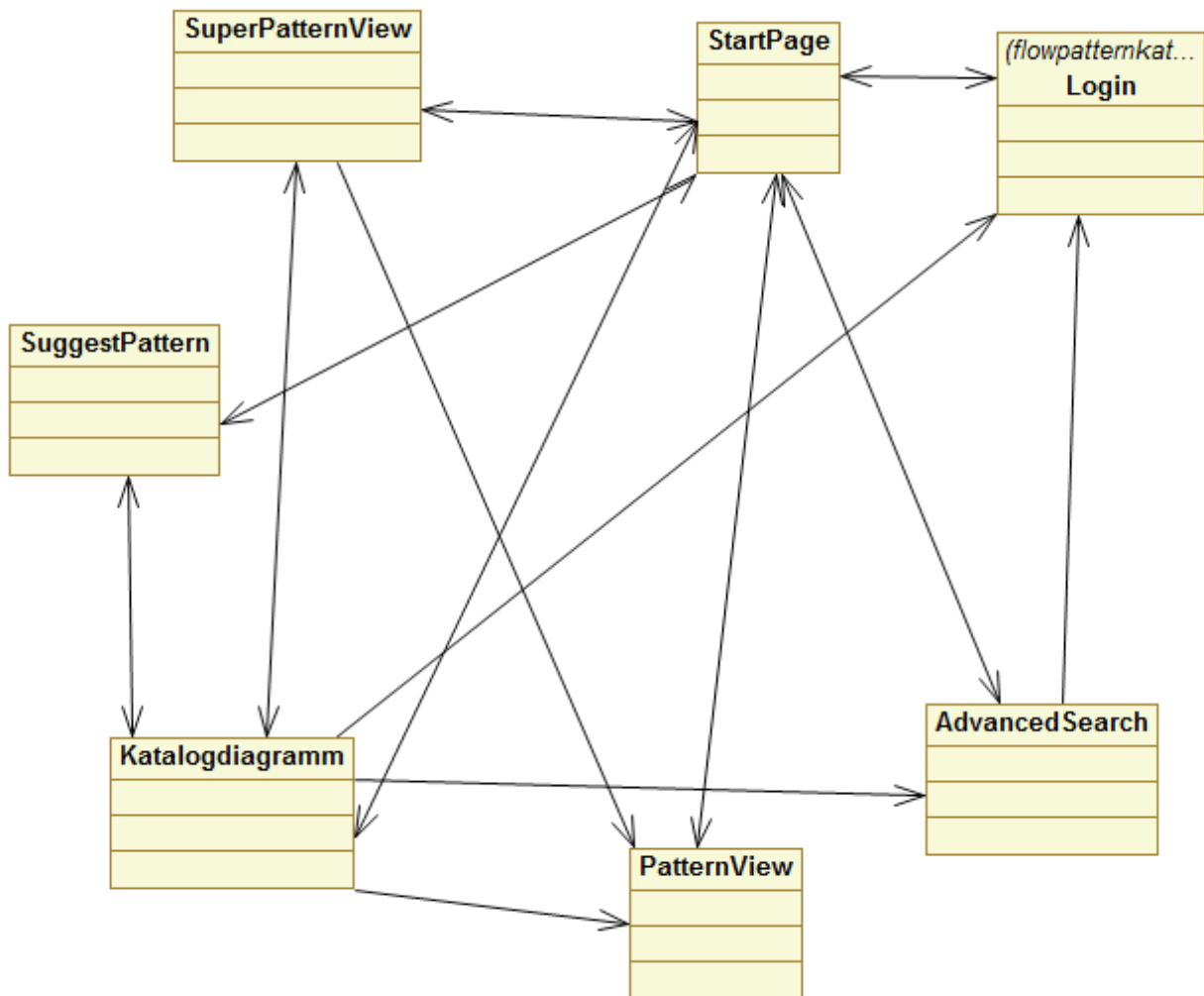


Abb. 9: Klassendiagramm der Ansicht für normale Benutzer

5.4 Modell der Ansichten in der Administration

Den zweiten Teilbereich der Anwendung stellt die Administration dar. Diese wurde separat zu der Hauptansicht modelliert. Beide Bereiche sind über die Ansicht „Login“ miteinander verbunden. Die Administration teilt sich entsprechend den entworfenen Rollen in drei Bereiche auf. Dabei sind die Account-, Muster- und Strukturadministration über die Startseite der Administration verbunden. Je nach Rolle kann der Benutzer die Bereiche auf der Startseite der Administration weder sehen noch betreten. Jedoch kann jeder Benutzer seinen eigenen Account in der Account-Administration, bis auf die Rolle, ändern, es sei denn er ist Superadministrator.

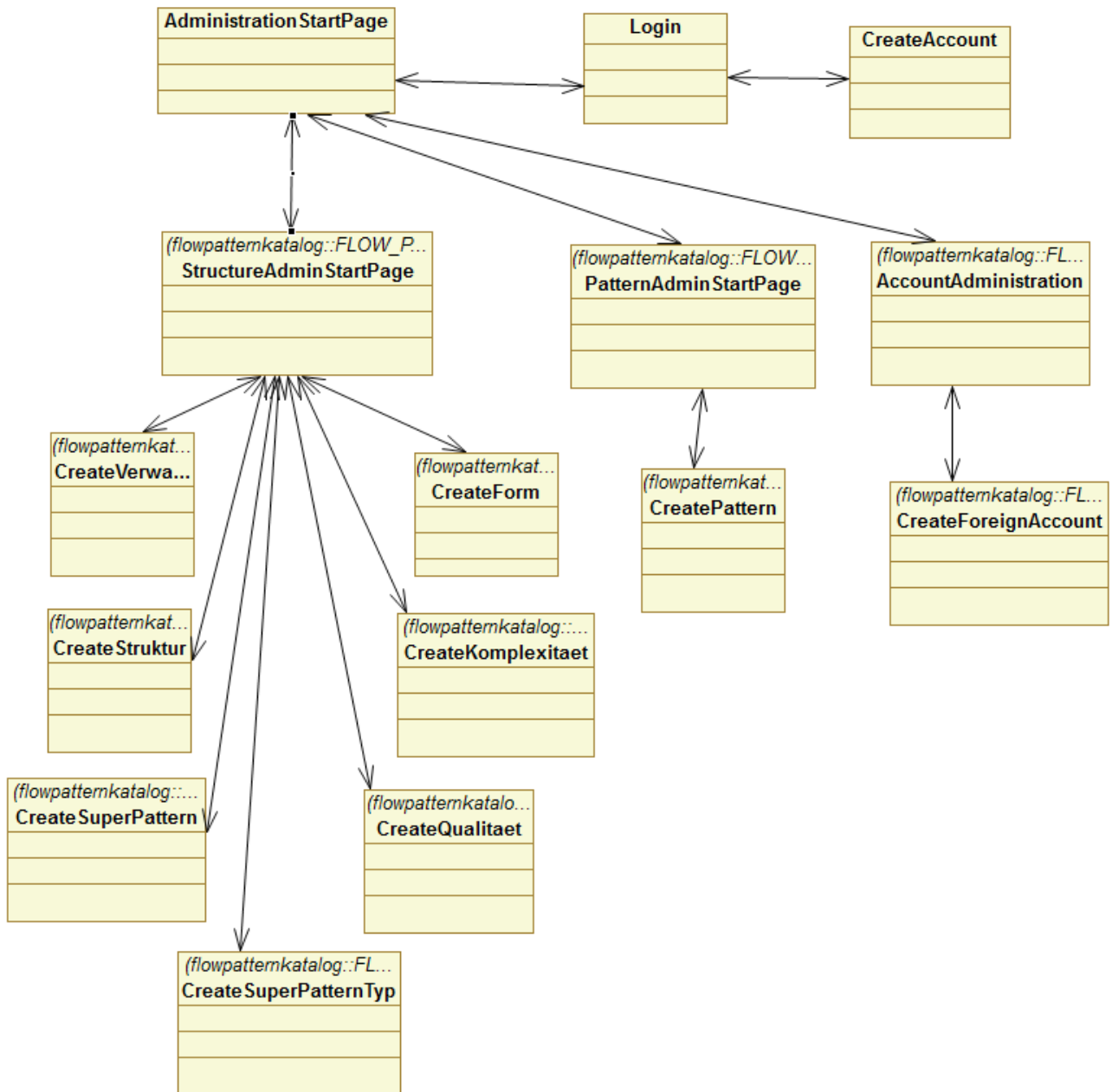


Abb. 10: Modell der Administrationsoberfläche

Das obige Modell ist jedoch aus Platzgründen nicht ganz vollständig, denn zu jeder „Create...“-Ansicht, innerhalb der Strukturadministration, existiert eine entsprechende „Edit...“-Ansicht, die aber die gleichen Verknüpfungen hat.

5.5 DAO-Pattern

Um zu verstehen wie die generierten Controller, die mit den dazugehörigen Ansichten und der Datenbank kommunizieren muss man das Data-Access-Object-Pattern(DAO) verstehen, denn Acceleo generiert auch diese Programmstruktur schon vor.

Zunächst muss man für sämtliche CRUD-Operationen, die man mit den dazugehörigen Objekt machen möchte, eine sogenannte DAO-Klasse benutzen. Diese Klasse heißt genauso wie das dazugehörige Objekt nur mit einem „DAO“ hinten angehängt.

Der Vorteil dieses Konstruktes ist, dass wenn man mal die Datenbank ändert, von z.B. bisherigen MySql zu PostgreSQL, so muss man alle CRUD-Operationen nur in den DAO-Klassen ändern.

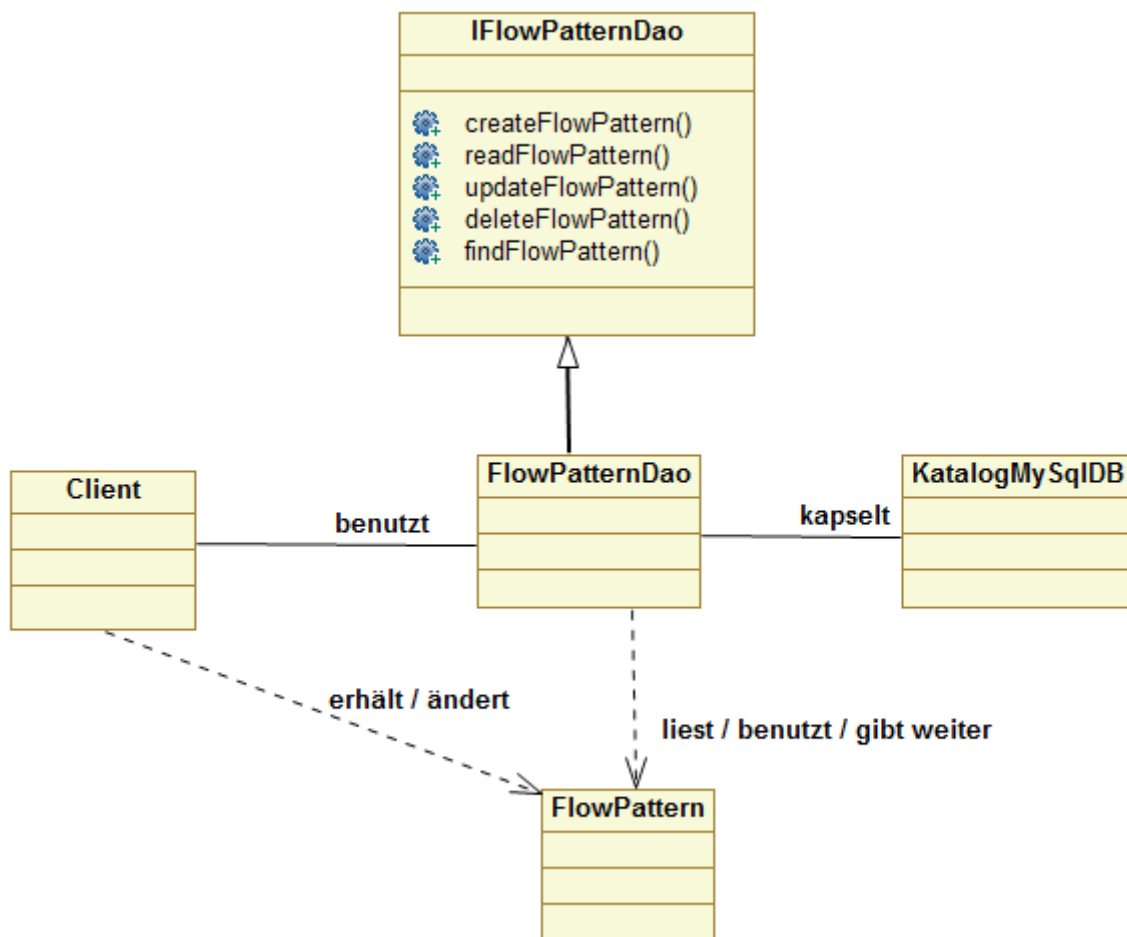


Abb. 11: Klassendiagramm des verwendeten Dao-Patterns am Beispiel „FlowPattern“

5.6 Factory-Proxy-Pattern

Zusätzlich zu dem obigen DAO-Pattern werden zu jeder Entität des Modells ein Factory-Pattern angelegt. Das Factory-Pattern hat den Vorteil, dass wenn sich mal eine Entität, die die Factory erzeugt, ändert und jetzt anders instanziiert wird, muss diese Änderungen nur in der erzeugenden Methode der Factory geändert werden, d.h. das die Factory-Methode den Aufrufer von der Erzeugung des tatsächlichen Objektes entkoppelt.

Tatsächlich arbeitet man aber nicht auf einem Objekt der Entitäten-Klasse sondern auf einem Proxy-Objekt, das die Entitäten-Klasse erweitert, d.h. die Fabrik erzeugt kein Objekt der Entität sondern ein Objekt der entsprechenden Proxy-Klasse. Das wird deshalb gemacht um die Realisierung der Relationen zwischen den einzelnen Entitäten von den Entitäten selber zu lösen und in die jeweilige Proxy-Klasse zu verlagern.

Wenn sich nun also eine Beziehung ändert, z.B. ein neues Attribute also eine neue Relation zu einer Entität zu der Musterschablone hinzukommt, muss man diese Änderungen nur in der FlowPatternProxy-Klasse durchführen und nicht in allen Klassen, die das FlowPattern-Objekt benutzen.

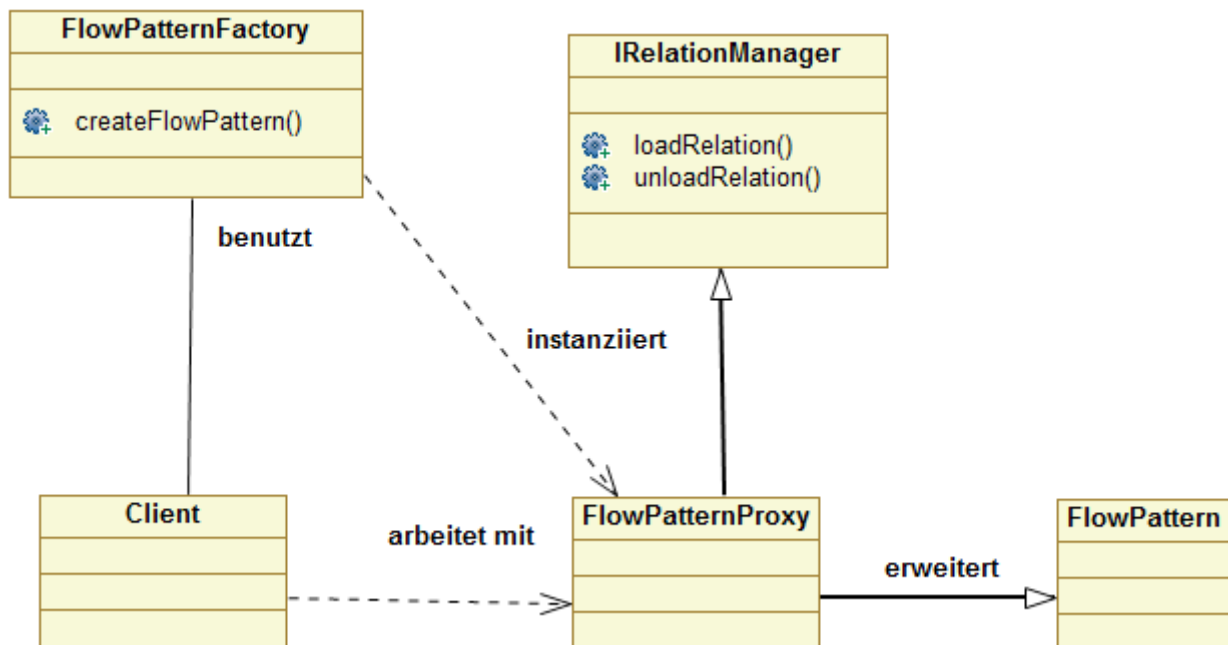


Abb. 12: Klassendiagramm der beiden kombinierten Factory- und Proxy-Pattern am Beispiel „FlowPattern“

5.7 Rich-Text-Editor

Um die Eingabe neuer Muster möglichst benutzerfreundlich zu gestalten, d.h. vor allem mit dem Ziel eine möglichst effiziente Bedienbarkeit zu gewährleisten, wurde ein Rich-Text-Editor, namentlich „FckEditor“, für die Eingaben ausgewählt.

Dieser Editor ist selber in Javascript geschrieben und wird nur im Administrationsbereich eingesetzt. Er bietet zudem eine PHP-Schnittstelle an, die auch von Smarty aus direkt integriert werden kann.

Sämtliche Eingaben werden, nach dem WYSIWYG-Prinzip, genauso wie man es im Editor sieht in HTML und CSS übersetzt. Die so entstandenen Eingabedaten werden in der Datenbank abgelegt.

Der Vorteil dieses Editors ist außerdem, dass man die Präsentation der Ausgabe bereits bei der Eingabe der Daten gestalten kann. Das spart Entwicklungszeit, die man sonst für das separate CSS bzw. eine Eigenentwicklung eines neuen Editors aufwenden müsste.

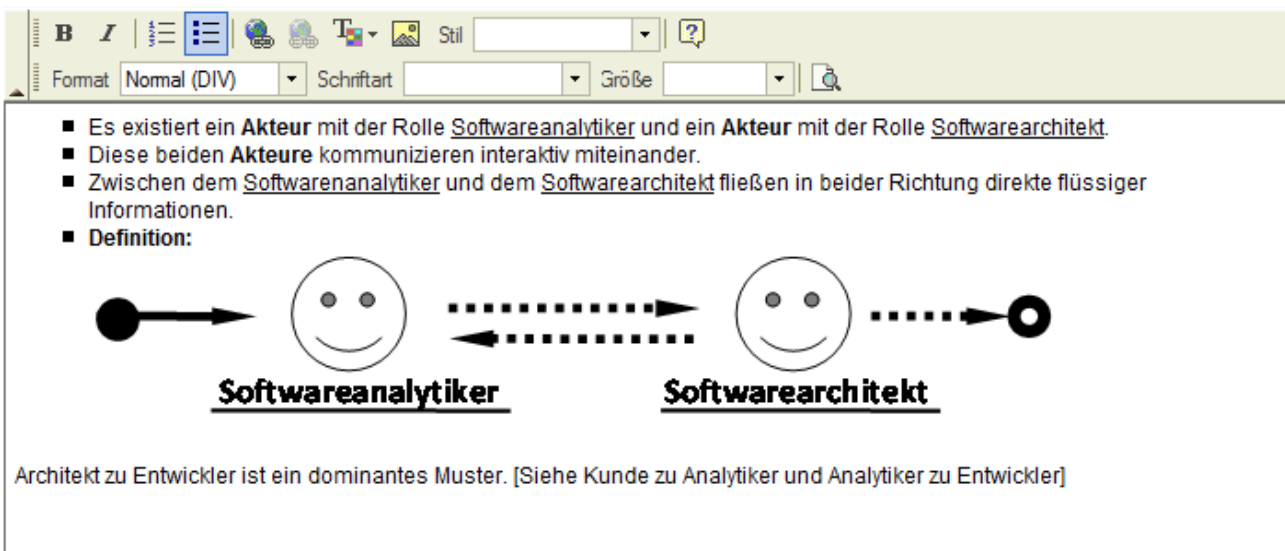


Abb. 13: Screenshot des Editors

5.8 Farbwähler

Mit dem Ziel, die Eingabe von neuen Daten bzw. das Ändern von bestehenden möglichst bequem und effizient zu halten, wurde entschieden ein zusätzliches Werkzeug für die Eingabemaske der Arten der Verwandtschaftsintensität zu verwenden. Es handelt sich dabei um einen „Farbwähler“, der, wie der Name schon erahnen lässt, das Ändern der Farbcodierung erleichtern soll.

Dieses Werkzeug ist wie der Rich-Text-Editor in Javascript geschrieben und kommt nur in der Administration zum Einsatz.

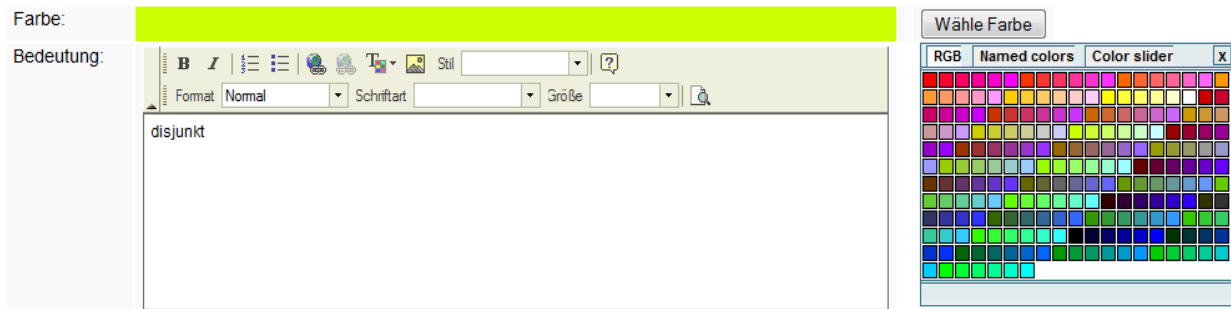


Abb. 14: Screenshot des Farbwählers

6. Usabilitytest

6.1 Ziel

Nachdem der Katalog in der ersten Version fertig gestellt wurde, gilt es zu überprüfen ob man alle an den Online Katalog gestellten Anforderungen erfüllen kann. Deshalb wurde ein Usabilitytest mit freiwilligen Testpersonen initiiert.

Aus der Definition für Usability [siehe Kapitel 3.3.1], wird auch deutlich welche Anforderungen an den Usabilitytest gestellt werden. Der Test muss zeigen wie gut sich der Online-Katalog im Vergleich zu dem Papier-Katalog in Bezug auf „Bedienbarkeit“, hinsichtlich der „Effizienz“, und „Konformität“ verhält.

Ziel ist dabei festzustellen ob der Online-Katalog tatsächlich auch eine Verbesserung zu den Nachteilen des Papier-Katalogs darstellt. Wenn hier vom „Papier-Katalog“ die Rede ist, ist damit die einfache digitalisierte Form gemeint. Um beide Katalog-Formen vergleichen zu können werden den Testern mehrere Aufgaben, die alle wesentlich Anforderungen an die Usability (Konformität) überprüfen, gestellt.

Die Tester führen die selben Aufgaben in beiden Katalogen durch und es wird dabei die Zeit gemessen, die sie für das Erledigen der Arbeit benötigen. Schließlich wird dann ein Vergleich hinsichtlich der Bedienbarkeit gemessen an der Effizienz angestellt, da die Bedienbarkeit eine Hauptanforderungen an den Online-Katalog darstellt.

6.2 Aufgabenzusammenstellung

Da die Anwendung in zwei Bereiche aufgeteilt worden ist, der Teilbereich für normale Benutzer und der Teilbereich für die Administratoren, bietet es sich an, auch die Aufgaben entsprechend aufzuteilen. Deshalb wurden je Teilbereich drei Aufgaben gestellt.

Die Aufgaben wurden so angeordnet, dass der Tester zu erst die ersten drei Aufgaben für den Teilbereich der normalen Nutzern und dann die drei Aufgaben für die Administratoren erledigen muss.

Da aber eine Anforderung zusätzlich besagt, dass der Katalog gut navigierbar sein soll, wurde zum Schluss noch eine 4. Aufgabe zu dem Teilbereich für normale Nutzer gestellt, damit der Tester mindestens zweimal zwischen den Bereichen navigieren muss, vorausgesetzt der Tester geht die Aufgaben wie beschrieben durch.

Der Aufwand für die Tester wurde zudem reduziert in dem man ihnen beide Kataloge für den Test mit nur je 8 Informationsflussmustern darin bereitstellte.

Folgenden Aufgaben wurden an die Benutzer gestellt:

- **Normale Benutzer**

1. Finden Sie das Muster „**Buerokratie**“ und die Form seiner Komplexitätsklassifikation
 - *Ziel dieser Aufgabe ist es den Tester dazu zu bringen sich bis zur Ansicht „PatternView“, also zur Ansicht eines konkreten Musters, durch zu navigieren und festzustellen wie schnell sich der Tester einen Überblick verschaffen kann indem er speziell die Komplexität des Musters suchen muss.*
2. Finden Sie alle Muster, der Mustergruppe „**Mainly Writing**“
 - *Der Tester soll sich nun zu der Suchmaske navigieren und entsprechend der Aufgabe einen Suchvorgang durchführen. Es wurde mit Absicht kein Hinweis auf die Suchmaske gestellt, um zu messen wie intuitiv sich der Online-Katalog bedienen lässt.*
3. Finden Sie ein zu „Buerokratie“ „**disjunktes**“ Muster
 - *An dieser Stelle sollte der Benutzer wieder seinen Weg zur „PatternView“ suchen. Die Idee dahinter ist, wieder die Navigation zu prüfen.*
4. Finden Sie alle Muster mit der Formklassifikation „**quantitativ**“
 - *Diese Aufgabe wurde als letzte, noch hinter den Aufgaben für die Administratoren, gestellt. Der Grund dafür ist, dass der Tester gezwungen wird zweimal zwischen den Teilbereichen zu wechseln. Diese Aufgabe ist nur über die Suchmaske lösbar. Außerdem soll geprüft werden ob die Suche tatsächlich im Vergleich zu dem Katalog ins Papierform verbessert wurde, denn anders als bei der ersten Suchaufgaben existiert für diese keine tabellarische Übersicht im Papier-Katalog.*

- **Administratoren**

1. Ändern Sie die Klassifikation der Komplexität von „**Buerokratie**“ in eine andere Form
 - *Diese erste Aufgabe im Administrationsbereich soll den Tester dazu bringen sich zu der Musteradministration durch zu navigieren. Es wird auch geprüft wie gut sich der Benutzer in der Oberfläche zurecht findet, denn die Ansicht für die Klassifikation befindet sich nicht auf der ersten Seite, die man sieht wenn man ein Muster ändert bzw. neu anlegt.*
2. Ändern Sie die Breite des Titelbildes von „**Buerokratie**“
 - *An dieser Stelle soll der Benutzer wieder zur ersten Seite der Eingabemaske wechseln und nun auch eine Eingabe in dem Rich-Text-Editor tätigen. Es soll hier auch die Bedienbarkeit des Rich-Text-Editors getestet werden.*
3. Ändern Sie die Farbe der Verwandtschaftsintensität „**disjunkt**“
 - *Ziel dieser Aufgabe ist es den Tester sich zur Strukturadministration zu navigieren und ein Element zu ändern. Vor allem soll aber der Farbwähler in dieser Aufgabe auf seine Bedienbarkeit hin ebenfalls getestet werden.*

Es sei darauf hingewiesen, dass die Benutzeradministration von keiner der Aufgaben abgedeckt wird. Der Grund dafür ist zuerst, dass dieser Bereich ein Novum im Vergleich zu dem Papier-Katalog darstellt, d.h. er lässt sich nicht vergleichen weil es nichts Vergleichbares gibt. Außerdem ging aus den Gesprächen mit dem Kunden hervor, dass eh nur sehr wenige Personen überhaupt Zugang , also den Status „Superadministrator“ erlangen, zu diesem Bereich haben werden.

Außerdem gibt es keine Aufgabe bei der der Benutzer lange Tippen muss, um zu vermeiden, dass die Ergebnisse sich aufgrund von unterschiedlichen Tippfähigkeiten verfälscht. Deshalb wird keine Aufgabe hinsichtlich der Erweiterbarkeit gestellt.

Welchen Anteil der Konformität, also den an die Oberfläche gestellten Anforderungen, wurden mit den Aufgaben konkret getestet ?

- **Aufgabe 1. :**
Der Tester ist in dieser Aufgabe dazu gezwungen sich von der Startseite aus zu einem speziellen Muster zu navigieren, d.h. hier wird die Anforderung an die Navigation und intuitive Bedienbarkeit geprüft.
- **Aufgabe 2. :**
Bei dieser Aufgabe geht es vor allem um die Anforderungen an die Suche.
- **Aufgabe 3. :**
Um diese Aufgabe zu lösen muss man sich erst wieder durch den Online-Katalog zurück zu dem Muster navigieren, d.h. es wird wieder die Navigation als auch intuitive Bedienbarkeit getestet

- **Aufgabe 4. :**
Erneut muss sich der Benutzer durch die Anwendung navigieren und im Anschluss eine Änderung in der Klassifikation des Musters durchführen, d.h. es wird neben Navigation vor allem geprüft wie effizient man ein Muster ändern kann.
- **Aufgabe 5. :**
Diese Aufgabe prüft vor allem die intuitive Bedienbarkeit eines Katalogs.
- **Aufgabe 6. :**
An dieser Stelle wird die Bedienbarkeit des Farbwählers
- **Aufgabe 7. :**
Diese Aufgabe ist die aufwändigste Suchaufgabe unter allen gestellten. Zudem wirkt sich die eingeschränkte Möglichkeit Muster in dem Papier-Katalog zu sortieren stark auf die Effizienz aus.

Hier eine Auflistung der Anforderungen an die Oberfläche und ein Vergleich ob sie überprüft wurden:

Anforderung an	Überprüft in
Navigation	A1, A3, A4
Suche	A2, A7
Intuitive Bedienbarkeit	A1, A3, A4, A5, A6, A7
Sortierung	A7
Weitergabe	Getestet, da alle Tester auf dem selben Katalog gearbeitet haben
Änderbarkeit	A4, A5, A6
Erweiterbarkeit	Nicht getestet

6.3 Risiko und Gegenmaßnahmen

- **Problem:** Um möglichst viele Ergebnisse zu erhalten wurde, neben persönlich angesprochenen Testpersonen, ein Aufruf im Forum der Informatikfachschaft gestartet. Da jedoch nun viele Tests anonym durchgeführt werden, entstehen einige Risiken hinsichtlich der Richtigkeit der Ergebnisse.
- **Gegenmaßnahme:** Um sicherzustellen, dass die Ergebnisse, die von den Testern zurückgeliefert auch tatsächlich mit der Erledigung der Aufgabe einhergehen, wird hinter jeder Aufgabe ein Lösungsfeld hinzugefügt. Jeder Tester muss also nicht nur seine Zeit für die Aufgabe sondern auch die Lösung mit eintragen. Die Konsequenz daraus ist, dass man Ergebnisse ohne der Angabe einer Lösung nicht in der Auswertung berücksichtigen kann.

- **Problem:** Eine andere Gefahr stellt die Reihenfolge der vorformatierten Zeitfelder pro Aufgabe dar. Geht man davon aus, dass der Tester vorher weder den Online- noch den Papier-Katalog gesehen hat, dann wird er mit Sicherheit bei jeder Aufgabe erstmal nach Zusammenhängen suchen.

Hat er den Hintergrund der Aufgabe verstanden, als er diese zuerst mit dem Online-Katalog gemacht hat, dann ist es sehr wahrscheinlich, dass er im Papier-Katalog wesentlich weniger Zeit benötigt, da er schon weiß wonach er suchen muss, d.h. die Antwort schon kennt.

- **Gegenmaßnahme:** Dieser Gefahr lässt sich zunächst so begegnen, dass man pro Teilbereich der Anwendung nochmals je drei ähnliche Fragen stellt, die dann abwechselnd im Online- oder dem Papier-Katalog gemacht werden müssen. Das Problem dabei ist jedoch, dass die Vergleichbarkeit darunter leidet.

Eine andere Möglichkeit ist, dass man die Felder für die Zeitangaben unterschiedlich anordnet, d.h. z.B. dass der Tester bei der ersten Aufgabe zuerst im Online-Katalog, bei der zweiten zuerst im Papier-Katalog, usw. eine Antwort eintragen muss. Das Problem besteht damit aber immer noch, nur dass die „Eingewöhnungsphase“ jeweils sich von einer Seite zur anderen pendelt.

Das gewählte Lösungskonzept sieht schließlich so aus, dass man 4 unterschiedliche Versionen des Aufgabenstellungsblattes anlegt.

1. Bei jeder Aufgabe ist der Online-Katalog als erstes zu bearbeiten
2. Bei jeder Aufgabe ist der Papier-Katalog als erstes zu bearbeiten
3. Bei jeder Aufgabe wechselt die Katalog-Form, beginnend mit dem Online
4. Bei jeder Aufgabe wechselt die Katalog-Form, beginnend mit dem Papier

Damit man mit der Gegenmaßnahme Erfolg hat, braucht man aber mindestens einen Tester je Aufgabenblatt, dafür bleiben die Ergebnisse noch gut vergleichbar.

6.4 Ergebnis

Eine ausführliche Darstellung mit allen erhaltenen Ergebnissen finden Sie im Anhang.

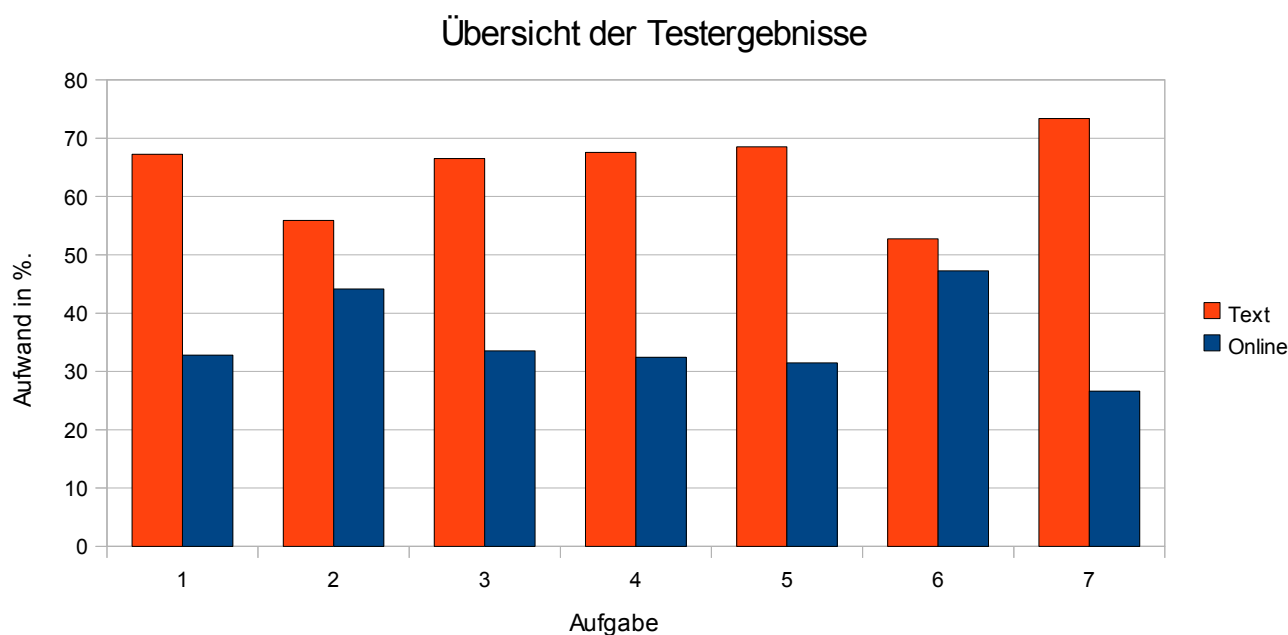


Abb. 15: Übersicht aller erhaltenen Testergebnisse

6.5 Auswertung

In der obigen Darstellung der Ergebnisse sehen Sie zunächst zwei Balken. Der blaue Balken repräsentiert alle Ergebnisse des Online-Katalogs und der orange die Ergebnisse des Papier-Katalogs.

Auf der X-Achse findet man die im einzelnen durchgeführten Aufgaben und auf der Y-Achse den dafür benötigten Aufwand in Relation zu der insgesamt für die jeweilige Aufgabe benötigten Zeit. Beide Balken zeigen einen über alle Ergebnisse berechneten Durchschnittswert an, das bedeutet das z.B. in Ausgabe 1. die Tester durchschnittlich alle knapp 70% der Arbeitszeit, die sie für Aufgabe 1. insgesamt verbrauchten, in den Papier-Katalog und gut 30% in den Online-Katalog aufgewandt haben.

Man kann auf den ersten Blick sehen, das der Online-Katalog in jeder Aufgabe weniger Zeit beansprucht hat als der Papier-Katalog. Jedoch gibt es eine deutliche Annäherung in Aufgabe 2. und Aufgabe 6. Das liegt zunächst daran, dass es zu Aufgabe 2. im Papier-Katalog eine vorbereitete Tabelle, die das Ergebnis direkt liefert, d.h. die Suche ist beendet wenn man die Tabelle gefunden hat. Dementsprechend sieht der Aufwand aus. Vergleicht man aber die sehr ähnliche Aufgabe 7. mit Aufgabe 2., sieht man wie der Aufwand aussieht wenn mal eine solche Tabelle fehlt. Um also den Papier-Katalog ähnlich effizient wie den Online-Katalog zu gestalten müsste man für alle denkbaren Suchkombinationen eine eigene Tabelle anlegen.

Die Annäherung bei Aufgabe 6. lässt sich so erklären, dass die Aufgabe als solche in beiden Katalogen keinen großen Aufwand erfordert, denn im Papier-Katalog muss man dies wie im Online-Katalog nur an einer einzigen Stelle ändern. Da aber an dieser Stelle vor allem die Bedienbarkeit des Farbwählers getestet werden sollte und der Aufwand trotzdem im Online-Katalog geringer war, kann das Ergebnis als kleiner Erfolg für den Online-Katalog angerechnet werden.

Der Balken zum Papier-Katalog in Aufgabe 5. spiegelt jedoch nicht genau die Realität wieder, denn es wurden auch zwei Ergebnisse eingereicht, die die geforderte Funktionalität in ihrem Textverarbeitungsprogramm überhaupt nicht finden konnten. Daher wäre der Vergleich mit den Ergebnissen des Online-Katalogs an dieser Stelle hinfällig.

Wie bewertet man dieses Problem?

Man kann davon ausgehen, dass die jeweilige Testperson deutlich mehr Zeit damit verbracht hat nach der Funktionalität zu suchen, als sie es für die Erledigung im Online-Katalog benötigte. Die Zeit wäre im schlimmsten Fall unendlich, wenn diese Funktion überhaupt nicht existiert und der Tester an dieser Stelle nicht aufgeben würde. Allerdings lässt sich ein unendlicher Wert nicht abbilden. Um diesem Problem zu begegnen habe ich mich entschlossen an den entsprechenden Stellen eine Näherungen einzusetzen.

Es wurde daher ein Faktor ermittelt, der die durchschnittliche Abweichung zwischen den Ergebnissen des Online- und Papier-Katalog in dieser Aufgabe darstellt. Dieser Faktor liegt bei einem Wert von 2,18, d.h. durchschnittlich braucht ein Tester 2,18 Mal länger für eine Aufgabe im Papier- als er es in dem Online-Katalog braucht. Das fehlende Ergebnisse im Papier-Katalog wird dann so ergänzt, dass man den Wert des Online-Katalogs für die jeweilige Aufgabe nimmt und mit dem Faktor multipliziert.

Außerdem liegt ein Ergebnis vor bei dem ein Tester zur Aufgabe 7. weder im Online- noch im Papier-Katalog eine Antwort finden konnte. Deshalb wurde diese Aufgabe an der entsprechenden Stelle mit jeweils 0 bewertet.

Im Detail sieht man, dass für Suchaufgaben, bzw. für die Navigation, durchschnittlich 62,24% der aufgewandten Suchzeit im Papier-Katalog verbracht hat. Ähnlich sieht es in der Administration aus, die im Papier-Katalog durchschnittlich benötigte Zeit liegt bei 63,15% von der insgesamt für die Administration aufgewandten.

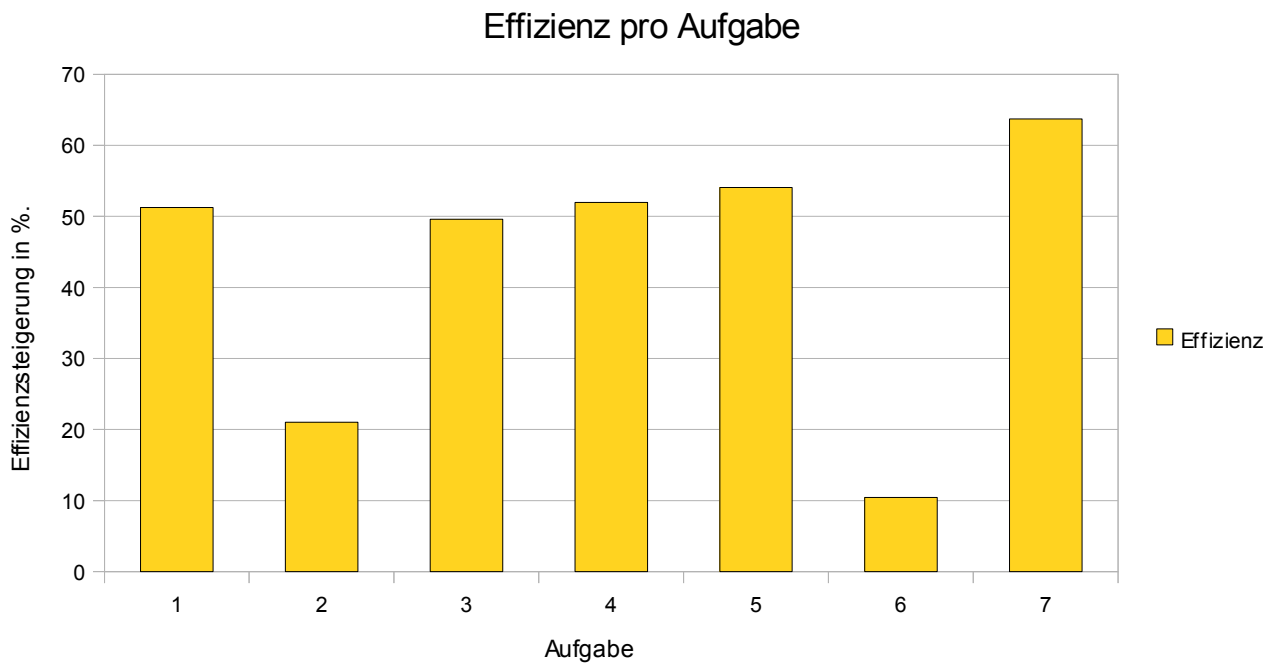


Abb. 16: Übersicht der Effizienz

Was wurde also hier gezeigt?

Ziel war es mit Hilfe der Effizienz eine Aussage über Bedienbarkeit, die Wahl der Aufgaben einen Eindruck zur Konformität und damit insgesamt über die Benutzbarkeit (Usability) machen zu können.

An den vorliegenden Ergebnissen kann man bereits ablesen, dass der Online-Katalog im Bereich der „normalen“ Benutzer sich 39,33% und im Bereich der Administratoren 41,65% effizienter bedienen lässt.

Es sei jedoch noch drauf hingewiesen, dass aufgrund relativ weniger Tester, dieses Ergebnis nicht als exakt zu nehmen ist, sondern eher nur eine Tendenz erkennen lässt, wo der Online-Katalog im Vergleich zum Papier-Katalog steht.

7. Zusammenfassung und Fazit

Diese Arbeit greift die Theorie der Informationsflussmuster, die Xiaoxian Ge in ihrer Masterarbeit erarbeitete, und den dort entstandenen Katalog für Informationsflussmustern auf. Aus den erhobenen Anforderungen hat sich herausgestellt, dass die Usability vor allem hinsichtlich der Bedienbarkeit eine der zentralen Anforderungen darstellt.

Bedingt durch die technischen Anforderungen wurde eine Client-Server-Systemarchitektur festgelegt. Aufgrund hoher Komplexität wurde eine 3-Schichten-Softwarearchitektur, bestehend aus Daten, Logik und Präsentationsschicht, konstruiert.

Mit dem Ziel die Entwicklung zu beschleunigen und möglichst flexibel hinsichtlich struktureller Erweiterungen bzw. Änderungen zu gestalten, wurde eine modellgetriebene Entwicklung angestrebt. Im Speziellen wurde sich für den MDA Ansatz von Acceleo entschieden, da Acceleo genau die vorher festgelegte Softwarearchitektur und Technologie aus einem Modell generiert.

Schließlich wurde die Anwendung entsprechend den Anforderungen des Kunden modelliert und implementiert. Eines der größten angestrebten Ziele hinter dieser Arbeit war es die Nachteile des Papier-Kataloges auszugleichen.

Um zu prüfen und damit auch nachzuweisen, dass der neue Katalog tatsächlich den an ihn gestellten Anforderungen, vor allem die Bedienbarkeit betreffend, entspricht, wurde ein Usabilitytest mit freiwilligen Testpersonen durchgeführt.

Aus den Testergebnissen ging hervor, das der Online-Katalog gegenüber dem Papier-Katalog rund 40% effizienter bedienen lässt und damit die Hauptanforderung zu erfüllen scheint.

7.1 Ausblick

Obwohl alle wesentlichen Funktionen bereits fertiggestellt sind, so gibt es noch einige Features, mit denen man den Katalog erweitern könnte:

- Zusätzliche Javascript Komponenten in den Oberflächen für „normale“ Benutzer.
Man könnte die Suchmaske um zusätzlich eine dynamische Sortierung der Suchergebnisse erweitern.
- Katalogdiagramm nicht tabellarisch sondern mit einem Graphen abbilden.
Um die Suche nach verwandten Mustern noch effizienter zu gestalten, wäre es möglich die Tabelle durch einen Graphen zu ersetzen. Zudem könnte man die Abbildung des Graphen mit einer Suche verknüpfen, sodass man sich z.B. die Vierwandschaften aller „dominanten“ Informationsflussmuster angucken könnte. Es würde sich auch anbieten nicht nur die Zugehörigkeit zu einer Mustergruppe sondern die angegebenen Verwandtschaftsintensitäten graphisch abzubilden.

- **Community-Funktionen**
Bis auf die Möglichkeit neue Muster vorzuschlagen und direktem Mail-Kontakt zu einem Muster-Autor gibt es kaum ein Chance mit anderen an FLOW interessieren Personen oder gar FLOW-Experten in Kontakt zu kommen. Es würde sich auf lange Sicht vermutlich lohnen, auch ein Forum oder eine andere Möglichkeit über Muster zu diskutieren mit im Online-Katalog zu integrieren. Zudem hätte man dann die Möglichkeit die angemeldeten Personen direkter zu ihren Erfahrungen mit FLOW und Informationsflussmustern befragen.

Literaturverzeichnis

[XG 2008] Masterarbeit „FLOW Patterns: Beschreibung und Diskussion von Informationsflussmustern in der Softwareentwicklung“, Xiaoxian Ge, 2008
[Informationsflussmustern_in_der_Softwareentwicklung](#)

[OMG] [Model Driven Architecture](#), Paper von der Object Management Group, 12.06.2003
[OMG_Paper_zu_MDA](#)

[ACC] Acceleo Dokumentation, Firma Obeo, 09.06.2009
[Acceleo_PHP_Generator_Dokumentation](#)

[J2EE] [Data-Access-Object-Pattern](#), J2EE Handbuch von Sun, 09.07.2009
[J2EE_Handbuch](#)

[Wiki 1] Tagcloud(Schlagwortwolke), Artikel in Wikipedia, 09.01.2009
[Schlagwortwolke](#)

Anhang

- Tabelle mit allen Testergebnissen
- Handbuch zur Installation und Konfiguration des Online-Katalogs
- Vollständiger Code und Modell des Online Katalogs
- Daten DVD mit folgenden Daten, beginnend ab dem Wurzelverzeichnis des DVD-Laufwerks:
 - Bachelorarbeit/Bachelorarbeit.odt
Diese Datei ist Abschlussarbeit im OpenDocument-Text-Format
 - Bachelorarbeit/Bachelorarbeit.pdf
Dies ist die Abschlussarbeit im PDF-Format, so wie sie auch gedruckt wurde.
 - Bachelorarbeit/Anhang/flowpatternkatalog
Dies ist das vollständige Eclipse-Projekt des Online-Katalogs. Sämtlicher Code sowie das Modell finden Sie hier.
 - Bachelorarbeit/Anhang/Handbuch.odt
Dies ist das Handbuch mit allen notwendigen Schritte zur Installation und Konfiguration des Online-Katalogs.
 - Bachelorarbeit/Anhang/Handbuch.pdf
Das ist nochmal das Handbuch aber im PDF-Format.
 - Bachelorarbeit/Anhang/Usabilitytest/Testergebnisse.ods
Diese Datei enthält sämtliche eingegangenen Testergebnisse des Usabilitytests, sowie die Berechnungen für die Auswertung.
 - Bachelorarbeit/Anhang/Usabilitytest/Usability-Test-Aufgabenblatt.doc
Das sind die Usabilitytest-Bögen, wie sie die Tester erhielten.
 - Bachelorarbeit/Anhang/Usabilitytest/Usability-Test-Aufgabenblatt-Vers-2.doc
 - Bachelorarbeit/Anhang/Usabilitytest/Usability-Test-Aufgabenblatt-Vers-3.doc
 - Bachelorarbeit/Anhang/Usabilitytest/Usability-Test-Aufgabenblatt-Vers-4.doc
 - Bachelorarbeit/Anhang_Handbuch/pearinstaller.php5
Dies ist der WEBINSTALLER, den Sie für die Installation von PEAR einsetzen können.