Universität Duisburg-Essen
Institut für Informatik und Interaktive Systeme
Fachgebiet Wissensbasierte und natürlichsprachliche Systeme

Context-aware Web Engineering – ein adaptives, ontologiebasiertes Musikportal

Ausarbeitung des Vortrags im Seminar „Benutzermodellierung in natürlichsprachlichen Systemen"
im Sommersemester 2006 bei Prof. Dr. Wolfgang Hoeppner

von
Joel Fischer {Matrikelnr. 744440, E-mail: joel.fischer@fit.fraunhofer.de}
Daniel Westheide {Matrikelnr. 743954, E-mail: daniel.westheide@stud.uni-due.de}

4. September 2006

Inhaltsverzeichnis

1 Einleitung

Nutzer web-basierter interaktiver Systeme sehen sich einer steigenden Informationsflut ausgesetzt. Das Resultat kann eine kognitive Überlastung sein, der Nutzer muss ständig die für ihn relevanten Informationen herausfiltern. Oft führt die syntaktische Suche über weit verbreitete Internet-Suchmaschinen nur zu unzureichenden Ergebnissen. Die Unsicherheit bleibt, eventuell relevantere Dokumente nicht zu finden, da das explizite Suchwort nicht vorkommt. Durch die Entwicklung und Verbreitung interaktiver Web-Anwendungen in verschiedene gesellschaftliche Bereiche wie Arbeit und Freizeit haben sich auch die Anforderungen an die Systeme verändert. Nutzer interagieren mit web-basierten Systemen aus unterschiedlichen Kontexten heraus und Erwartungen über die Aufgaben, die das System erfüllt, können entgegenlaufend sein. Um dem Nutzer „the right thing at the right time in the right way" [6] zu präsentieren, wird ein kontextadaptives System notwendig.

Die vorliegende Arbeit dokumentiert ein Referat, das im Rahmen des Seminars „Benutzermodellierung in natürlichsprachlichen Systemen" im Sommersemester 2006 gehalten wurde. Das Referat basiert auf dem Studienprojekt „Context-aware Web-Engineering", welches im Masterstudiengang „Angewandte Kommunikations- und Medienwissenschaft" am Institut für Informatik und Interaktive Systeme im Fachgebiet „Interaktive Systeme und Interaktionsdesign" im Wintersemester 2005/2006 durchgeführt wurde. Grundlage der von den Studierenden durchgeführten Arbeit ist die Forschung des Instituts in den Bereichen der Kontextmodellierung, Kontext-Adaptivität und des ontologiebasierten Engineerings. Der vom Institut entwickelte Ansatz stellt Context-Engineering in einen systematischen Entwicklungsrahmen und konzeptualisiert diesen zum Beispiel in [5]. Der Ansatz wird außerdem im BMBF-geförderten Forschungsprojekt WISE verfolgt, an dem das Fraunhofer IAO, die Universität Stuttgart und die Universität Duisburg-Essen beteiligt sind.

Der Context-Engineering-Ansatz integriert Methoden für ontologiebasierte Kontextmodellierung in den systematischen Entwicklungsprozess von Web-Anwendungen. Ziel ist es hierbei, den Nutzer adaptiv bei der Suche nach relevanten Informationen und Diensten zu unterstützen. In Abhängigkeit des Interaktionskontextes des Nutzers soll die Relevanz von Informationen automatisch bewertet werden und Inhalte, Dienste und Navigationsstruktur adaptiv dargeboten werden. Modellierungsphasen innerhalb des Web-Engineering-Prozesses sind in konzeptionelles, Navigations-, Präsentations- und Sichtenmodell unterteilt. Das Herz der Architektur ist das konzeptionelle Modell. In diesem ist der Inhaltsbestand, die Domäne des Systems, explizit in Form einer Ontologie modelliert. Ergänzt wird dies durch die Kontextontologie und -relationen, die festlegen, welche Faktoren die Adaption von Navigation, Sicht (Platzierung und Auswahl des Inhalts) und Präsentation (grafische Darstellung) determinieren und in welchem Zusammenhang sie stehen.

Die studentische Forschungsgruppe exploriert, welche Adaptionsmaßnahmen (Inhaltsauswahl, Einbindung externer Dienste, Navigation und Präsentation) in einer dynamischen Webapplikation sinnvoll und umsetzbar sind. Ziel des Projekts ist es, anwendungsspezifische, sowie allgemein verwendbare Komponenten zur Inhalts- und Kontextmodellierung, Kontexterfassung sowie Adaption von Inhalt, externen Diensten, Navigation und Präsentation zu entwickeln.

Um den Forschungsgegenstand einzugrenzen, beschränken wir uns auf einen Anwendungsfall, dessen Entwicklungen aber aufgrund von Komponentenorientierung, angestrebter Wiederverwendbarkeit und Interoperabitlität nicht auf diesen Fall beschränkt bleiben soll. Bei der Auswahl des Anwendungsfalls gingen wir der Frage nach, welche Anwendung sinnvoll adaptiv gestalten werden kann, so dass sich ein Mehrwert für den Nutzer ergibt.

Als Musikliebhaber findet man sich oft vor seinem CD-Regal wieder und ist unschlüssig, welche CD zu seiner aktuellen Stimmung passt. Auch wünscht man sich oft mehr Auswahl, nur – zu seinem eigenen Musikgeschmack sollte die Musik schon passen. Aber wie sucht man nach Musik, von der man weder Interpret noch Titel kennt? Auch möchte sich ein Musikfreund gerne über Künstler informieren. Die Art der Produktpräferenz kann vielschichtig sein. So können Rezensionen von Alben oder Auftritten, Interviews mit Künstlern oder Beschreibungen ganzer Musikgenres im Bereich des Interesses des Nutzers liegen. Auch hier ergibt sich für den Informationssuchenden wieder das Problem, dass er Künstler, Musikstücke oder Alben, die nach seinem Geschmack sind, nicht kennt und somit eine gezielte Suche nach seinem „Musikgeschmack" nicht möglich ist. Dienste, die Produkte im Musikbereich anbieten, sind zahlreich. Konzertkarten, Bücher und Fanartikel können webbasiert bei vielen Anbietern bezogen werden. Was, wenn der Musikfreund an Konzerten von Künstlern, die aus seiner Region stammen, interessiert ist? Oder Musik nach seinem Geschmack live in der Nähe seines Wohnorts erleben möchte?

Die Erkenntnis, dass es ein adaptives Musikinformationsportal, welches alle relevanten Inhalte und Dienste bündelt, in dieser Form im Netz noch nicht gibt, eine blühende Gemeinde von Musikinteressierten hingegen schon, und dass die Domäne Musik zahlreiche, sinnvolle Möglichkeiten zur Adaption von Inhalten, Navigation und Präsentation enthält, hat uns motiviert, ein adaptives, webbasiertes Musikportal als Anwendungsfall zu verwenden. In diesem Projekt liegt der Schwerpunkt auf einer strukturierten Methode zur Modellierung von Domäne und Kontext, den Komponenten und Mechanismen der Kontexterfassung und den inferierten Adaptionsmaßnahmen. Die inhaltliche Ausgestaltung hat niedrigere Priorität, hierfür sollen externe Inhalte und Dienste eingebunden werden. Das Portal mit Inhalten zu füllen übersteigt die Kapazitäten dieses Projekts. Wir sehen einen wertvolleren Beitrag darin, bestehende, distribuierte Inhalte im Netz zu integrieren, anstatt neue zu schaffen.

2 Ansätze zur Adaption von Musikinhalten – ein Überblick

Zum gegenwärtigen Zeitpunkt existieren im World Wide Web bereits eine Reihe von Musikportalen, die adaptive Komponenten beinhalten, um dem Benutzer auf ihn zugeschnittene Inhalte anbieten zu können. Die dahinter stehenden Ansätze zur Benutzer- und Domänenmodellierung und die verwendeten Adaptionstechniken unterscheiden sich zum Teil grundlegend voneinander, aber auch von dem Ansatz, den wir in unserem Forschungsprojekt verfolgt haben. Um unsere eigene Herangehensweise einordnen zu können, sollen deshalb im Folgenden die gängigsten Konzepte anhand konkreter Beispiele von Musikportalen, in denen diese verwendet werden, beschrieben und miteinander ver-

gleichen werden. Dabei werden die Vor- und Nachteile der unterschiedlichen Konzepte aufgezeigt, um schließlich den von uns verwendeten Ansatz in diesem Feld einzuordnen und von den bestehenden Ansätzen abzugrenzen.

2.1 Expertenbasierter Ansatz

Will man dem Benutzer Musikinhalte jedweder Art anbieten, die genau seinen Geschmack treffen, so gibt es unterschiedliche Möglichkeiten, von dem Wissen über die vom Benutzer bisher bevorzugten Künstler oder Musikstücke auf neue Inhalte zu schließen. Eine dieser Möglichkeiten besteht darin, dem Benutzer solche Inhalte anzubieten, die im System auf irgendeine Weise als ähnlich zu den bisher vom Benutzer bevorzugten Inhalten spezifiziert sind. Wann aber sind sich zwei Künstler oder Musikstücke ähnlich?

Ein Ansatz, um die Ähnlichkeit von Musik bestimmen zu können, besteht darin, die Musik mit einer möglichst objektiven und detaillierten Beschreibung ebendieser zu versehen, basierend auf einer musikwissenschaftlichen Analyse durch Experten. Die Musikwissenschaftler analysieren, beschreiben und kategorisieren die Musik mit hohem Aufwand anhand definierter Eigenschaften. Es handelt sich also um eine inhaltsbasierte Vorgehensweise, bei der aber die Merkmale nicht automatisiert extrahiert werden, sondern aufgrund einer musikwissenschaftlichen Analyse von Experten modelliert werden, weshalb wir dieses Konzept als expertenbasierten Ansatz bezeichnen wollen.

Umgesetzt wird dieses Konzept etwa im sogenannten „Music Genome Project" (MGP), welches von der Firma Pandora Media, Inc. im Jahr 2000 ins Leben gerufen wurde. Angelehnt an das „Human Genome Project", welches die Gene des Menschen kartiert, analysierten über dreißig Musikwissenschaftler rund vierhundert eindeutige Schlüsselelemente von mehr als 300.000 Musikstücken von 10.000 Künstlern [11]. Die erstellte Taxonomie beinhaltet beispielsweise den Rhythmus, die Melodie, die verwendete Tonart, die Instrumentierung, die Stimmlage des Interpreten und auch den Inhalt und die Art der Texte. Da die vierhundert Merkmale bei jedem Lied untersucht werden müssen, dauert die Analyse eines einzelnen Musikstücks bis zu dreißig Minuten [11].

Abbildung 1: Pandora.com Web-Radio-Interface mit Begründung für gerade gespielten Titel [11]

Durch die Verwendung einer genau definierten Terminologie und fortwährender Qualitätskontrolle soll eine hohe Datenintegrität gewährleistet sein [10]. Vergleichsmöglichkeiten und Ähnlichkeiten sollen gezielt aufgespürt werden können. Dem Nutzer des Online-Radios „Pandora.com" oder dem Musikverkäufer beziehungsweise -käufer soll auf diese Weise Musik angeboten werden, die zu seinem bisherigen Musikgeschmack passt, die aber von einem Künstler stammen kann, den er zuvor noch nicht kannte. Der Vergleich der Gene der Musikstücke soll „[dem Nutzer] helfen, an die Musik anzuschließen, die [er mag]" (eigene Übers., [9]). Auf Wunsch bietet Pandora.com auch eine detaillierte Beschreibung der Ähnlichkeiten zum zuvor gespielten Lied (siehe Abbildung 1). Durch eine positive Bewertung durch den Hörer erhält das System die Rückmeldung, dass der gewählte Titel wirklich zum Musikgeschmack passt und kann seine Ergebnisse weiter verfeinern.

2.2 Fuzzy-Ansatz

Ob einem ein bestimmter Künstler oder ein Musikstück gefällt, hängt nicht unbedingt von objektiven Kriterien oder einer Einordnung in ein bestimmtes Genre ab, sondern oft auch davon, ob es zur gegenwärtigen Gemütslage desjenigen passt, der nach neuer Musik sucht. Auf dieser Idee baut ein weiterer Ansatz auf, der zwar nicht gänzlich auf objektive Kriterien verzichtet, diese aber in den Hintergrund rückt und auf eine eher vage Beurteilung dieser Merkmale vertraut. Umgesetzt wird dieses Konzept in der Webapplikation MusicLens [4], die vom Musik-Onlineshop FineTunes eingesetzt wird. Auch hier werden die Musikinhalte mit Metadaten angereichert, die Personen, die die Musik einordnen, besitzen jedoch nicht unbedingt eine musikwissenschaftliche Ausbildung. Neben den objektiven Kriterien *Volume*, *Tempo* und *Size* gibt es die Merkmale *Purpose*, *Sex*, *Age*, *Mood*, *Colour* und *Time*, durch die eher die durch die Musik vermittelten Gefühle zum Ausdruck gebracht werden. Die bewusst subjektiven und vagen Beurteilungsmöglichkeiten, durch die die Musikstücke von den Modellierern mit Metadaten versehen werden, bilden die Grundlage für die verwendete Fuzzy Logic-Suchtechnologie, die für ebenso vage Suchanfragen passende Ergebnisse liefern soll. Abbildung 2 zeigt die Benutzeroberfläche von MusicLens für den Endbenutzer mit der dreidimensionalen Visualisierung des vom Benutzer angegebenen Suchprofils.

2.3 Collaborative Filtering

Ein Ansatz, der sich von den beiden bisher vorgestellten grundlegend unterscheidet, ist der des *Collaborative Filtering*. Eigenschaften der Musikinhalte oder ihnen zugeschriebene Merkmale finden dabei im Gegensatz zu den inhaltsbasierten Ansätzen keinerlei Beachtung. Stattdessen verlässt man sich alleine auf statistische Zusammenhänge, um einem Nutzer passende Inhalte anbieten zu können. Dabei geht man von der Annahme aus, dass einem Nutzer A, der einen ähnlichen Musikgeschmack hat wie ein anderer Nutzer B, auch solche Künstler gefallen dürften, die Nutzer B zusätzlich zu den gemeinsamen Favoriten der beiden Nutzer mag. Dieser Ansatz ist wohl der populärste der drei vorgestellten und wird nicht nur in einer Vielzahl von Online-Shops als Grundlage eines Recommender-Systems benutzt, sondern auch von *last.fm*, bei dem die Benutzer mittels eines Plug-ins für ihr Musi-

Abbildung 2: MusicLens-Oberfläche [www.musiclens.de]

kabspielprogramm Informationen über die von ihnen gehörte Musik an den Server des Musikportals übermitteln. Ob einem Benutzer ein Künstler oder ein Musikstück gefällt, wird hier daran abgelesen, ob und wie häufig der Benutzer Musik von diesem Künstler gehört hat. Zusätzlich besteht die Möglichkeit, die Musikstücke, die dem Benutzer über das ebenfalls zur Verfügung stehende adaptive Webradio angeboten werden, mit einer Bewertung zu versehen.

2.4 Vergleich und Schlussfolgerungen

Aus den Unterschieden der drei beschriebenen Ansätze ergeben sich jeweils eine Reihe von Vor- und Nachteilen. Während die beiden inhaltsbasierten Ansätze - die Analyse musikwissenschaftlicher Merkmale und die vage, eher subjektive Zuschreibung von Eigenschaften - eine zum Teil sehr aufwändige, manuelle Pflege des Datenbestandes erfordern, die von Experten beziehungsweise einer größeren Zahl von qualifizierten Laien durchgeführt werden muss, kann darauf beim Collaborative Filtering-Ansatz vollständig verzichtet werden, da hier ausschließlich Algorithmen zum Einsatz kommen, um auf geeignete Inhalte zu schließen - die tatsächliche Ähnlichkeit der Inhalte spielt dabei keine Rolle. Diesem Vorteil des Collaborative Filtering-Ansatzes steht das sogeannte Kaltstart-Problem gegenüber: neue Inhalte, die bisher noch nicht von den Benutzern bewertet wurden, haben keine Chance empfohlen zu werden.

Vergleicht man die beiden inhaltsbasierten Ansätze miteinander, so ist der Aufwand der Katalogisierung beim expertenbasierten Ansatz sicherlich deutlich höher als beim Fuzzy-Ansatz. Ob diese detaillierte Analyse musikwissenschaftlicher Kriterien aber auch zu besseren Ergebnissen führt, ist fraglich. Dieser Ansatz ist gut geeignet, um bisher unbekannte Musik zu finden, die aus musikwissenschaftlicher Sicht gut zum Stil eines anderen Stückes oder Künstlers passt. Musik aus neuen Genres zu entdecken, aus denen der Benutzer bislang keinen bestimmten Künstler kennt, ist hinge-

gen schwieriger. Hier ist nicht nur der Ansatz des Collaborative Filtering im Vorteil, der aufgrund statistischer Zusammenhänge auch Empfehlungen generieren kann, die etwaigen stilistischen Ähnlichkeiten entgegen laufen. Auch der Fuzzy-Ansatz von MusicLens scheint hier geeigneter zu sein: Für die meisten Personen spielen bei der Musik Stimmungen und Gefühle eine größere Rolle als objektive, musikwissenschaftliche Kriterien. Hat der Benutzer keine klare Vorstellung davon, nach welcher Musik er sucht, sehr wohl aber von der gewünschten Stimmung, so scheint der Ansatz von MusicLens zu besseren Ergebnissen zu führen.

In unserem Forschungsprojekt verfolgen wir einen vierten Ansatz: Durch die explizite Modellierung von Beziehungen zwischen Konzepten und Instanzen der Domäne „Musik" in einer Ontologie versuchen wir, einen Wirklichkeitsausschnitt abzubilden, der den Wünschen des typischen Musikinteressierten gerecht wird. Aufgrund des hohen manuellen Modellierungsaufwandes am ehesten noch mit den beiden inhaltsbasierten Ansätzen vergleichbar, werden hier keine Merkmale der Musik selbst ausgezeichnet. Zu den Konzepten der Domäne gehören vielmehr unter anderem Künstler, CDs, CD-Rezensionen, Interviews, Berichte über oder Karten für Konzerte, Konzert-Locations oder die Heimatorte der Künstler. Aufgrund der zahlreichen Beziehungen zwischen Instanzen dieser Konzepte lassen sich so neue, auf den Benutzer zugeschnittene Inhalte finden. So lassen von mehreren Künstlern gemeinsam absolvierte Konzerte etwa darauf schließen, dass Personen, die einen der Künstler mögen, auch mit den anderen Künstlern etwas anfangen können, und Künstler, die aus dem gleichen Ort kommen, sind oft Teil einer Szene mit ihrem ganz eigenen, typischen Sound, so dass auch auf der Basis dieser Informationen passende, dem Benutzer bisher nicht bekannte Inhalte gefunden und angeboten werden können.

Durch das häufige Ansteuern der Inhalte (und einer eventuellen Bestätigung in der Form „diesen Artikel fand ich hilfreich") erkennt das System, welche Konzepte und Instanzen der Nutzer bevorzugt. Hier geht es zum einen um die Musik selbst, also um den Künstler, die Band oder das Genre und zum anderen um die Form der Inhalte, die mit der Musik zu tun haben, also darum, ob der Nutzer eher Rezensionen liest oder Interviews bevorzugt, ob er gerne Konzertkarten kauft, Neuigkeiten betrachtet oder CDs erwirbt. In Zusammenhang hierzu stehen die Elemente Zeit und Ort, die Einfluss darauf haben, ob dem Nutzer zum Beispiel eine Karte für ein Konzert angeboten wird, welches zu einer bestimmten Zeit in seiner Nähe stattfindet.

Des Weiteren sollen häufig benutzte Pfade in der Navigationshierarchie auf oberster Navigationsebene durch Shortcuts hervorgehoben und verkürzt werden. Bevorzugte Inhalte erlangen eine höhere Bedeutung im jeweiligen Nutzermodell und werden zentral auf der Webseite angeboten und im Bereich der Navigation hervorgehoben. Vom Nutzer wenig aufgerufene Inhaltskonzepte treten in den Hintergrund, bleiben aber durch die Navigation weiterhin erreichbar. Erkennt das System die Vorliebe eines Nutzers für ein bestimmtes Musikgenre, kann zudem das Design des Portals farblich und optisch angepasst werden.

3 Context Engineering

In der Vorgehensweise haben wir uns an den in der Einleitung vorgestellten Context-Engineering-Ansatz gehalten. Die Kontextmodellierung sieht vor, mit der Modellierung der Domänenontologie zu beginnen (Kaltz et al. 2005a). Wie bereits geschildert, haben wir uns für die explizite Modellierung zwischen Konzepten der Musikdomäne entschieden, die in einer in OWL [7] modellierten Ontologie münden sollte.

In der Domänenontologie ist der Informationsbestand des adaptiven Systems repräsentiert. Da die für den Nutzer relevanten Adaptionsmaßnahmen von Inhaltsauswahl, Navigation und Präsentation auf der modellierten Ontologie beruhen, ist die Erstellung der Domänenontologie der notwendige erste Schritt im Context-Engineering-Ansatz. Die Entwicklung der Ontologie erfolgte manuell.

Exkurs: OWL – Web Ontology Language

OWL [7], ein maschinenlesbarer XML-Dialekt, dient als Beschreibungssprache für Ontologien, also für eine Menge von Begriffen oder Konzepten einer Domäne und die Relationen, die zwischen ihnen bestehen. Zwar existieren auch noch andere Beschreibungssprachen für Ontologien, OWL hat sich allerdings als De-facto-Standard durchgesetzt. Es erweitert RDF (Resource Description Framework) um weitere Sprachkonstrukte.

In OWL lassen sich Konzepte (auch als Klassen bezeichnet) auszeichnen, wobei sich durch die Verwendung von subclass-Beziehungen Klassenhierarchien abbilden lassen. Klassen oder Konzepte können darüber hinaus Instanzen besitzen, ähnlich wie im Paradigma des objektorientierten Designs. Eigenschaften, die Instanzen einer Klasse haben, werden über so genannte *ObjectProperties* beschrieben, die wiederum auf andere Klassen verweisen. Dadurch lässt sich ausdrücken, dass Instanzen der beteiligten Klassen in einer bestimmten Beziehung stehen. Ein Beispiel aus der hier verwendeten Ontologie ist: `Artist hasMusician Person`. Auf Ebene der Instanzen kann dann festgelegt werden, welche konkrete Instanz der Klasse `Artist` welcher Instanz der Klasse `Person` verbunden ist.

DatatypeProperties lassen sich für Eigenschaften verwenden, die nicht auf eine andere Klasse verweisen, sondern auf einen einfachen Datentyp (*Literal*). Dabei sind alle Datentypen verwendbar, die in *XML Schema*[13] zur Verfügung stehen. Ein Beispiel dazu aus unserer Ontologie ist: `Person hasBirthday Date`.

3.1 Domänenmodell

Das Domänenmodell unseres Systems stellt eine Abbildung von in der Realität vorhandenen Beziehungen dar, indem es die vielfältigen Beziehungen unter anderen zwischen Künstlern, ihren veröffentlichten Produkten, den beteiligten Musikern und Produzenten, Herkunfts- und Auftrittsorten

beschreibt. Eine Ausnahme bilden die Zuordnung eines Künstlers oder eines Musikproduktes zu einem Musikgenre sowie die Beziehungen zwischen den unterschiedlichen Musikgenres, welche ebenfalls in der Domänenontologie modelliert werden. Die Musikgenres und ihre Beziehungen stellen ein Konstrukt dar, um Musik kategorisieren und so ohne großen Aufwand eine Vorstellung davon gewinnen zu können, wie die Musik eines bestimmten Künstlers klingt. Wenn man von dieser Ausnahme absieht, werden im Domänenmodell jedoch ausschließlich Beziehungen abgebildet, welche genau so in der Realität vorhanden sind. Aus der Kombination der Einordnung von Künstlern in Musikgenres mit solchen reichhaltigen Beziehungen erhoffen wir uns, dass im Adaptionsprozess für den Nutzer relevante Inhalte erkannt werden, welche in einem Domänenmodell, das ausschließlich auf der künstlichen Einordnung der Musik in einer Genre-Hierarchie oder anderen, subjektiven Klassifizierungen beruht, nicht gefunden würden.

Da das Domänenmodell unseres Systems aufgrund ständig hinzukommender neuer Inhalte und möglicher Änderungen bereits bestehender Beziehungen keinesfalls statisch ist, ist in der Praxis eine aufwändige manuelle Pflege des Modells nötig, wie bereits in 2.4 erläutert wurde. Wir sind jedoch der Überzeugung, dass dieser Nachteil durch die hinzugewonnenen Adaptionsmöglichkeiten, die sich aus diesen Daten ergeben, aufgewogen wird und eine solche Vorgehensweise somit gerechtfertigt ist.

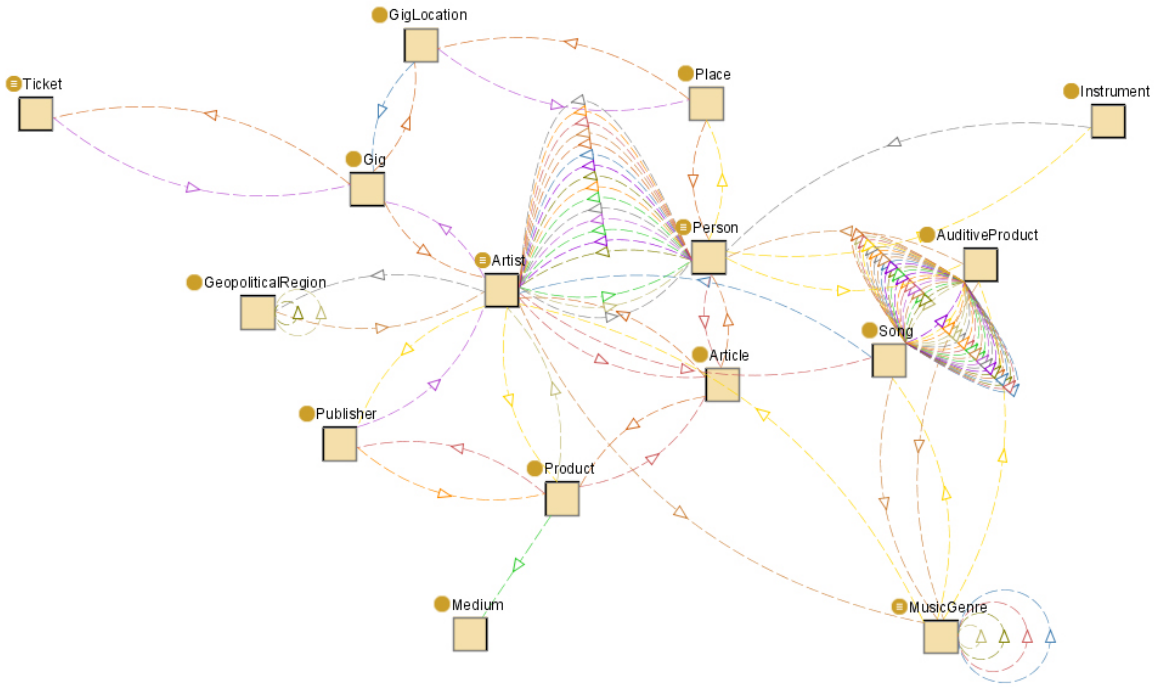


Abbildung 3: Übersicht der Konzepte des Domänenmodells und der Relationen, die zwischen ihren Instanzen bestehen

Die Konzepte, welche im Domänenmodell verwendet werden und auf der obersten Ebene der Konzepthierarchie angesiedelt sind, sowie die Relationen, welche zwischen Instanzen dieser Konzepte bestehen, sind in Abbildung 3 dargestellt. Weitere Relationen bestehen zwischen Instanzen von Unterklassen der hier gezeigten Konzepte. Diese werden, um den Graphen übersichtlich zu halten,

hier jedoch nicht dargestellt. Der aktuelle Stand des Domänenmodells, inklusive der beispielhaft verwendeten Ontologie des Black Music-Genres, befindet sich, dargestellt im OWL-Format, vollständig in Anhang **??**. An dieser Stelle sollen deshalb nur einige wichtige Aspekte des Domänenmodells erläutert werden.

Artist Wie in Abbildung 3 deutlich zu erkennen ist, ist das Konzept `Artist` dasjenige, welches die meisten Verbindungen mit anderen Konzepten aufweist. Folgende Eigenschaften können für eine Instanz dieses Konzeptes modelliert werden:

- `hasFormerMember`: Ist die Instanz von `Artist` über diese Eigenschaft mit einer Instanz der Klasse `Person` verbunden, so handelt es sich bei der Person um ein ehemaliges Mitglied der Band.
- `hasGig`: Sämtliche Konzerte, welche der Künstler gespielt hat oder bereits angekündigt sind, sind über diese Eigenschaft mit dem Künstler verbunden.
- `hasLabel`: Verweist auf die aktuelle Plattenfirma des Künstlers, eine Instanz der Klasse `Publisher`.
- `hasMember`: Über diese Eigenschaft ist die `Artist`-Instanz mit allen aktuellen Mitgliedern verbunden.
- `hasMusicGenre`: Ein `Artist` kann einem oder mehreren Musikgenres zugeordnet sein.
- `hasProduct`: Alle Produkte, die der Künstler veröffentlicht hat, sind über diese Relation mit ihm verbunden.
- `hasYearOfDisbandment`: Das Auflösungsjahr einer Band, falls diese nicht mehr existiert.
- `hasYearOfFoundation`: Das Gründungjahr einer Band, soweit bekannt.

Product, Medium und Article Von zentraler Bedeutung für das System sind neben `Artist` die Konzepte `Product`, `Medium` und `Article` sowie deren Unterklassen, welche in Abbildung 4 zu sehen sind. Es wird zwischen auditiven Produkten, visuellen Produkten und Druckerzeugnissen unterschieden. Dies ist zum einen deshalb nötig, weil über externe Dienste entsprechende Produkte angeboten werden sollen, zum anderen, da Produktrezensionen ein elementarer Bestandteil des Musikportals sind. So existiert eine Unterklasse `ProductReview` des Konzeptes `Article`, welche wiederum Unterklassen hat, welche den verschiedenen Produktarten entsprechen. Schließlich kann ein und dasselbe Produkt auf verschiedenen Medien angeboten werden, ein Album etwa kann in Form einer `CompactDisc` oder einer `VinylDisc` angeboten werden, aber auch `Downloadable` oder `Streamable` sein. Mithilfe der Unterklassen der drei Konzepte können dem Nutzer je nach seinen Präferenzen entsprechende Artikel präsentiert oder über externe Dienste Produkte angeboten werden. So wird jemand, der bislang hauptsächlich Musik in Form von Downloads erworben hat, ebensolche Produkte eher angeboten bekommen als etwa Musik, die auf Vinyl erhältlich ist.

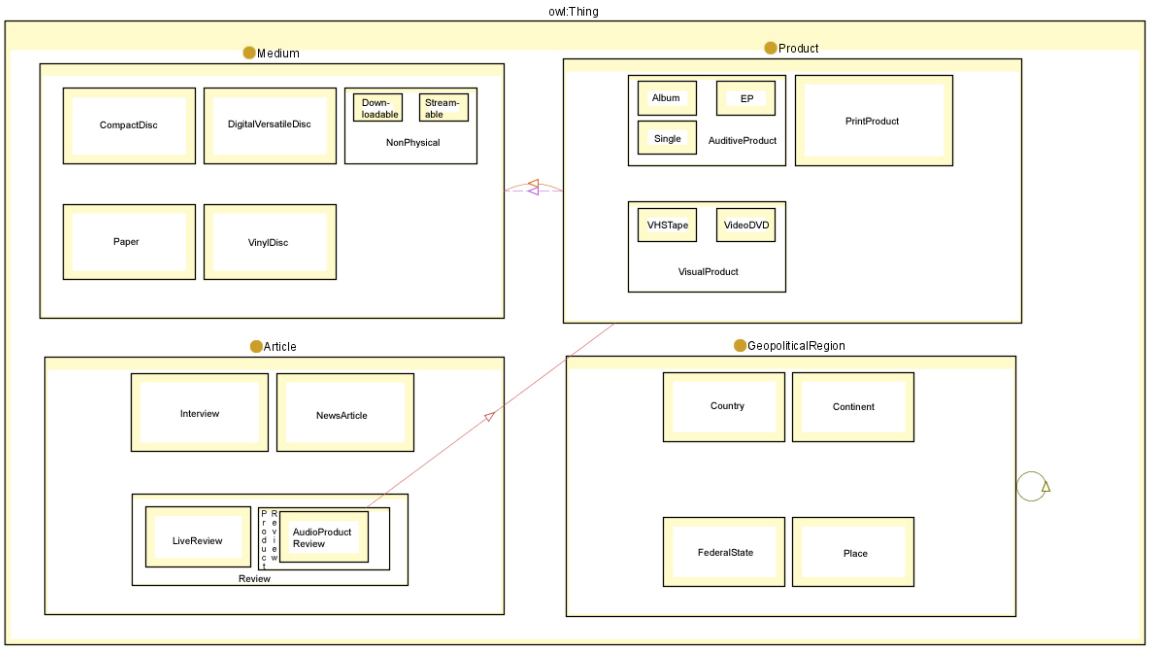


Abbildung 4: Darstellung der Konzepte Medium, Product, Article und GeopoliticalRegion sowie ihrer Unterklassen

MusicGenre Intuitiv bietet sich für die Modellierung einer Hierarchie von Musikgenres an, jedes Genre als eigenes Konzept zu betrachten, welches in einer `is-subclass-of`-Beziehung mit einem anderen Genre steht. So ist etwa das Genre „Chicago Blues" ein Untergenre von „Delta Blues", so dass das Konzept `ChicagoBlues` über eine `is-subclass-of`-Relation, wie sie von OWL angeboten wird, mit dem Konzept `DeltaBlues` verbunden. Für unsere Zwecke erwies sich eine solche Vorgensweise jedoch nicht als ausreichend. So sollen nicht nur Subgenre-Beziehungen zwischen zwei Musikgenres möglich sein, sondern auch modelliert werden, welche Genres ein gegebenes Genre beeinflusst haben, um eine historische Betrachtungsweise mit einzubeziehen. Aufgrund dieser Differenzierung werden alle Genres als Instanzen des Konzepts `MusicGenre` betrachtet, wie in Abbildung 5 am Beispiel einer Black Music-Ontologie veranschaulicht wird. Dabei sind zwischen zwei Instanzen des Konzepts `MusicGenre` folgende Relationen möglich:

- `hasChildGenre`: Eine Instanz von `MusicGenre` kann an der Entstehung beliebig vieler anderer Genres beteiligt sein und diese so beeinflusst haben.
- `hasSubgenre`: Eine Instanz von `MusicGenre` kann beliebig viele Subgenres haben, welche jeweils eine Teilmenge der Musik kennzeichnen, die vom gegebenen Genre umfasst wird.
- `isChildGenreOf`: Eine Instanz von `MusicGenre` kann durch beliebig viele andere Genres beeinflusst worden sein.
- `isSubgenreOf`: Eine Instanz von `MusicGenre` kann nur einem einzigen übergeordneten Genre zugeordnet sein, von dem es eine Teilmenge darstellt.

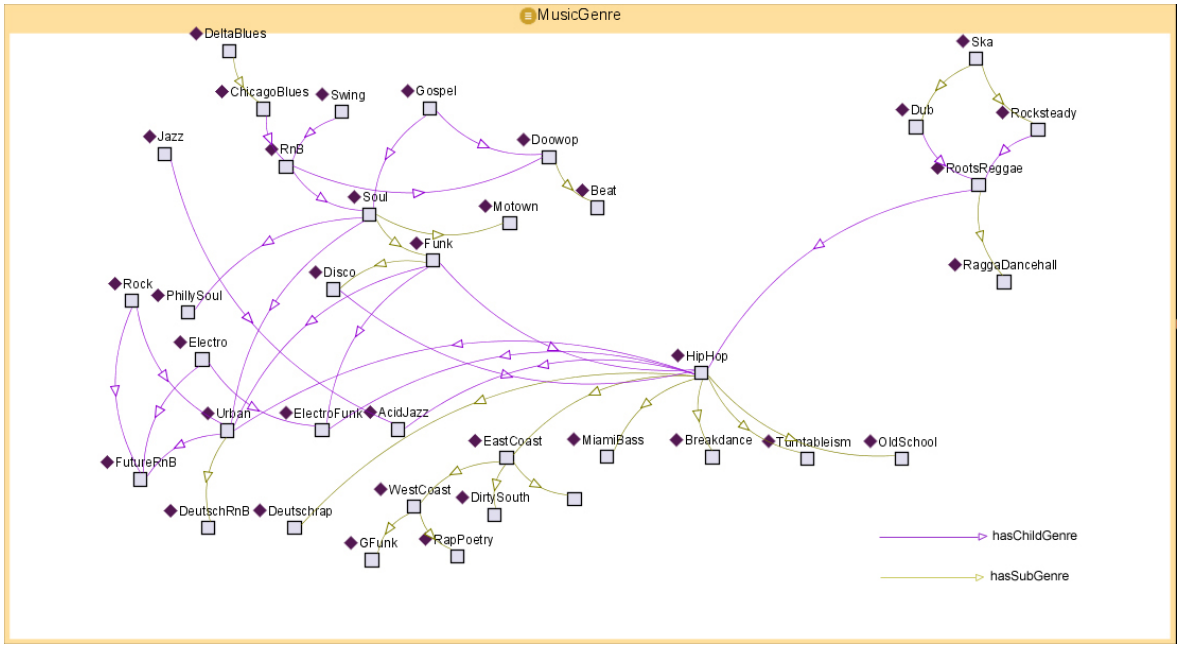


Abbildung 5: Beispielhafte Darstellung einer Black Music-Ontologie

3.2 Kontextmodell

Um die Adaptionsmaßnahmen für den Nutzer einzuleiten, müssen relevante Kontextfaktoren identifiziert werden und ebenfalls modelliert werden. In einer ersten Phase sollen Kontextontologien durch die Analyse relevanter Kontextaspekte erstellt werden. In der folgenden zweiten Phase werden adaptionsrelevante Konzepte der Domänenonotologie mit Konzepten der Kontextontologie in Beziehung gesetzt [5]. An dieser Stelle ist es nötig, über die Definition von Kontext nachzudenken. Dey et al. (2001) schlagen für den Bereich der Mensch-Computer Interaktion folgende Definition vor:

> „Context [is] any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves." (zit. n. [16]).

Winograd führt eine für uns wichtige Differenzierung der Definition ein, indem er Kontext auf die tätigkeitsrelevanten Bestandteile der Umgebung beschränkt [15]. Demnach hat eine Kontextmodellierung nicht die Aufgabe, die Welt zu modellieren, sondern sich auf die Abbildung relevanter Kontextzustände zu konzentrieren. Somit ist die Relevanz des modellierten Ausschnitts entscheidend für die Qualität der Systemadaption im konkreten Anwendungsfall. Um die Komplexität des abzubildenden Kontexts zu vereinfachen, schlagen Ziegler et al. [16] für den Bereich adaptiver Web-Anwendungen die Einteilung des Interaktionskontextes in folgende Kategorien vor:

- *Benutzer und Rolle:* In dieser Kategorie werden Gruppen oder individuelle Nutzer anhand ihrer unterschiedlichen Rollen definiert.

- *Aufgabe:* Diese Kategorie enthält den zielbezogenen Kontext wie Arbeitsaufgaben oder persönliche Zielsetzung der Nutzer.
- *Ort:* In dieser Kategorie werden Modelle über den physischen oder virtuellen (z.B. in Netzwerken) Aufenthaltsort des Nutzers beschrieben.
- *Zeit:* Soll die Zeit (Jahreszeit, Uhrzeit, Wochentag etc.) Einfluss auf die Adaption haben, so wird diese Kategorie modelliert.
- *Gerät:* Das Gerät (z.B. PDA, Handy, Desktop, im öffentlichen WLAN angebundener Laptop) kann die Adaption der Web-Anwendung beeinflussen.

3.2.1 Kontextfaktoren in unserem Anwendungsfall

Wie bereits erwähnt, ist die Relevanz des modellierten Ausschnitts der Welt entscheidend für die Qualität der Adaptionsmaßnahmen des Systems. Dies macht auch eine Beschränkung auf die relevanten Kontextfaktoren notwendig. Da sich unser System in erster Linie auf die Adaption von Inhalten hinsichtlich des persönlichen Musikgeschmacks des Nutzers konzentriert und daraufhin vielschichtige Inhalte präsentiert, verzichten wir auf die Erfassung des Geräts und Adaption auf kleine Displays wie PDA, da hierdurch eine für unser System nicht akzeptable Informationsbeschneidung entsteht.

Für unser System haben die Kontextkategorien des Nutzers Priorität: Der *Musikgeschmack* und der *Inhaltsgeschmack* des Nutzers. Letzteres bedarf einer Erläuterung: Der Inhaltsgeschmack wird durch die präferierte Lesegewohnheit des Nutzers inferiert. Liest der Nutzer beispielsweise hauptsächlich Rezensionen von Musikalben und Interviews von Künstlern, so wird ihm das System konsequenterweise ebensolche vermehrt beziehungsweise an prominenterer Stelle präsentieren.

Die Kategorie *Ort* ist in doppelter Hinsicht relevant für die Kontexterfassung des Nutzers in unserem System. Einerseits ist die Lokation des Nutzers im virtuellen Informationsbestand entscheidend. Hierdurch wird der Musikgeschmack und der Inhaltsgeschmack des Nutzers inferiert. Durch die in der Domänenontologie definierten musikalischen Zusammenhänge der Instanzen von Genres, Künstlern und Produzenten wird der Kontext des individuellen Musikgeschmacks des Nutzers inferiert. Über die Häufigkeit der Navigation des Nutzers auf die Konzepte, die den Inhaltsgeschmack definieren (Unterklassen von `Article`, `Gig` und `Product`) wird dieser inferiert. Andererseits wird der tatsächliche physikalische Aufenthaltsort erfasst, um Adaptionen diesbezüglich einzuleiten. So werden dem Nutzer zum Beispiel vermehrt Konzertdaten und Locations in seiner Nähe dargeboten. Ebenso kann das System vermehrt Produkte von Künstlern an prominenter Stelle platzieren, die aus der Nähe des Nutzers kommen. [8] zeigt auch, dass es sinnvoll ist, die Kategorie *Ort* für die Erstellung adaptiver Systeme in aufgezeigter Weise semantisch doppelt zu besetzen.

Unser System konzentriert sich auf die individuellen Präferenzen des Nutzers, so dass wir von einem User-Model im Zentrum unserer Kontextmodellierung sprechen.

Die *Zeit* des Nutzers soll ebenfalls Teil der Kontexterfassung sein. So können etwa verschiedene Zeitpunkte des Zugriffs auf das System auf vermehrte Aktivität des Nutzers in einem bestimmten Bereich analysiert werden.

3.2.2 Kontextrelationen

Um Kontextinformationen zur Adaption von Inhalten zu verwenden, werden Kontextualisierungsrelationen benötigt [16]. Diese Relationen bestehen einerseits zwischen Konzepten der Domänenontologie und andererseits zwischen Konzepten der Domänenontologie und den oben beschriebenen Kontextfaktoren.

Die Modellierung der Beziehungen zwischen Kontextfaktoren und Konzepten der Domäne sowie zwischen Konzepten der Domäne kann auf korrelativen, auf statistischen Zusammenhängen basierenden Relationen beruhen. Stellt man eine Ursache-Wirkungs-Relation fest, so spricht man auch von gerichteten Kontextrelationen [16]. Im konkreten Anwendungsfall unseres Portals stehen zum Beispiel die Instanzen des Konzeptes `MusicGenre` in korrelativen Zusammenhängen. Dies ist insofern notwendig, da dem Nutzer verwandte Instanzen präsentiert werden sollen, diese aber nicht zwangsläufig in Ursache-Wirkungs-Relation stehen müssen. Ergo können einem Liebhaber von Soul ebenso Produkte, die mit der `MusicGenre`-Instanz `Urban` verbunden sind, präsentiert werden, wie einem Liebhaber von Urban Produkte, die mit der `MusicGenre`-Instanz `Soul` verbunden sind. Abbildung. 5 zeigt das modellierte Konzept mit seinen Instanzen.

3.2.3 Kontexterfassung

Damit eine Web-Anwendung adaptive Anpassungen für den Nutzer vornehmen kann, sind komplexere Interaktionen von Kontextfaktoren und Domänenontologie im Hintergrund des Systems notwendig. Wir haben das Domänenmodell, die Kontextfaktoren und -relationen in unserem Modell bereits beschrieben. Um die Prozesse der Kontextrelationen zu verdeutlichen, werden hier anhand des in 3.2.2 geschilderten Beispiels die Kontexterfassung und die entstehenden Relationen zwischen Kontext und Domäne erläutert.

Abbildung 6 zeigt schematisch die Kontextrelationen zum oben genannten Beispiel. Die Kontextkategorien Musik- und Inhaltsgeschmack basieren auf dem Aufenthaltsort des Nutzers im Informationsraum. Dieser wird vom System bei jedem User-Request erfasst. Durch die Navigation des Nutzers auf eine bestimmte Instanz des Konzepts `Article`, in diesem Fall eine Instanz der Unterklasse `Interview`, wird eine Präferenz des Nutzers für Interviews im Allgemeinen inferiert. In der modellierten Domäne bestehen Verbindungen zwischen den Instanzen des Konzepts `Article` und den Instanzen des Konzepts `Artist`. Diese sind wiederum mit den Instanzen des Konzepts `MusicGenre` verbunden. Über diese Verbindung wird, wie in Abbildung 6 gezeigt, der Musikgeschmack durch das Folgen von semantischen Pfaden inferiert. Die dargestellten Kontextrelationen sind mit Gewichten behaftet, die die Stärke des Einflusses widerspiegeln [16]. Die Werte sind le-

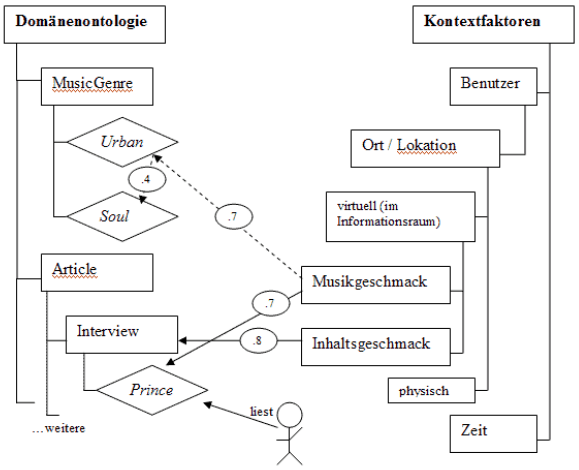


Abbildung 6: Zusammenspiel von Kontext und Domäne

diglich zur Veranschaulichung gewählt. Die genauen Werte errechnen sich durch den Ansatz des *Spreading Activation*, der erstmals von Collins und Loftus (1975) beschrieben wurde.

4 Architektur

Die Architektur unseres Systems basiert auf dem CATWALK-Framework (**C**ontext-aware **A**daption through **T**ransformations for **W**eb **A**pplications **L**everaging **K**nowledge) für kontextadaptive Webapplikationen, welches wiederum auf dem *Cocoon*-Framework [1] aufsetzt, das von der *Apache Software Foundation* bereitgestellt wird.

4.1 Cocoon

Um die Architektur und die Art und Weise verstehen zu können, in der in CATWALK, und damit auch in unserem System, Nutzeranfragen verarbeitet werden, ist es zunächst nötig, kurz auf die Besonderheiten des *Cocoon*-Frameworks einzugehen. *Cocoon* ist ein auf XML und Java basierendes Web Application Framework, welches nach dem Pipeline-Prinzip arbeitet. Bei einer Pipeline handelt sich um eine Verkettung unterschiedlicher Verarbeitungskomponenten, die jeweils ihren Anteil an der Erzeugung einer Antwort auf die Anfrage eines Clients haben und dabei bestimmte, festgelegte Aufgaben übernehmen. Nachdem ein HTTP-Request einer Pipeline zugeordnet wurde, beginnt die erste Komponente der Pipeline mit der Verarbeitung. Die Kommunikation mit der jeweils folgenden Komponente geschieht stets in Form eines SAX-Streams, also in XML. Der letzte Schritt in einer Pipeline ist schließlich die Generierung einer HTTP-Response. *Cocoon* verwendet eine Reihe von vorgefertigten Komponententypen, wobei bei der Verarbeitung eines Requests mindestens jeweils eine Komponente der folgenden Typen beteiligt ist: *Matcher* (nötig, um einen Request einer Pipeline zuzuordnen), *Generator* (erzeugt aus einer beliebigen Quelle einen XML-Stream, beispielsweise aus

einer Datei, die lokal auf dem Webserver liegt) und *Serializer* (serialisiert die HTTP-Response, üblicherweise im HTML-Format). Üblicherweise werden aber in einer Pipeline auch eine oder mehrere Komponenten des Typs *Transformer* Verwendung, welche einen SAX-Stream empfangen, die Daten auf eine bestimmte Weise verändern und diese dann an die nächste Komponente weiterreichen. *Cocoon* stellt von jedem Komponententyp bestimmte Standardimplementationen bereit. Ein verbreiteter *Transformer* transformiert etwa die eingehenden XML-Daten mithilfe eines XSLT-Stylesheets.

4.2 CATWALK

Da *Cocoon* starken Gebrauch vom *Excalibur*-Framework [2] macht, ist auch die Architektur unserer Anwendung von den durch *Excalibur* forcierten Design-Patterns, *Seperation of Concerns* und *Inversion of Control*, geprägt. Ein Vorteil, der sich daraus ergibt, ist die leichte Austauschbarkeit von Komponenten. Hier setzt CATWALK an, indem es eigene *Transformer* verwendet. Des Weiteren wird von der Möglichkeit Gebrauch gemacht, eigene Komponententypen in Form von Java-Interfaces zu definieren, für die CATWALK jeweils eine Standard-Implementation anbietet. Dadurch bietet CATWALK eine ähnliche Flexiblität wie *Cocoon*. Um die Ziele unseres Projektes zu realisieren, machen wir uns diese Flexibilität zu Nutze, indem wir einige der Standard-Implementationen der CATWALK-Komponenten durch eigene Implementationen ersetzen. Im Folgenden sollen kurz die wichtigsten CATWALK-Komponenten und ihre Rolle bei der Verarbeitung einer Anfrage vorgestellt werden.

- **Context Extractor:** Komponenten dieses Typs stehen stets zu Beginn der Verarbeitung einer Anfrage. Hier werden anhand von Daten, die dem HTTP-Request entnommen werden können, Kontextfaktoren unterschiedlicher Kategorien erkannt, darunter Ort, Zeit, Benutzer, Gerät und Domäne. Besonders wichtig sind für unser Projekt Kontextfaktoren der Kategorie Domäne: Diese werden anhand der Navigation des Nutzers erkannt. Es wird bei jeder Anfrage ein Kontextfaktor erkannt, der den Eintrag im Domänenmodell repräsentiert, auf den der Nutzer gerade navigiert hat. Für jede Kontextkategorie steht ein eigener *Context Extractor* zur Verfügung.
- **ContextReasoner:** Jeder *Context Extractor* teilt dem *Context Reasoner* mit, welche Kontextfaktoren von ihm erkannt wurden. Aufgabe des Reasoners ist es nun, anhand der erkannten Kontextfaktoren weitere Kontextfaktoren zu erschließen, die ebenfalls relevant sind und deshalb aktiviert werden sollten. Für das Erschließen weiterer Kontextfaktoren sind verschiedene Verfahren denkbar, wie etwa Case-based Reasoning oder Bayesian Networks. In unserem Projekt haben wir uns dazu entschlossen, zu diesem Zweck einen Spreading Activation-Mechanismus zu implementieren. Da die Implementation dieser Komponente den Kern unseres Projekts darstellt, wird auf das Context-Reasoning mithilfe von Spreading Activation in Teil 5 ausführlich eingegangen.
- **Navigation Transformer:** Diese Komponente ist dafür zuständig, die Navigationselemente der zu generierenden Webseite zu erstellen, basierend auf dem Navigationsmodell, welches in OWL modelliert wurde und Verweise auf Einträge im Domänenmodell enthält.

- **Content Transformer:** Analog zum *Navigation Transformer* erzeugt diese Komponente anhand des Navigationsmodells und der Einträge im Domänenmodell den Teil der Webseite, der als Inhaltsbereich definiert wurde.
- **Context Manager:** Hat der *Context Reasoner* die weiteren Kontextfaktoren erschlossen, so wird der gegenwärtige Kontextzustand an diese Komponente weitergeleitet. Andere Komponenten können den gegenwärtigen Kontextzustand wie auch vergangene Kontextzustände vom *Context Manager* abfragen.
- **Adaptation Engine**: *Content Transformer* und *Navigation Transformer*, aber auch die hier nicht beschriebenen weiteren *Transformer* aus der Präsentationsschicht, greifen auf diese Komponente zu, um die im Navigationsmodell spezifierten adaptiven Elemente mit Inhalten zu füllen. Die *Adaptation Engine* greift dazu über den *Context Manager* auf den gegenwärtigen Kontextzustand, eine Menge von Kontextfaktoren, zu und gibt an die *Transformer* eine Menge von relevanten Items zurück. Im Falle der beiden hier beschriebenen *Transformer* handelt es sich dabei um Einträge aus der Domänenontologie. Auch diese Komponente wurde in unserem Projekt durch eine eigene Implementation ersetzt, auf die aber in dieser Arbeit nicht im Detail eingegangen werden soll.

5 Context-Reasoning mit Spreading Activation

Eine Vorgehensweise, die uns für den Anwendungsfall eines adaptiven Musikportals als besonders geeignet erschien, um anhand erkannter Kontextfaktoren weitere zu erschließen, ist die der Aktivationsausbreitung (Spreading Activation). Bevor der von uns implementierte, konkrete Mechanismus im Detail erläutert werden soll, werden wir zunächst auf die Ursprünge und die dahinter stehende Idee eingehen.

5.1 Grundlagen des Spreading Activation-Ansatzes

Spreading Activation Networks sind ein Modell, welches 1975 von Collins und Loftus vorgestellt wurde [3] und dessen ursprüngliche Anwendung in der Sprachpsychologie und dem semantischen Priming lag: Verschiedene Konzepte stehen in einem Zusammenhang zu anderen Begriffen oder Konzepten. Wird nun eines dieser Konzepte aktiviert, so breitet sich diese entlang der Zusammenhänge auf andere Konzepte aus, die anschließend ebenfalls schneller und leichter abrufbar sind.

Die Konzepte und ihre Beziehung werden hier als neuronale Netze repräsentiert. Dabei handelt es sich um eine Menge von Knoten, die durch gewichtete Kanten miteinander verbunden sind, wobei das Netz durch eine Veränderung der Kantengewichte in der Lage ist zu lernen. Die einzelnen Knoten haben jeweils einen Aktivierungsgrad sowie Eingabe- und Ausgabekanten. Die Berechnung der Aktivierung eines Knotens erfolgt mithilfe einer Eingabefunktion und einer Aktivierungsfunktion, die das Ergebnis eben dieser Eingabefunktion als Eingabewert erhält. Die Eingabefunktion wird

beschrieben durch die Summe der Produkte der Gewichte der Kanten zwischen dem zu aktiviernden Knoten und den vorgelagerten Knoten mit deren Aktivierungsgrad. Gängige Aktivierungsfunktionen wiederum sind auf der einen Seite die Schrittfunktion oder die Signumsfunktion, auf der anderen Seite Sigmoidfunktionen, deren Funktionsgraphen eine S-förmige Kurve beschreiben. Eine solche Funktion wird auch in unserem Spreading Activation-Mechanismus verwendet.

5.2 Der SpreadAcContextReasoner

Der Spreading Activation-Mechanismus, der in unserem System zum Einsatz kommen soll, ist angelehnt an den Ansatz von [12], wenn diese ihn auch einen in einem anderen Anwendungskontext, der semantischen Suche im Web, verwendeten. Der Spreading Activation-Mechanismus operiert in unserem Fall auf der in OWL repräsentierten Domänenontologie. Dies geschieht mithilfe des JENA-Toolkits, welches eine Java-Schnittstelle zum Umgang mit solchen Ontologien bereitstellt. Die gerichteten Kanten zwischen den Knoten des Netzes, den Instanzen und Konzepten der Ontologie, werden in einer externen XML-Konfigurationsdatei mit Gewichten versehen. Es existieren vier vordefinierte Kantentypen:

- `has-instance`: eine Kante zwischen einer Klasse und einer Instanz dieser Klasse
- `is-instance-of`: eine Kante zwischen einer Instanz und ihrer Klasse
- `has-subclass`: eine Kante zwischen einer Klasse und einer Unterklasse
- `has-superclass`: eine Kante zwischen einer Klasse und einer anderen Klasse, deren Unterklasse die erstgenannte ist.

Kanten zwischen zwei Instanzen werden mittels *ObjectProperties* realisiert. Auch für diese können Kantengewichte angegeben werden.

Der `SpreadAcContextReasoner` stellt eine Alternative zur Standardimplementation der `ContextReasoner`-Komponente in CATWALK dar.

5.2.1 Vorgehensweise

Im Folgenden soll die Vorgehensweise des von uns implementierten `SpreadAcContextReasoners` beschrieben werden.

Ausgangspunkt ist stets die Navigation des Nutzers auf eine bestimmte Instanz oder ein bestimmtes Konzept aus dem Domänenmodell. Die `ContextExtractor`-Komponente für die Kontextkategorie *Domäne* extrahiert aus der Navigation des Nutzers einen oder mehrere Kontextfaktoren (die Einträge in der Domänenontologie, die auf dem Navigationspfad liegen, werden in abgeschwächter Form ebenfalls aktiviert). Die erfassten Kontextfaktoren der Kategorie *Domäne* werden an den Reasoner übergeben und stellen die Initialknoten für den Spreading Activation-Prozess dar. Bevor dieser Prozess beginnt, werden sämtliche Kontextfaktoren, die nach dem vorherigen Request

aktiviert waren, mithilfe der `ContextManager`-Komponente geladen, damit der Reasoner die neuen Aktivationsgrade basierend auf den vorherigen Werten errechnen kann – mit der Zeit bildet sich ein Nutzerprofil, bestehend aus einer Vielzahl von Aktivationsgraden für bestimmte Einträge der Domänenontologie.

Zu Beginn des eigentliches Spreading Activation-Vorgangs werden die Intialknoten aktiviert und in einer Prioritätswarteschlange eingefügt, sortiert nach absteigendem Aktivierungsgrad. Anschließend erfolgt die Abarbeitung der Knoten in der Schlange:

1. Die Aktivierung wird an alle benachbarten Knoten weitergeleitet, falls dies nicht durch eine Beschränkung verhindert wird (siehe 5.2.5).
2. Die neu aktivierten Knoten werden in die Prioritätswarteschlange einsortiert, falls diese nicht bereits als verarbeitet markiert sind.
3. Der Knoten, dessen Nachbarknoten nun aktiviert wurden, wird als verarbeitet markiert.
4. Der Vorgang wird abgebrochen, sobald eine vorher definierte Abbruchbedingung erfüllt wird (siehe 5.2.4). Sämtliche nun neu aktivierten Kontextfakoren sowie die bereits vorher aktivierten Kontextfaktoren werden nun an den `ContextManager` übergeben, der den neuen Kontextzustand für sämtliche anderen Komponenten verfügbar macht.

5.2.2 Die Eingabefunktion

Die Eingabe-Aktivierung eines Knotens berechnet sich aus dem vorherigen Aktivierungsgrad des Knotens, des Aktivierungsgrads desjenigen Knotens, der der derzeitige Ausgangsknoten ist, dessen Nachbarknoten aktiviert werden sollen, und aus dem Gewicht der Kante zwischen diesem Knoten und dem zu aktivierenden Knoten. Es ist gilt also für die Eingabe-Aktivierung:

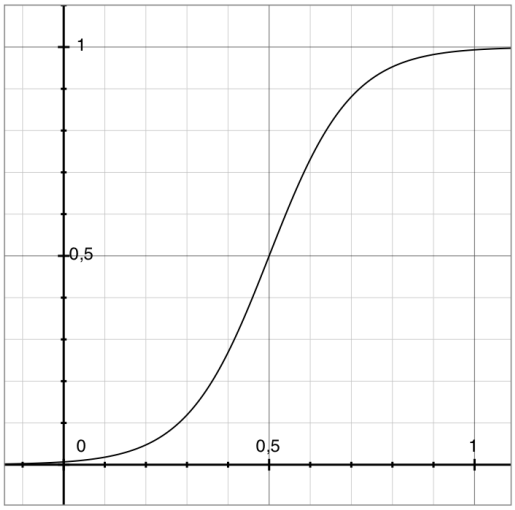
$$I_j(t+1) = A_j(t) + O_i(t) \times w_{ij} \times a \tag{1}$$

5.2.3 Die Aktivierungsfunktion

Die Aktivierungsfunktion lässt sich zwar beliebig konfigurieren, als Standard wird jedoch eine Sigmoidfunktion verwendet, genauer gesagt die logistische Funktion:

$$A_j(t) = \frac{1}{1 + e^{-w \cdot (I_j(t) - s)}} \tag{2}$$

Abbildung 7 zeigt den Kurvenverlauf der logistischen Aktivierungsfunktion unter Verwendung der als Standard vorgesehenen Werte für w und s.



$$A_j(t) = \frac{1}{1+e^{-10\cdot(I_j(t)-0.5)}}$$

Abbildung 7: Graph der logistischen Aktivierungsfunktion für $w = 10$ und $s = 0.5$

5.2.4 Abbruchbedingung

Die Abbruchbedingung lässt sich ebenso wie nahe sämtlichen anderen Parameter per XML konfigurieren. Hier kann auch eine beliebige Java-Klasse eingebunden werden, die das Interface `TerminationCondition` implementiert. Es existieren zwei vordefinierte Abbruchbedingungen:

- Erreichen eines Maximums aktivierter Knoten
- Erreichen eines Maximums verarbeiteter Knoten

Verarbeitete Knoten sind solche Knoten, die in die Prioritätswarteschlange einsortiert wurden und dann selbst Ausgangsknoten für eine Aktivationsausbreitung waren.

5.2.5 Constraints

Die Ausbreitung der Aktivierung lässt sich durch unterschiedliche Arten von Constraints kontrollieren und beschränken:

- **Concept type constraints:** Es lässt sich festlegen, dass Aktivierung entweder nicht durch Instanzen bestimmter Konzepte fließen darf oder dass sie nicht durch bestimmte Konzepte fließen darf.
- **Fan-out constraints:** Wird diese Art von Beschränkung verwendet, so fließt durch Knoten mit mehr als k Kanten keine Aktivierung.
- **Distance constraints:** Aktivierung fließt nicht durch Knoten mit einer Entfernung von mehr als k zu den Startknoten.

In unserem Projekt wird von allen drei Arten von Beschränkungen Gebrauch gemacht. So kann es etwa sinnvoll sein, dass Aktivierungen unter keinen Umständen durch das Konzept `Article` hindurch fließen. Wird beispielsweise eine bestimmte Instanz von Interview aktiviert, so fließt die Aktivierung abgeschwächt zum Konzept Interview. Würde sie nun von dort über das Konzept Article zu anderen Unterklassen von Article fließen, so käme es zu einer Verfälschung des Inhaltsgeschmacks. Während das Lesen des Interviews eigentlich dazu führen sollte, dass dem Nutzer in Zukunft allgemein vermehrt Interviews angeboten werden, würden ohne eine Beschränkung des Aktivierungsflusses andere Artikelarten, welche den Nutzer nicht interessieren, ebenfalls eine Erhöhung ihres Aktivierungsgrads erfahren. In einem solchen Fall ist also beispielsweise die Verwendung von *Concept-type contraints* angebracht.

5.2.6 Wirkungsweise

Wie der Spreading Activation-Mechanismus im Zusammenhang mit dem Domänenmodell unseres Projektes, arbeitet, lässt sich am besten anhand eines kleinen Beispiels verdeutlichen: In dem oben beschriebenen Beispiel (Abbildung 6) wird zunächst die konkrete Instanz des vom Nutzer gelesenen Interviews aktiviert. Die Aktivierungsenergie fließt dann unter anderem zum Konzept `Interview`, einer Unterklasse des Konzepts `Article`. So beeinflusst die Navigation des Nutzers den für ihn im System festgehaltenen Inhaltsgeschmack, indem der Aktivierungsgrad des Konzeptes `Interview` sich um einen bestimmten Wert erhöht hat. Propagiert wird die Aktivierung der konkreten `Interview`-Instanz aber natürlich auch auf Instanzen anderer Konzepte, zum Beispiel der Konzepte `Artist` und `MusicGenre`, und von dort beispielsweise über gemeinsame Bandmitglieder zu weiteren Instanzen von `Artist`. So werden also aufgrund des gelesenen Interviews indirekt weitere Interviews sowie weitere Künstler aktiviert, die mit dem Künstler, der Thema des gelesenen Interviews war, in einer Beziehung stehen.

Im Falle eines erfassten Kontextfaktors *Sommer* würde außerdem von dem entsprechenden Knoten Aktivierungsenergie zu benachbarten Knoten fließen. Falls bestimmte Genres mit dieser Jahreszeit verknüpft sind, beispielsweise Reggae, so würde die Aktivierung dieser Instanz von `MusicGenre` ebenfalls erhöht werden und die Wahrscheinlichkeit, dass der Nutzer in irgendeiner Form Reggae-Musik oder Artikel über Reggae-Künstler angeboten bekäme, würde entsprechend steigen.

Die Möglichkeiten, auf welche Weise Aktivierungsenergie hin zu anderen Instanzen und Konzepten fließen kann, sind aufgrund des reichhaltigen Domänenmodells vielfältig. Generell ist festzuhalten, dass die Konzepte, über die Inhaltsgeschmack inferiert werden soll, und die Instanzen der Konzepte, über die Musikgeschmack inferiert werden soll, durch *Spreading Activation* aktiviert werden. Tabelle 1 gibt einen Überblick über einige der relevanten Konzepte.

Musikgeschmack	Inhaltsgeschmack
Artist	Article
MusicGenre	Product
Publisher	Medium
Song	Gig

Tabelle 1: Für Kontextfaktoren relevante Klassen

5.3 Erstellung eines Nutzerprofils

Das System soll lernfähig sein, d.h., dass es auf zurückliegende Kontextsituationen zurückgreift, um die Adaptionen iterativ dem Musikgeschmack des Nutzers anzupassen. Hierfür greifen wir den Vorschlag von Ziegler et al. (2005) auf, den Kontext diachron zu betrachten. Durch Rückgriff auf systematisch und strukturiert im User-Model gespeicherte Informationen ist eine schrittweise Verfeinerung der Adaptionsmaßnahmen und eine Schärfung des Profils des Nutzers möglich. Hierfür ist es nötig, den Nutzer eindeutig zu identifizieren, somit ist eine Registrierung des Nutzers (Login) notwendig. Im User-Model sollen die nutzerspezifischen Aktiverungsgrade der einzelnen Knoten des *Spreading Activation*-Netzes persistent gespeichert werden. Die Kantengewichte und Kantentypgewichte sowie die Knoten und die Relationen zwischen ihnen hingegen werden global gespeichert. Die Standard-Aktiverungsgrade der Knoten aber werden stets durch die durch die im Profil des aktuellen Nutzers gespeicherten Aktivierungsgrade überschrieben.

6 Diskussion

Ziel unseres Projektes ist es, eine bessere Adaption zu erreichen als mit gängigen Adaptionsmechanismen, wie etwa auf Collaborative Filtering beruhenden Recommendersystemen. Erreicht werden soll dies durch die reichhaltige Modellierung der Domäne in Kombination mit einer Reasoning-Komponente, die den hier vorgestellten Spreading Activation-Mechanismus verwendet.

Ob mit dem beschriebenen Ansatz aber überhaupt bessere, also sinnvollere Adaptionen zustande kommen, lässt sich erst durch eine empirische Untersuchung, einen Vergleich unterschiedlicher Mechanismen, bestimmen. In einer Untersuchung konnten Stenzel und Kamps zeigen, dass bei der Ähnlichkeitsbestimmung von Musik das von ihnen zum Vergleich herangezogene Collaborative Filtering-System den ebenfalls untersuchten genrebasierten beziehungsweise inhaltsbasierten Systemen überlegen war [14]. Es scheint also, dass eine aufwändige Auszeichnung von Eigenschaften, wie in inhaltsbasierten Systemen oft nötig, nicht unbedingt zu besseren Ergebnissen führt, wenn auf der anderen Seite ein starker Algorithmus steht, der auf eine große Zahl kollaborativer Daten zugreifen kann.

Es stellt sich somit die Frage, ob der sehr hohe Aufwand, der mit dem von uns verwendeten Ansatz verbunden ist, überhaupt gerechtfertigt ist. Die Domäne muss nicht nur aufwändig modelliert werden, auch das Einpflegen neuer Inhalte ist mit einem gewissen Aufwand verbunden. Hinzu

kommt, dass die Spreading Activation-Komponente mit ihrer Vielzahl von konfigurierbaren Stellschrauben ein wenig einer Blackbox gleicht, denn wie genau sich die Änderung eines bestimmten Parameters auf das Context-Reasoning auswirkt, kann nicht immer exakt vorhergesagt werden, sondern muss im Zweifel durch Trial und Error herausgefunden werden. In jedem Fall ist es notwendig, die Spreading Activation-Komponente passend zur jeweiligen Domäne zu konfigurieren.

Kann man sich mit dem hohen Aufwand der Modellierung der Domäne und der Konfiguration der Aktivationsausbreitung abfinden, so ist die Verwendung dieses Ansatzes auch in anderer Hinsicht teuer: Die Komplexität des Algorithmus macht es schon bei Domänenmodellen mit etwa einhundert Instanzen und Konzepten erforderlich, leistungsstarke Server einzusetzen, um akzeptable Response-Zeiten zu erreichen.

Auf der anderen Seite hat der verwendete Ansatz gegenüber Collaborative Filtering-Systemen den Vorteil, dass es kein Kaltstartproblem geben kann. Kommen neue Inhalte hinzu, in unserem Fall etwa ein neuer Künstler, so wird dieser von einem Modellierer direkt anhand der vorhandenen *ObjectProperties* mit anderen Elementen der Ontologie in Beziehung gesetzt. Dieser Künstler wird somit, sobald ein Benutzer auf ein nun mit dem neuen Künstler verbundenes Element des Domänenmodells navigiert, in abgeschwächter Form aktiviert und somit schnell als relevant eingestuft.

Literatur

[1] APACHE COCOON PMC, The: *The Apache Cocoon Project*. http://cocoon.apache.org. Version: Februar 2006

[2] APACHE SOFTWARE FOUNDATION, The: *Apache Excalibur*. http://excalibur.apache.org. Version: März 2006

[3] COLLINS, A.M. ; LOFTUS, E.F:: A spreading-activation theory of semantic processing. In: *Psychological Review* 82 (1975), S. 407–428

[4] DDD DESIGN, Gesellschaft für Multimedia mbH: *MusicLens Broschüre*. http://www.ddd-system.de/pdf/musiclens_v2.pdf. Version: März 2006

[5] KALTZ, W. ; LOHMANN, S. ; ZIEGLER, J.: Eine komponentenorientierte Architektur für die kontextsensitive Adaption von Web-Anwendungen. In: *Informatik 2005 - Informatik Live!, Beiträge der 35. Jahrestagung der Gesellschaft für Informatik e.V.* Bonn : Köllen-Verlag, 2005

[6] KAPPEL, G. ; PRÖLL, B. ; RETSCHITZEGGER, W. ; SCHWINGER, W.: Customisation for ubiquitous web applications - a comparison of approaches. In: *Int. J. Web Eng. Technol.* 1 (2003), S. 79–111

[7] MILLER, Eric ; HENDLER, Jim: *Web Ontology Language (OWL)*. http://www.w3.org/2004/OWL/. Version: März 2006

[8] OPPERMANN, R.: From User-adaptive to Context-adaptive Information Systems. In: *i-com. Zeitschrift für interaktive und kooperative Medien* 3 (2005)

[9] PANDORA MEDIA, Inc.: *About Pandora*. http://www.pandora.com/corporate/. Version: März 2006

[10] PANDORA MEDIA, Inc.: *Music Genome Project*. http://www.pandora.com/corporate/mgp.shtml. Version: März 2006

[11] PANDORA MEDIA, Inc.: *Pandora*. http://www.pandora.com. Version: März 2006

[12] ROCHA, C. ; SCHWABE, D. ; ARAGAO, M.P. de: A hybrid approach for searching in the semantic web. In: *Proceedings of the 13th International World Wide Web, Conference*. New York, Mai 2004, S. 374–383

[13] SPERBERG-MCQUEEN, C.M. ; THOMPSON, Henry: *W3C XML Schema*. http://www.w3.org/XML/Schema. Version: Juli 2006

[14] STENZEL, Richard ; KAMPS, Thomas: Ähnlichkeitsbestimmung für Musik - Vergleiche und Kombinationen von Systemen. In: *Informatik Spektrum* 17 (2005), Oktober, S. 375–380

[15] WINOGRAD, T.: Archtitectures for Context. In: *Human-Computer Interaction* 16(2) (2001), S. 401–419

[16] ZIEGLER, J. ; LOHMANN, S. ; KALTZ, W.: Kontextmodellierung für adaptive webbasierte Systeme. In: STARY, C. (Hrsg.): *Mensch & Computer 2005: Kunst und Wissenschaft*. München : Oldenbourg Verlag, 2005