G. Szwillus, J. Ziegler (Hrsg.): Mensch & Computer 2003: Interaktion in Bewegung.
Stuttgart: B. G. Teubner, 2003, S. 267-276

XMendeL – Web-gestützte objektorientierte Datenhaltung im Usability-Engineering

Ronald Hartwig, Michael Herczeg
Institut für Multimediale und Interaktive Systeme – Universität zu Lübeck

Zusammenfassung

Während eines benutzer- und aufgabenzentrierten Entwicklungsprozesses werden aus der Analysephase heraus eine große Anzahl von miteinander verknüpften Informationen dokumentiert und zu Anforderungen, Entwicklungsvorschlägen, Richtlinien und Qualitätskriterien weiter entwickelt. Das im Folgenden vorgestellte Werkzeug soll dies durch ein flexibles Interface und Bereitstellung von Methoden, die sich bei der Komplexitätsbewältigung in objektorientierten Entwicklungsprozessen bereits bewährt haben, vereinfachen. Die Erfahrungen aus dem Einsatz im Rahmen eines Entwicklungsprozesses für multimediale und interaktive Lernmodule bilden den Ausgangspunkt für eine erste Bewertung der Vor- und Nachteile dieser Lösung gegenüber anderen bestehenden Ansätzen.

1 Einleitung

Im Verlaufe eines Usability-Engineering Prozesses werden eine Reihe von Daten gesammelt, aus denen Informationen zur konkreten Gestaltung des Systems abgeleitet werden. Das hier vorgestellte Tool „XMendeL“ (eine Kombination der Abkürzung XML und des Namens des Vaters der Vererbungslehre Mendel) soll helfen, die Prozessinformationen eines szenarienbasierten, nutzer- und aufgabenzentrierten Herstellungsprozesses zusammenzuhalten, semantisch zu verknüpfen und allen Prozessbeteiligten eindeutig referenzierbar zugänglich zu machen. Die hier dargestellte Objektstruktur und das Werkzeug zu seiner Erstellung, Pflege und Nutzung sind im Rahmen der Tätigkeit der Autoren als Beratungs- und Qualitätssicherungsinstanz mit dem Schwerpunkt „Software-Ergonomie“ in e-Learning-Projekten entstanden.

Ausgangspunkt ist die Umsetzung, Anpassung und praktische Verwendung eines szenarienbasierten Qualitätsmodells (Dzida et al. 2001), das auf den Ideen des „Contextual Design“ (Holtzblatt & Beyer 1996) bzw. des „Scenario Based Design“ (Rosson & Carroll 2002) und des aufgaben- und nutzerzentrierten Entwicklungsprozesses der ISO 13407 (ISO 1999) basiert. Die Grundidee ist ein strenger Ableitungsprozess, der Anforderungen und Mängel durch Rückverfolgung bis zum Ausgangsszenario konsolidiert. In (Hartwig et al. 2002) ist dieses Modell bezogen auf die Entwicklung im Kontext der virtuellen Lehre umgesetzt.

Der Eindruck der Autoren ist, dass die in diesem Prozess anfallenden Daten in der Regel entweder nicht oder als (technisch) unverbundene Einzeltexte dokumentiert werden. Ziel des hier vorgestellten Werkzeuges ist es, dies stattdessen in einem für kooperative, verteilte Kontexte, wie sie in praktisch jedem größeren Projekt existieren, geeigneten Format den Prozessbeteiligten anzubieten. Das vorgestellte Prozessmodell ist prinzipiell skalierbar, d.h. es bleibt den Prozessbeteiligten überlassen, die Granularität der zu dokumentierenden Informationen für den Prozess festzulegen. So ist es für kleinere Projekte oder bei Ressourcenmangel möglich, die Szenarien nur auf einer sehr allgemeinen Ebene zu beschreiben und nur wenige, zentrale Ziele, Anforderungen und Krite-

rien herauszuarbeiten. Um aber in komplexeren Projekten eine größere Menge Informationen handhabbar zu machen, bieten sich Methoden aus der objektorientierten Welt an. Modelliert man eine Kontextanalyse im Rahmen einer objektorientierten Analyse (OOA) so können Prinzipien wie Abstraktion, Generalisierung aber auch bewusste Unterspezifikation helfen, die Daten hierarchisch zu ordnen und zu verwalten. Der Herleitungs- und Erhärtungsprozess für Usability-Requirements ähnelt dabei dem Übergang von der objektorientierten Analyse (OOA) zum objektorientierten Design (OOD).

2 Entwicklung des Werkzeuges

Es würde den Rahmen eines solchen Artikels sprengen, den gesamten Entwicklungsprozess des vorgestellten Werkzeuges abzubilden. Deshalb werden nur ausgewählte Teile der Ziele und Anforderungen näher erläutert, die aus Sicht der Autoren besonders typisch für das Szenario der Usability-Qualitätssicherung in einem längerfristigen (im vorliegenden Falle ca. 5 Jahre im BMBF-Leitprojekt „VFH – Virtuelle Fachhochschule“ und ca. 3,5 Jahre im BMBF-Projekt „medin – Multimediales Fernstudium Medizinische Informatik“) Prozess sind.

2.1 Ziele

Zu den Aufgaben der Usability-Qualitätssicherung im Kontext der virtuellen Lehre gehört die (ergonomische) Beratung während der Entwicklung und die abschließende Bewertung der produzierten Lerneinheiten (siehe (Hartwig et al. 2002) für eine ausführlichere Darstellung der Aktivitäten). Ausgangspunkt der Entwicklung eines eigenen Werkzeuges war die Feststellung, dass die bisherige Form der Datenhaltung in Textdokumenten dabei die Beratung und Begutachtung nur ineffizient unterstützte. Gerade wegen der Vielzahl der (im Beispiel VFH ca. 15) parallel und relativ unabhängig voneinander arbeitenden Entwicklungsteams mussten viele Entscheidungen immer wieder mit Rückgriff auf frühere Aussagen bei anderen Teams getroffen werden. Das Durchsuchen von früheren Einzelfallentscheidungen wurde zunehmend schwieriger und fehlerträchtiger. Kritisch war dann das Hinzukommen von einer Vielzahl von Evaluationsinformationen aus Befragungen und teilnehmenden Begutachtungen, die sowohl einen Bezug zu den Forderungen hatten, da sie diese be- oder entkräfteten, als auch zu den Produkten, in denen Merkmale als potenzielle Mängel entdeckt wurden. Dem vorgestellten Prozessmodell folgend, müssen diese (bis dahin potenziellen) Mängel dann noch im Rahmen einer „Erhärtungsprüfung“ auf ihre Relevanz hin untersucht werden. Es zeigte sich, dass dies mit den Mitteln der Textverarbeitung nicht mehr effizient zu leisten war. Ein neues Werkzeug hat als Ziel also die Unterstützung der Arbeit des Usability-Experten im Projekt und die effiziente Datenhaltung unter Beibehaltung der vielfältigen Bezüge der Informationen untereinander.

Aus Platzgründe sind im Folgenden neben den werkzeugunabhängigen Anforderungen auch die Verweise auf die tatsächliche Implementierung und die Unterschiede zu bestehenden anderen Werkzeugen beschrieben.

2.2 Objektverwaltung

Ein Werkzeug, das die dargestellte Usability-Engineering-Methode unterstützt, muss zunächst einmal Objekte verwalten. Ein Objekt wird als Behälter für Informationen aus und für den Entwicklungsprozess genutzt. Im vorgestellten Prozessmodell sind dies zumeist Klartexte und Refe-

renzen zu anderen Objekten, sowie Zusatzinformationen zur Verwaltung des Datenbestandes (Versionen, Berechtigungen, Autoren). Als Klartexte vorhanden sind zum Beispiel Szenarien, Benutzertestprotokolle, Auswertungen von Fragebögen, Anforderungsdefinitionen oder aber auch Gestaltungsrichtlinien, die auf diesen basieren. Daneben sind auch andere Datentypen, wie zum Beispiel numerische Werte aus der Fragebogenauswertung, zu dokumentieren. Diese Anforderung erfüllen bereits einfache Textverarbeitungsprogramme und unserer Beobachtung nach ist dies auch der Stand der Technik bei der Begleitung des Usability-Engineering. Besser geeignet sind jedoch Datenbanksysteme, um so zumindest der Menge der Daten besser gerecht zu werden und den Zugriff zu beschleunigen. Aus der räumlichen Verteilung und den sehr unterschiedlichen Aufgaben und Rollen der Prozessbeteiligten ergeben sich noch die Anforderungen, dass die Informationen räumlich verteilt parallel genutzt werden können und dass rollenspezifische Sichten auf den Datenbestand ermöglicht werden. Dazu bietet sich ein web-gestütztes System an.

2.3 Referenzierbarkeit

Eine weitere Anforderung zur effizienten Arbeit mit diesen Daten ist, dass die Bezüge zwischen den Objekten, die typischerweise voneinander abgleitet sind oder sich aufeinander beziehen, erhalten bleiben müssen. Diese Forderung einer eindeutigen Referenzierbarkeit ergibt sich auch aus den Ansprüchen der Qualitätssicherung, dass Maßnahmen nachvollziehbar dokumentiert werden müssen, also im Zweifelsfall ein Dritter die Begründung, die zu einer Anforderung oder einem Kriterium führte, dokumentiert findet. Auch für den Usability-Experten ist die effiziente Navigation innerhalb der Datenmenge von zentraler Bedeutung. Dies ist in reinen Textsystemen mittels Querverweisen zwar auch möglich, doch hier ist eine Darstellung mit den Möglichkeiten des Hypertexts sicher überlegen, da die Verweismechanismen und die Betrachtungswerkzeuge (Browser) weiter entwickelt sind. Einem einfachen HTML-System fehlt aber zunächst die Möglichkeit einer (semantischen) Klassifizierung von Verweisen, um solche Links, die eine Spezialisierungsbeziehung darstellen, von solchen die einen Ableitungsbezug herstellen, zu unterscheiden. Zumeist ist dies nur durch explizite Benennung an der jeweiligen Verweisstelle möglich. Im aktuellen XHTML-Standard sind zwar auch Klassifikationen von beliebigen Auszeichnungen (und damit auch Verweisen) erlaubt und mittels CSS auch bei der Ausgabe visualisierbar, doch die betrachteten Werkzeuge unterstützten dies nur unvollkommen.

Eine Schlüsselanforderung ist, dass Verweise bidirektional sein können, d.h. dass nicht nur die Verweisquelle eine sichtbare Verbindung zum Verweisziel hat, sondern auch umgekehrt das Verweisziel eine Information darstellt, für welche Quellen es ein Ziel ist. Solche bidirektionalen Links sind in der Dokumentation eines Usability-Engineering-Prozesses wichtig, denn erst durch sie wird der bequeme Einstieg an einem beliebigen Punkt des Modells möglich. Je nach Bedarf können sowohl die aus den gerade betrachteten Objekten folgenden Konsequenzen also auch die hinführenden Überlegungen verfolgt werden. Ein weiterer Vorteil der konsequenten Nutzung von bidirektionalen Links ist auch, dass leicht identifiziert werden kann, welche Anforderung *nicht* in Form einer Regel umgesetzt wurde oder z.B. welches kritische Benutzertestergebnis *nicht* zu einer Maßnahme geführt hat. Hier entstehen neue Potenziale zur Qualitätssicherung in komplexen Projekten. In konventionellen Content-Management oder Hypertext-Autorensystemen sind solche „Rückverweise“ zumeist manuell zu pflegen und bilden so eine potenzielle Quelle inkonsistenter Datenbestände.

2.4 Vererbungsmechanismen

Die andere Schlüsselanforderung ist, dass die Möglichkeiten der objektorientierten Modellierung unterstützt werden, d.h. dass hierarchische Strukturen und Abstraktions- und Spezialisierungsmechanismen zur Beherrschung der Komplexität verwendet werden können. Eine sich daraus ergebende Nebenanforderung dazu ist die nutzergerechte Darstellung dieser Modellierung. Für einige der potenziellen Nutzer ist es notwendig, dass die Vererbung von Informationen von höheren Ebenen auf speziellere Teil-/Kind-Objekte nicht nur durch entsprechende Verweise auf Eltern-Objekte visualisiert wird, sondern dass das Werkzeug selbst, wenn gewünscht, die verschiedenen möglichen Vererbungsmechanismen anwendet und so ein in sich vollständiges Objekt erzeugt. In Abbildung 2**Fehler! Verweisquelle konnte nicht gefunden werden.** werden beispielsweise die allgemeinen Daten bzgl. der Durchführung des Benutzertests mit in die Benutzertestergebnisse hinein vererbt, so dass eindeutig klar ist, wie diese zustande gekommen sind. End-Nutzer aber auch andere Rollen im Prozess werden so entlastet, da sie nicht erst die (nicht immer sofort intuitive) Objektidee und Vererbungsmechanismen verstehen müssen, um das System zu nutzen. Aber auch für Experten vermindert die automatische Vererbung die Gefahr bei der hierarchischen Struktur den Überblick zu verlieren oder versehentlich falsche Attribute für die höher liegenden Objekte anzunehmen.

2.5 Anpassbarkeit zur Laufzeit

Die letzte aber ebenfalls entscheidende Anforderung entsteht daraus, dass das Prozessmodell selbst während des Projektes noch in der Entwicklung war und dass die Objektstruktur und die in den Objekten enthaltenen Informationen selbst während ihrer Anwendung angepasst werden mussten. Es zeigte sich, dass eine direkte Übernahme von Analysedatenformaten (bspw. (Hackos & Redish 1998), (Pradeep 1998), (ISO9241 1996) oder (Dzida et al. 2001)) und Objektmodellen (bspw. aus der OOA (Balzert 2000) oder spezielleren Anwendungsfeldern (Herczeg 1999)) aus dem Arbeitssystemkontext in den Kontext der virtuellen Lehre nicht möglich war. Die Anforderung an das System ist also, dass die Datenstruktur flexibel ist und mit wenig Aufwand während des Entwicklungsprozesses angepasst werden kann. Außerdem sollen bereits erstellte oder externe Dokumente (z.B. Textverarbeitungsdateien, Bilder, Filme) eingebunden und weiter genutzt werden können und umgekehrt sollen die Daten aus dem Werkzeug wieder in ein Standardformat exportiert werden können, um so die Nachhaltigkeit auch bei einem Scheitern des Werkzeuges oder in anderen Kontexten sicherzustellen.

2.6 Umsetzung als „XMendeL“

Zum Zeitpunkt des Projektbeginns erschien es so, dass die oben beschriebenen Anforderungen, insbesondere die explizite Vererbung nicht operationalisierter Inhalte, von bisherigen Werkzeugen nicht erfüllt werden könnten. Dies war ausschlaggebend für die Neuentwicklung eines entsprechenden Systems. Es wurde ein System entwickelt, das die Daten des Entwicklungsprozesses in einer zentralen sequentiellen Datenbank mit XML ausgezeichnet bereit stellt. Die Ein-/Ausgabe der Daten erfolgt über eine Web-Schnittstelle unter Verwendung von Java-Servlets und zur Laufzeit generiertem (dynamischem) HTML (siehe Abbildung 1).

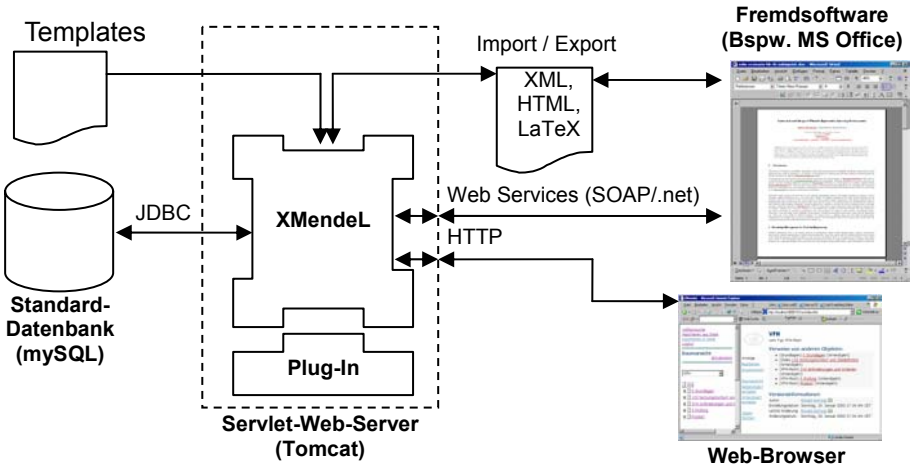

Abbildung 1: Systemaufbau mit den verschiedenen Kommunikationskanälen (nach (Kutsche 2000))

Neben dem Export der Daten in Dateien wird auch die direkte Kommunikation von Anwendungen mit dem System als Web-Service oder die anwendungsspezifische Erweiterung über eine eigene Java-Servlet-Plugin-Schnittstelle angeboten. Dadurch können auch Programme wie Microsoft Office™ über die Mechanismen von „.net“ direkt mit dem System kommunizieren und die Mechanismen der Vererbung und einer verteilten Webanwendung nutzen. Letzteres befindet zurzeit gerade im Aufbau, da die notwendigen Technologien noch relativ neu sind.

In Abbildung 2**Fehler! Verweisquelle konnte nicht gefunden werden.** ist die Standard-Datensicht zu sehen. Auf der linken Seite befindet sich eine Übersicht über die Objekte. Die Vererbungsbeziehung ist dabei als Baumstruktur mit den gewohnten Mitteln dargestellt. Die ebenfalls mögliche Mehrfachvererbung wird dabei nicht abgebildet. Sie wurde im dargestellten Projekt bisher auch nicht verwendet. Mittels direkter Manipulation können die Objekte zur Betrachtung oder zur Bearbeitung geöffnet oder innerhalb der Hierarchie verschoben und geordnet werden.

Auf der rechten Seite befindet sich die Textansicht des eigentlichen Objektes. Jedes Objekt hat einen Titel, ein übergeordnetes Objekt und eine veränderbare Anzahl von Attributen, die mit (formatierten) Texten, Zahlen, internen und externen Verweisen sowie eingebundenen Bildern und anderen Fremdformaten gefüllt sein können. Am Ende jedes Objektes wird angezeigt, welche Objekte auf das gerade betrachtete Objekt zeigen („Verweise von anderen Objekten“). Dabei wird sowohl die Art des Objektes (z.B. „Benutzertestbeobachtung“), sein Titel, als auch die Art des Verweises (Spezialisierung im Sinne der Vererbung „Unterobjekt“ oder Querverweis „link“) angegeben. Zusätzlich wird noch angezeigt, wann und von wem das Objekt erzeugt bzw. geändert wurde.

Die Daten werden über diese Web-Schnittstelle eingegeben und gepflegt. Dabei wird eine einer Textverarbeitung nachempfundene Werkzeugleiste angeboten und die Texte nach dem WYSIWIG-Prinzip bereits bei der Bearbeitung in ihrer späteren Darstellung angezeigt. Externe Inhalte können entweder als gemeinsam nutzbare Ressourcen auf dem Server des Systems gespeichert werden oder werden als externe Links in Form einer URL eingebunden. Umgekehrt kann jedes Objekt im System per URL (http://server/ObjektID&Sicht) von externen Anwendungen aus eineindeutig referenziert werden.

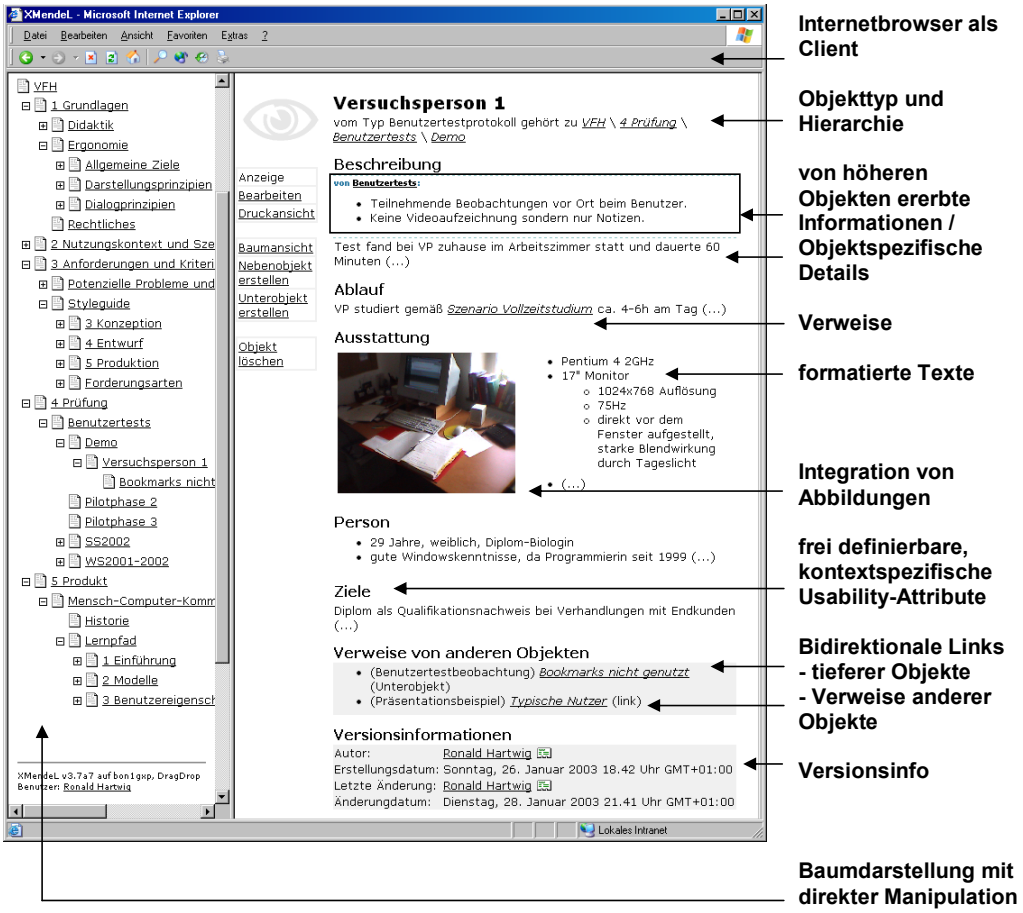

Abbildung 2: Screenshot mit gekürzten Beispieldaten

Objekte werden entweder einem bereits bestehenden Objekt untergeordnet, gleichrangig daneben erstellt oder als generische neue Objektklassen auf oberster Ebene neu erzeugt. Attribute und ihre Inhalte werden an untergeordnete Objekte weitervererbt, indem sie den zu den Inhalten des untergeordneten Objektes hinzugefügt werden. Die Vererbung der Inhalte kann aber auch für Teilbäume beschränkt und für bestimmte Attribute und Ansichten ganz abgeschaltet oder auf Ersetzen statt Hinzufügen umgestellt werden.

Die Nutzer können zu jeder Zeit die Art der Objekte definieren, indem sie „Objekttypen“ zuweisen. Dies ist frei definierbar und dient als Texthinweis sobald das Objekt angezeigt wird. Die Attribute sind bis auf wenige, allen Objekten gemeinsame, Pflichtfelder ebenfalls zur Laufzeit frei definierbar. Während der Nutzung können also an jeder beliebigen Stelle der Objekthierarchie Attribute hinzugefügt werden, ohne dass Änderungen am Programm oder Anpassungen der Datenbank notwendig wären. Diese werden dann so lange nach Default-Vorgaben angezeigt bis spezielle Darstellungsregeln (ebenfalls hierarchisch) festgelegt werden.

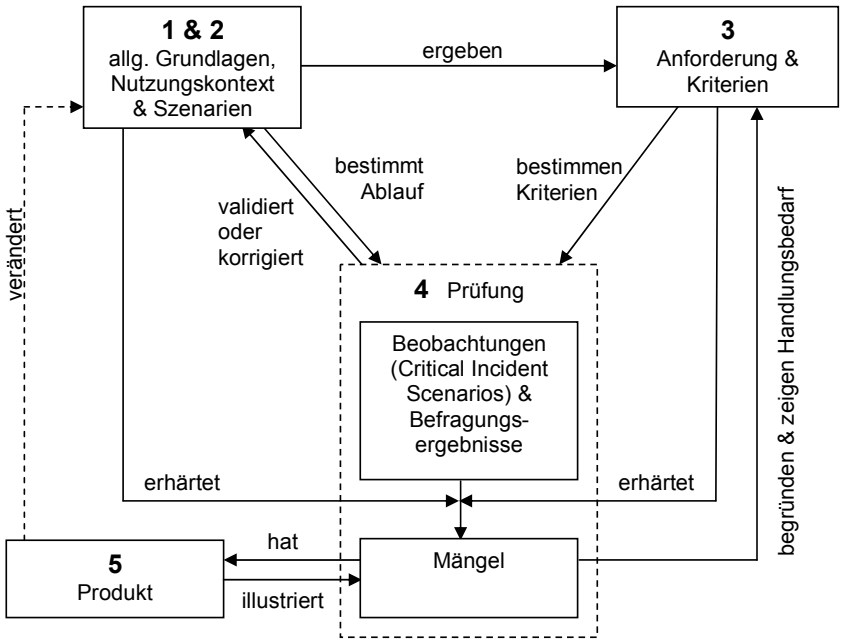

Abbildung 3: Struktur der Objektklassen und die Semantik der Querverweise

Das System verfügt daneben über weitere Schnittstellen zu externen Anwendungen: Die Daten können aus anderen Systemen als CSV-Dateien (z.B. um Exporte aus Datenbanken entgegenzunehmen), als komplette HTML-Website (z.B. zur Übernahme von externen Inhalten) oder als XML importiert werden. Import und Export arbeiten über den gleichen Mechanismus und können ohne Änderung des eigentlichen Systems über ein eigenes CSS-ähnliches Vorlagensystem (Templates) an die individuellen Bedürfnisse angepasst werden. Dabei werden neben XML-Dialekten auch Formate wie LaTeX oder andere SGML-basierte Format prinzipiell unterstützt. Der Export in eine HTML-Struktur inklusive der enthaltenen zusätzlichen Ressourcen, stellt sicher, dass die Daten auch ohne das XMendeL-System lesend, zum Beispiel mobil, genutzt werden können.

2.7 Objektstruktur

Aus der praktischen Nutzung des Systems ergab sich eine Objektstruktur, die sich zwar an die typischen Phasen des dargestellten Prozesses anlehnt, deren Detailliertheit und Ausführlichkeit aber stark variiert. Es zeigte sich, dass viele gemäß Vorgehensmodell wichtige Informationen in der Praxis nicht in der gewünschten Ausführlichkeit innerhalb des vorgegeben Ressourcenrahmens analytisch oder empirisch zu entwickeln waren (siehe (Hartwig et al. 2002) für eine detaillierte Beschreibung). Die in der Nutzung identifizierten grundlegenden Objektklassen sind im Folgenden kurz beschrieben.

1. **Grundlagen**: Im Rahmen des Projektes waren Grundlagenhandbücher und Handreichungen verschiedener Teilbereiche (Ergonomie, Didaktik, Technik etc.) entstanden. Es zeigte sich, dass viele der späteren Forderungen sich auf diese bezogen und es deshalb, auch mit Blick auf die spätere gemeinsame Nutzung als Wissensbasis für den Entwicklungsprozess, vorteilhaft ist, wenn diese auch im System enthalten und ebenfalls objektorientiert modelliert sind. So gibt es im Bereich „Ergonomie“ neben einem allgemeinen

Objekt mit der Beschreibung von „Ergonomie“ noch Unterobjekte zu den Dialog- und Darstellungsprinzipien, sowie den dort wiederum untergeordneten Beispielen.

2. **Nutzungskontext** und **Zieldefinitionen**: Unter diesem Bereich wurden alle Objekte zusammengefasst, die mit Vorgaben des Projektes aber auch z.B. durch teilnehmende Beobachtung entwickelte Nutzungsszenarien betrafen. In (Hartwig et al. 2002) ist beschrieben, wie dabei mit Problemen der Szenarienerhebung im Rahmen einer, zunächst hypothesengestützten, iterativen Weiterentwicklungsstrategie umgegangen wurde.

3. **Anforderungen** und **Kriterien**: Ausgangspunkt für das Projekt ist ein umfangreiches Richtliniendokument „VFH-Styleguide“ (Hartwig et al. 2002a), in dem sowohl Kriterien als auch Anforderungen dokumentiert sind. Dieser ist gegliedert nach den Prozessphasen, in denen die jeweiligen Richtlinien anzuwenden sind und mittels der Abstraktion strukturiert von allgemeinen Anforderungen bis hinunter zu Einzelkriterien.

4. **Prüfung**: In diesen Objekten sind die Ergebnisse der Befragungen und Beobachtungen abgelegt. Dabei sind zunächst die Rohdaten (Befragungsauswertung und Roh-Protokolle der Vor-Ort-Benutzertests im Sinne von so genannten „Critical Incident Scenarios“) dokumentiert worden. Daraus wurden Beschreibungen allgemeiner potenzieller Mängel entwickelt, die die Beobachtungen mit den Ergebnissen der Befragung als auch den Grundlagen, dem Nutzungskontext (inkl. Szenario) und den Zielen verknüpfte und so die geforderte Erhärtungsprüfung ermöglichte. Diese potenziellen Mängel dienten dann, sofern sie sich als begründet erhärten ließen, als Begründung für die Richtlinienweiterentwicklung und Bewertung der Produkte (hier Lernmodule). Diese „Mangelzentrierung“ spiegelt die Idee des Falsifikationsprinzips des Qualitätssicherungsprozesses wieder, d.h. es wird von der Tauglichkeit des Produktes ausgegangen bis tatsächlich relevante Mängel dagegen sprechen.

5. **Produkt**: Bei den hier betrachteten Lernmodulen gibt es den Sonderfall, dass diese selbst sich sehr gut und effizient mit den Mitteln der objektorientierten Datenhaltung handhaben lassen. Metainformationen oder z.B. weiterführende Übungsaufgaben lassen sich sowohl auf Gesamtmodulebene als auch auf Kapitel- oder Seitenebene festlegen. Durch das flexible Interface können nun Konzeptioner und Autoren ihre Inhalte selbst webgestützt, mit den gewohnten Mitteln der Textverarbeitung und dem WYSIWYG-Prinzip folgend verwalten, ohne mit der unterliegenden XML-Auszeichnung in Kontakt zu kommen. Außerdem können weitere Produktionsinformationen, z.B. Drehbücher mit dokumentiert werden. In (Kritzenberger et al. 2001) ist dieses Vorgehen im Detail beschrieben.

Abbildung 3**Fehler! Verweisquelle konnte nicht gefunden werden.** zeigt, welche Zusammenhänge bei der Dokumentation des QS-Prozesses verwendet werden. Ohne die Zuhilfenahme der Abstraktionsebenen zur Zusammenfassung aber auch Vereinfachung der teilweise sehr detailliert vorliegenden Informationen, wäre ein solches Modell kaum handhabbar.

3 Zusammenfassung

Der Hauptnutzen im praktischen Einsatz war zunächst die Verwaltung der Evaluationsdaten (Benutzertestprotokolle und Befragungen), die sich mit den Mitteln von Hypertext und Vererbung effizienter gestaltete als die bis dahin genutzte Textverarbeitungsmethode. Gerade die Klassifizierung von vielen kleinen Textbausteinen durch einen Usability-Experten wird durch die bidirektionalen Links und hierarchische Strukturen wesentlich übersichtlicher und schneller. Der Rest des

Objektmodells wächst nun von diesem Schwerpunkt ausgehend und wird weiter ausgebaut. Zur Zeit sind ca. 500 Objekte allein zum Projekt VFH in der Datenbank enthalten und der Bestand erweitert sich ständig.

Das endgültige Ziel ist der konsequente Einsatz als Experteninformationssystem und die Verknüpfung der Usability-Engineering-Daten mit den Objekten des Produktes. Dadurch ist es möglich, dass zum Beispiel Mängel direkt auf das Objekt des Produktes zeigen können, auf denen sie auftreten. Umgekehrt kann durch die Bidirektionalität der Verweise auf jeder Modulseite nachgesehen werden, welche Mängel noch enthalten sind. Zu guter Letzt kann in der Mangelbeschreibung nachgesehen werden, wo sich ein Beispiel für diesen Mangel findet. Diese umfassendere Projektdokumentation wird derzeit im Rahmen des e-Learning-Projektes „medin“ erprobt. Daneben wird im Laufe einer gesonderten Diplomarbeit das bisher sehr generische, damit aber auch nicht immer optimale, Interface des Werkzeuges weiter auf die speziellen Bedürfnisse eines e-Learning-Herstellungsprozesses hin angepasst.

Das Werkzeug gibt keine Modellierung fest vor, so dass auch andere Modellierungsarten zum Beispiel unter Berücksichtigung eines Schichtenmodells (siehe (Herczeg 1994)) oder andere Kontexte (bspw. Anwendungssoftwareentwicklung) erprobt werden sollen.

Literatur

Balzert, H. (2000): *Lehrbuch der Software-Technik: Software-Entwicklung*. (Bd. 1, 2. Aufl.), Heidelberg: Spektrum Akademischer Verlag

Dzida, W.; Hofmann, B.; Freitag, R.; Redtenbacher, W.; Baggen, R.; Geis, T.; Beimel, J.; Hartwig, R.; Hampe-Neteler, R.; Peters, H. (2001). Gebrauchstauglichkeit von Software. ErgoNorm: Ein Verfahren zur Konformitätsprüfung von Software auf der Grundlage von DIN EN ISO 9241 Teile 10 und 11. *Schriftenreihe der Bundesanstalt für Arbeitsschutz und Arbeitsmedizin*, Forschung Fb 921, Bremerhaven: Wirtschaftsverlag NW

Hackos, J. T.; Redish, J. C. (1998): *User and task analysis for interface design*, New York: Wiley & Sons

Holtzblatt, K.; Beyer, H. (1996): *Contextual Design: Principles and Practice - Field Methods for Software and Systems Design*. D. Wixon and J. Ramey (Eds.), New York: John Wiley & Sons, Inc.

Hartwig, R.; Triebe, J.K.; Herczeg, M. (2002): Software-ergonomische Evaluation im Kontext der Entwicklung multimedialer Lernmodule für die virtuelle Lehre. In: Herczeg, M; Prinz, W.; Oberquelle, H. (Hrsg.): *Mensch & Computer 2002*: Vom interaktiven Werkzeug zu kooperativen Arbeits- und Lernwelten. Stuttgart: B.G. Teubner, 2002, S.313-322

Hartwig, R.; Triebe, J.K.; Herczeg, M. (2002a): *Styleguide - Richtlinien zur Qualitätssicherung bei der Realisierung von Studienmodulen im Projekt VFH*. Universität zu Lübeck - Institut für Multimediale und Interaktive Systeme; http://www.imis.uni-luebeck.de/de/forschung/publikationen.html#2002

Herczeg, M. (1999): A Task Analysis Framework for Management Systems and Decision Support Systems. In: Proceeding of AoM/IaoM. 17. *International Conference on Computer Science*, *San Diego*. California, August 6-8, S. 29-34

Herczeg, M. (1994*): Software-Ergonomie. Grundlagen der Mensch-Computer-Kommunikation.* Bonn: Addison-Wesley-Verlag und München: Oldenbourg-Verlag

International Organization for Standardization (1996-2000): *ISO 9241 - Ergonomic requirements for office work with visual display terminals*, Parts 1-17, Berlin: Beuth Verlag

International Organization for Standardization (1999): *ISO 13407 - Human-centred design processes for interactive systems*. Berlin: Beuth Verlag

Kutsche, O. (2002): *Proof-of-concept der datenbank- und web-basierten Unterstützung von Entwicklungsprozessen für einen Prototypen*. Studienarbeit Informatik, Universität zu Lübeck

Kritzenberger, H.; Hartwig, R.; Herczeg, M. (2001): *Scenario-Based Design for Flexible Hypermedia Learning Environments*. In: Proceedings of ED-MEDIA 2001. AACE, Tampere, Finland, 25.-30. Jun. 2001

Pradeep, H. (1998): *User-centered information design for improved software usability* . Norwood, USA: Artech House

Rosson, M. B.; Carroll, J.M. (2002): *Usability Engineering – Scenario based development of human-computer interaction*. San Francisco, USA: Morgan Kaufmann Pub.

Kontaktinformationen

Ronald Hartwig, Michael Herczeg
Institut für Multimediale und Interaktive Systeme
Universität zu Lübeck
Media Docks, Willy-Brandt-Allee 31a
D-23554 Lübeck
hartwig|herczeg@imis.uni-luebeck.de
http://www.imis.uni-luebeck.de