

# Visuelle Spezifikation zur Stärkung der Auftraggeber-Kompetenz bei der Gestaltung interaktiver Systeme

**Thomas Memmel**  
Mensch-Computer Interaktion  
Universität Konstanz  
78457 Konstanz  
memmel@inf.uni-konstanz.de

**Harald Reiterer**  
Mensch-Computer Interaktion  
Universität Konstanz  
78457 Konstanz  
reiterer@inf.uni-konstanz.de

**Heiko Ziegler**  
Daimler Chrysler AG  
GR/ESP  
89013 Ulm  
heiko.ziegler@daimlerchrysler.com

**Richard Oed**  
Daimler Chrysler AG  
GR/ESP  
89013 Ulm  
richard.oed@daimlerchrysler.com

## Abstract

Der hohe Stellenwert gebrauchstauglicher Benutzungsschnittstellen wird von immer mehr Unternehmen erkannt und erfordert eine Anpassung verbreiteter Vorgehensweisen bei der Anforderungsermittlung und Systemspezifikation. Der Auftraggeber kann sich nicht darauf beschränken, die System- und Benutzeranforderungen zu definieren.

Der Auftragnehmer ist oft nicht in der Lage, selbständig ein benutzerfreundliches System nach den exakten Vorstellungen des Auftraggebers zu gestalten (Offergeld & Oed 2006). Wir stellen Vorgehensmodelle und Methoden zur Erzeugung interaktiver Prototypen und Systemspezifikationen vor. Durch eine visuelle Spezifikationen

können Kommunikations- und Kompetenzprobleme überwunden werden. Entwicklungszeiten werden bei gleichzeitiger Steigerung der Qualität verkürzt und die Zusammenarbeit optimiert.

## Keywords

Visuelle Spezifikation, User Interface Design

## 1.0 Einleitung

In (Offergeld & Oed 2006) haben wir erläutert, dass der konzeptionelle User Interface (UI) Entwurf bis hin zur Erstellung von dazugehörigen UI Prototypen immer öfter auch zu den Usability Engineering (UE) Aktivitäten des Auftraggebers (AG) gehört.

Normalerweise beschränkt sich die Tätigkeit des AG auf typische Formen der Anforderungsermittlung (engl. Requirements Engineering, RE). Unterschiedliche Fachabteilungen einigen sich auf funktionale Anforderungen und befragen Endbenutzer nach ihren Bedürfnissen. Anschließend werden die Anforderungen geprüft (Quality Gate) und in ein Lastenheft aufgenommen. In der Regel sind zu diesem Zeitpunkt sehr viele unterschiedliche Akteure (z.B. Endbenutzer, Fachabteilungsleiter, Funktionsverantwortliche, Produktverantwortliche, Marketing-, Domänen-, Software- und Ergonomie-Experten) involviert. Die unterschiedlichen Akteure drücken sich im Spezifikationsprozess oftmals mit unterschiedlichen Artefakten aus, wodurch gerade bei textuellen Spezifikationen Mehrdeutigkeiten und Ungenauigkeiten

entstehen können (Mommel et al. 2007a). Darüber hinaus sorgen knappere Entwicklungsbudgets und kürzere Entwicklungszeiten einerseits, sowie hohe Qualitätsanforderungen an Softwareprodukt und Softwareentwicklungsprozess andererseits, für ein herausforderndes Spannungsfeld. Fachabteilungen für RE generell und UE im Speziellen müssen geeignete Methoden und Werkzeuge zur Hand haben, mit deren Hilfe in kürzerer Zeit kostengünstige UIs von hoher Qualität spezifiziert werden können.

Daher ist es vor allem bei Anwendungen mit hohem Stellenwert des UI – zum Beispiel Webseiten oder Fahrerinformationssystemen – wirtschaftlich riskant, die Interpretation der Spezifikation und die Gestaltung der Benutzungsoberflächen externen Entwicklern zu überlassen. Obwohl die Tendenz hin zu einer Industrialisierung der IT eine Reduktion der internen Fertigungstiefe vorschreiben würde, und zu Offshoring sowie Outsourcing Prozesse anregt (Cappgemini 2007), muss beim UI Design mit Bedacht vorgegangen werden.

Es hat sich in der Industrie bewährt, neben der üblicherweise textuellen Beschreibung der Systemfunktionalitäten auch eine visuelle Spezifikation des interaktiven Systems für den IT Zulieferer (Auftragnehmer, AN) zu erstellen. Durch das Erleben der Spezifikationen werden auch fehlende Anforderungen noch rechtzeitig erkannt und ein Scheitern der Beauftragung vermieden (Standish Group 2003; Potts et al. 1994; Zave & Jackson 1997).

Dazu werden zum Beispiel UI Prototypen bereits auf AG-Seite entwickelt und unter Beteiligung von Nutzerrepräsentanten in Usability Tests auf Gebrauchstauglichkeit, aufgabengerechte Gestaltung und User Experience (UX) hin geprüft. Am Ende dieses Prozesses wird der Prototyp dann als visueller Teil der AG-Anforderungen an den AN zur Realisierung übergeben. So kann eine kostengünstige Offshore-Abteilung oder ein externer Dienstleister an einem Offshore Standort eine Benutzer-, Aufgaben- und vor allem Spezifikationsgerechte Umsetzung vornehmen.

In Abschnitt 2.0 wollen wir zunächst ein konkretes Werkzeug vorstellen, mit dem wir derzeit bei DC visuelle UI Spezifika-

tionen erstellen. Anschließend erläutern wir unsere gelebte Praxis in Abschnitt 3.0 im Kontext eines geeigneten Vorgehensmodells. Schließlich geben wir einen umfassenden Ausblick über die Weiterentwicklung und Erweiterung unserer visuellen Spezifikationsmethoden. In Abschnitt 4.0 stellen wir dazu ein zweites Werkzeug vor. Der Beitrag endet mit einer kurzen Zusammenfassung.

## 2.0 Das Werkzeug MAXpro

Wir finden in Spezifikationsprozessen in der Regel eher heterogene Werkzeuglandschaften vor. Darunter befinden sich neben Office Anwendungen wie Word, PowerPoint und Excel auch Bildbearbeitungsprogramme und Animationswerkzeuge (Adobe Flash, Adobe Flex). Techniker verwenden komplexere (CASE-)Werkzeuge (Memmel et al. 2007a). Die Verwendung visueller Spezifikationen und der korrespondierenden Werkzeuge führt zusätzlich zur eindeutigeren Bestimmung von Aussehen *und* Verhalten eines UI auch dazu, dass die unterschiedlichen Akteure ein einheitliches Ausdrucksmittel erhalten (Standardisierung). Im Folgenden fassen wir die wichtigsten Anforderungen an das von uns entwickelte Werkzeug namens MAXpro (Modellierung – Agile – XML-basiert) zusammenfassen:

- Die visuelle Spezifikation in Form eines interaktiven Prototypen ersetzt soweit wie möglich textbasierte Anforderungsdokumente
- Schnelle, generische Änderungen und unterschiedliche Varianten des UI können einfach und schnell erprobt werden (Rapid Prototyping)
- Frühzeitige Prototypen erlauben frühe Evaluationen von Aussehen und Verhalten vor der Umsetzung
- Späte und teure Änderungen werden vermieden (Kostenreduktion)
- Eine agile, iterative und inkrementelle Vorgehensweise wird unterstützt
- Der Prototyp kann im Webbrowser angezeigt werden und erlaubt so einfache Feedbackprozesse

- Die visuelle Spezifikation kann in einem maschinenlesbaren Austauschformat an den AN weitergegeben werden
- Die visuelle Spezifikation ist ein evolutionärer Prototyp, der auf AN-Seite einfach in ein finales System überführt werden kann

MAXpro ist ein domänenspezifisches Werkzeug, welches nach der Erhebung grundsätzlicher Modelle des RE für die konkrete UI Entwicklung eingesetzt wird. Die visuellen Spezifikationen unterscheiden sich von frühen Prototypen insofern, dass Sie eine verbindliche Spezifikation darstellen. Sind die grundlegenden (XML)Modelle für ein komplexes Projekt erst einmal erzeugt, lassen sich UI Spezifikationen schnell und kostengünstig erzeugen (siehe Kapitel 2.2). Sie dienen als Kommunikationsmittel, dienen zum frühen Testen von abstrakten Varianten, sowie zur Evaluation von detaillierten Designs. Visuelle Spezifikationen mit MAXpro integrieren daher die Vorteile von Low- und High-Fi Prototypen (Rudd et al. 1996).

## 2.1 Abgrenzung zu anderen Werkzeugen

Eine von uns durchgeführte Untersuchung von 148 Entwicklungswerkzeugen hat gezeigt, dass die meisten Werkzeuge für visuelle Spezifikationen nicht geeignet sind. Viele CASE-Werkzeuge erlauben zwar die Überführung von Modellen - wie z.B. UML - in ausführbare Prototypen. In der Regel entstehen mit solchen Werkzeugen (z.B. im Kontext von Modell-getriebener Entwicklung) lediglich Code-Frameworks oder sehr detailarme Dialogsysteme. Solche Simulation reichen nicht aus, um einen adäquaten Eindruck und ein Erleben des UI zu erlauben. Solche Werkzeuge sind zudem schwer zu bedienen (Jarzabek & Huang 1998).

Vielversprechendere Ansätze gerade für Entwicklungsphasen mit unterschiedlichen Akteuren, liefern die am Markt verfügbaren Werkzeuge iRise Suite<sup>1</sup>, Axure Pro<sup>2</sup>, Lucid Spec<sup>3</sup> oder Sofeainc Profesy<sup>4</sup>. Die Bedienung orientiert sich in weiten Teilen an Office Anwendungen, so dass die meisten Prozessbeteiligten mit ihnen sehr einfach UI Prototypen erzeugen können. Die gegenseitige Verknüpfung, z.B. einzelner Webseiten, zu einem Szenario ist mit visuellen Hilfsmitteln möglich. Die Werkzeuge nutzen jedoch Formate, die sich zur Weiterentwicklung kaum eignen. Gerade wenn komplexe Storyboards erzeugt worden sind, werden die Anwendungen schnell unübersichtlich und schnelle Änderungen sind kaum mehr möglich (Memmel et al. 2007b).

Mit einer domänenspezifischen Sprache (DSL) bestehen vergleichsweise mehr Möglichkeiten, visuelle Spezifikationen aus genau auf die Problemdomäne zugeschnittenen Ausdrucksmitteln zu erstellen (Memmel et al. 2007a). Durch die intelligente Verknüpfung geeigneter Werkzeuge können Prozessketten entstehen, die die Expertise unterschiedlicher Akteure optimal adressieren. Auch MAXpro stellt ein domänenspezifisches Werkzeug dar. Grundsätzlich kann MAXpro zur Spezifikation von Desktop- und Web-Anwendungen verwendet werden. Die Anwendungsdomäne ist bei DC zunächst der Bereich von aufwendigen Webanwendungen (Kapitel 2.3). Im Kontext von komplexen Informationsräumen, wie zum Beispiel der Webseite von Mercedes Benz (MB), ist das Ziel von MAXpro insbesondere eine leichte Erzeugung und Anpassung von hochgradig vernetzten Strukturen. MAXpro positioniert sich daher in Distanz zu Werkzeugen wie iRise und versucht ein UI

<sup>1</sup> [www.irise.com](http://www.irise.com)

<sup>2</sup> [www.axure.com](http://www.axure.com)

<sup>3</sup> [www.elegancetech.com](http://www.elegancetech.com)

<sup>4</sup> [www.sofeainc.com](http://www.sofeainc.com)

durch geeignete, abstrakte Modelle ohne GUI-Editor zu erzeugen.

## 2.2 Architektur

MAXpro verwendet eine DSL auf Basis von XML und unterscheidet drei Modellierungsbereiche. Das erste in XML ausgedrückte Modell beschreibt die Informationsarchitektur. Das zweite Modell bestimmt das Navigationskonzept, während das dritte Modell schließlich die Inhalte beschreibt. Durch die Trennung von Inhalt, Navigation und Architektur kann ein Domänen-Experte sich zunächst auf die Modellierung der Inhalte konzentrieren. Umgekehrt kann die Architektur auch modelliert werden, ohne dass bereits genaue Inhalte bestimmt sind. Dadurch eignet sich MAXpro auch sehr zum Testen von komplizierten Navigationskonzepten. Durch die Angabe von Verbindungen zwischen den unterschiedlichen Bestandteilen werden die Modelle später zusammengefügt. Die dynamische Erzeugung des Prototyps hat zur Folge, dass Änderungen an einzelnen Elementen der Modelle an alle betroffenen Objekte generisch vererbt werden. Zur Reduktion der Komplexität können Objekte auch in kleinere Einheiten aufgeteilt werden. Je nach Aufbau der Modelle können mit MAXpro einfache hierarchische oder komplexe vernetzte Strukturen erzeugt werden.

Zur Transformation der in XML beschriebenen Modelle in einen ausführbaren Prototyp verwenden wir XSL(T), JAVA und JDOM. Im Bereich webbasierte Anwendungen können wir mit unterschiedlichen Generatoren neben HTML auch Prototypen in dynamischen Sprachen (z.B. JSP) erzeugen. Grundsätzlich können somit zum einen statische Prototypen generiert werden, die ohne weitere Hilfsmittel in jedem Webbrowser ausgeführt werden können. Zum anderen können dynamische Versionen für Änderungen direkt zur Laufzeit zu verwendet werden.

## 2.3 Anwendungsbeispiel

Während der Entwicklung der Webseite von MB haben wir MAXpro für das frühe Entwickeln von Prototypen verwendet (siehe Abbildung 1). Durch die Modellierung einer Navigationsstruktur, der Benennung von Navigationspunkten und der Verknüpfung zu Blindtexten entsteht bereits ein abstrakter Prototyp der späteren Webseite. Die frühe Simulationen dient nun dazu, Diskussionen mit beteiligten Akteuren während Workshops zu führen und Evaluationen mit Endbenutzern vorzunehmen. So werden z.B. optimale Navigationspfade bestimmt und die Position einzelner Webseiten im Informationsraum erörtert.



Abbildung 1: Abstrakter Prototyp

Zur Verfeinerung der Prototypen und zur Evaluierung ausgearbeiteter Navigationselemente und Layouts, kann der UE-Experte Templates erzeugen und diese mit detaillierten Inhalten anreichern (Abbildung 2).



Abbildung 2: Detaillierter Prototyp

Die so erzeugten Repräsentationen (Views) werden mit den anderen Modellen verknüpft. So entstehen per Knopfdruck visuell ausgereifte Simulationen, die zur weiteren Inspektion zur Verfügung stehen. Später werden die erzeugten Prototypen an den AN weitergegeben. Der Entwickler hat sofort einen Eindruck über Aufteilung, Benutzerführung und Navigation und interaktives Verhalten.

## 2.4 Projekterfahrung

Aus unseren Erfahrungen können wir Vor- und Nachteile von MAXpro im Speziellen und visuellen Spezifikationen im Allgemeinen ableiten.

### Positive Aspekte

- MAXpro wird von den Akteuren als Kommunikationswerkzeug akzeptiert
- Akteure können sich mit einem einzigen Werkzeug austauschen
- Iterationen zwischen Diskussion, Designentscheidung und prototypischer Umsetzung werden deutlich verkürzt
- Frühe Evaluationen vermeiden teure, späte Änderungen und Mehrdeutigkeiten der Spezifikation
- Die Trennung in unterschiedliche Modellierungsbereiche erlaubt eine getrennte Bearbeitung
- Unterschiedliche Generatoren erlauben flexible Ausgabeformate
- Die Modellierung in XML erlaubt die einfache Weiterleitung an den AN
- Durch den Aufbau von Prototyping Kompetenz wird der AG flexibler bei der Wahl des AN

### Negative Aspekte

- Akteure müssen abstraktes XML verstehen, um am UI Spezifikationsprozess teilzunehmen
- Ein fehlender graphischer Editor erschwert die Modellierung für nicht-technische Akteure, speziell in frühen Phasen der Spezifikation
- Typische Aktivitäten einer frühen Anforderungsermittlung, wie z.B. Zeichnungen, werden nicht als Modelle erfasst

- Durch die fehlende Integration von Kontextinformation werden Designentscheidungen direkt in der Simulation nicht sichtbar
- Evaluationsergebnisse (Feedback) können zur besseren Korrelation und Kommunikation nicht direkt im Prototypen integriert werden
- Die Vertragsgestaltung mit dem AN muss den Grad der visuellen Spezifikation berücksichtigen
- Die Verwendung von visuellen Spezifikationen erfordert eine Anpassung etablierter Vorgehensweisen

Die Zusammenarbeit mit dem AN kann sich zunächst als schwierig herausstellen. So ist eine visuelle Spezifikation nicht mit der ausimplementierten Gestaltung des UI vergleichbar. Es ist daher wichtig, dass der Detailgrad der Spezifikation festgelegt wird. Der Aufwand auf AG-Seite muss geringer sein, als die tatsächliche Umsetzung.

Damit Akteure Ihr Feedback direkt in den Prototypen integrieren können (Rashid et al. 2006), wollen wir MAXpro entsprechend weiterentwickeln. Dazu soll eine direkte Verbindung zwischen Feedback-Prototyp und Entwicklungsumgebung geschaffen werden. Entwickler sollen ohne zeitliche Verzögerung in der Lage sein, Ergebnisse von Benutzertests einsehen zu können. Auf diese Weise können auch bei *Remote Usability Tests* schnell Änderungen am UI Prototypen vorgenommen werden, um Iterationszyklen zu verkürzen.

Damit der Einsatz von Werkzeugen wie MAXpro optimal in etablierte Entwicklungsmethoden eingebettet werden kann, müssen vorherrschende Vorgehensmodelle angepasst werden (Kapitel 3.0). Außerdem wäre eine visuelle Schnittstelle von Vorteil, mit der die Modellierung vereinfacht werden kann. Zusätzlich soll es auch möglich sein, graphische Notationen wie einfache UML Modelle oder Task Models zu integrieren. Damit kann im Kontext der Betrachtung eines Prototyps auf zugrunde lie-

gende Informationen zurückgegriffen werden. Dies schafft im Hinblick auf die Gestaltung gebrauchstauglicher Systeme mehr Transparenz und notwendiges, interdisziplinäres Verständnis (Kapitel 4.0).

### 3.0 Vorgehensmodell

Visuelle Spezifikationen zu erstellen ist aufgrund der verschärften Rahmenbedingungen bei der Entwicklung auch durch eine Öffnung für agile Vorgehensweisen motiviert. Auch Anforderungs- und Spezifikationsprozesse laufen bei DC in der Regel zyklisch ab. In diesem Zusammenhang haben wir ein Vorgehensmodell namens *Agile Usage-Centered Software Lifecycle* (AUCSL) entwickelt, welches Methoden des UE (Mayhew 1999) mit solchen des Agile Modeling (AM, Ambler 2002) und *eXtreme Programming* (XP; Beck 1999) zusammenbringt (Gundelsweiler et al. 2004). Dabei wird eine iterative und inkrementelle Vorgehensweise zur Erzeugung visueller Spezifikationen und UI Prototypen beibehalten.

Der Einsatz von durch die DSL erstellten Modellen zur Erzeugung ausführbarer Simulationen ist mit Prinzipien und Praktiken von XP und AM<sup>5,6</sup> vereinbar. Grundlegende Maximen wie *Model With Purpose, Model With Others, Model To Understand, Model To Communicate* oder *Rapid Feedback* treffen auf visuelle Spezifikationen zu (Mommel et al. 2007a; Mommel et al. 2007b). Stark verteilte Teams (z.B. Offshore) oder das Fehlen von Kollaborations- und Designräumen machen die Anwendung papierbasierter Methoden (CRC-Cards, Whiteboard, Post-ist, Flip Chart) unmöglich. Die Verwendung von computergestützten Werkzeugen widerspricht

jedoch den Maximen von AM nicht, wenn unterschiedliche Sichten auf die Anforderungen, eine Darstellung alternativer Lösungen oder unterschiedliche Abstraktionsstufen hilfreich sind. Ein entsprechendes Werkzeug kann zudem eine agile Dokumentation erlauben und Forward- (Code Generierung) und Reverse-Engineering unterstützen.

In unserem AUCSL Vorgehensmodell haben wir bereits Modelle und graphische Notationen vorgesehen, die den Disziplinen SE und UE entstammen. Für eine vollständige Interdisziplinarität müssen wir diesen noch UML oder BPMN<sup>7</sup> Modelle des *Business Engineering* (BE) hinzufügen. Auch für einen Brückenschlag zu dieser Disziplin schafft AM Schnittstellen<sup>8</sup>.

Der *Rational Unified Process* (RUP, Kruchten 2003) integriert zahlreiche Business Modelle und ist daher eine geeignete Quelle zur Extraktion passender Artefakte. RUP selbst sieht per se keine speziellen Methoden für UE und UI Design vor. Alternativ zu möglichen anderen Erweiterung von RUP um UE (Göransson et al. 2003) erweitern wir daher unseren AUCSL mit den Modellen eines agilen RUP, der *Agile Unified Process* (AUP, Ambler 2002) genannt wird.

Während der RUP eher formal und von UML 2.0 bzw. der Erstellung von *Use Case* Modellen geprägt ist, versucht der AUP mit informaleren und einfacheren Modellen zu arbeiten, die von allen Akteuren verstanden werden. Wichtig ist ein ausgewogenes Verhältnis zwischen dem Grad an Formalität und dem Nutzen für kollaborative Spezifikationsprozesse. Die Anzahl der Modelle und Notationen soll dabei maximal reduziert werden. Dazu haben wir in Anlehnung an (Gundelsweiler et al. 2004) nach ähnlichen Ausdrucksmitteln der drei Diszipli-

<sup>6</sup> [www.agilemodeling.com/practices.htm](http://www.agilemodeling.com/practices.htm)

<sup>7</sup> [www.bpmn.org](http://www.bpmn.org)

<sup>8</sup> [www.agilemodeling.com/artifacts/](http://www.agilemodeling.com/artifacts/)

<sup>5</sup> [www.agilemodeling.com/principles.htm](http://www.agilemodeling.com/principles.htm)

nen UE, SE und BE gesucht und diese miteinander kombiniert. Der kleinste gemeinsame Nenner an tatsächlich erforderlichen Artefakten ist der Ausgangspunkt für eine agile, interdisziplinäre Werkzeugunterstützung (Tabelle 1). Dabei behalten wir eine starke Orientierung an UE Methoden bei, weshalb wir uns als Hybridversion jeweils für das UE Modell entscheiden, wenn dieses starke Ähnlichkeiten zu denen der anderen Disziplinen aufweist. Ansonsten integrieren wir mehrere Modelle. Des Weiteren schließen wir Modelle aus, die nicht für die Spezifikation, sondern der Implementierung des finalen Systems dienen und daher auf AG-Seite nicht notwendig sind. So reduzieren wir die Komplexität und vermeiden Modelle, die nicht unbedingt benötigt werden.

#### 4.0 Visuelle Spezifikation mit Zoomable User Interface

Basierend auf unseren Erfahrungen mit MAXpro haben wir eine erweiterte Entwicklungsumgebung konzipiert, die mit innovativen Navigations- und Visualisierungsmethoden den Brückenschlag der Disziplinen UE, SE und BE schafft. Dabei nutzen wir weiterhin XML als Austauschformt für die Spezifikation, addieren jedoch zur Modellierung eine graphische Benutzeroberfläche. Das Werkzeug ermöglicht es, mindestens die Modelle aus Tabelle 1 zu erzeugen und diese in einer verständlichen Beziehung zueinander darzustellen. So stehen die visuellen Prototypen stets in Beziehung zu den zugrunde liegenden Modellen.

In Anlehnung an das konzeptuelle Modelle der Anwendungen Denim (Newman 2003) und Damask (Lin & Landay 2002), sowie basierend auf unseren Erfahrungen mit Zoomable User Interfaces (ZUIs, Gundelsweiler et al. 2007), verwenden wir für das Spezifikationswerkzeug ebenfalls Zoom-Techniken.

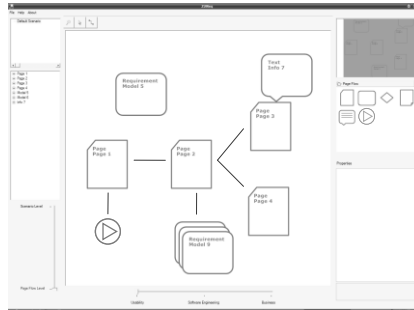


Abbildung 3: Startseite von INSPECTOR

Das Werkzeug namens INSPECTOR (*Interdisciplinary Specification Tool*) basiert auf einem zoombaren Canvas, auf welchem Modelle und deren Verbindungen dargestellt werden können. Durch die Operationen *Panning* und *Zooming* kann der Experte so eine Modelllandschaft erzeugen und explorieren. Zusätzlich stehen Mechanismen für einen *JumpZoom* in weiter entfernte Modellierungsbereiche zur Verfügung. Die Orientierung wird außerdem durch einen *Overview* unterstützt. Durch semantisches Zoomen geben Objekte bei näherer Betrachtung Detailinformationen Preis und bieten Werkzeuge zu deren Manipulation an. Modelle können zu Gruppen zusammengefasst werden. Dabei wird die Logik der zugrundeliegenden Modelle verwendet, so dass z.B. *Use Cases* zu *Use Case Diagrammen* gruppiert werden, die wiederum Teil eines *Storyboards* und schließlich eines UI Prototypen sind.

#### Vorteile INSPECTOR

- Integration von graphischen Notationen (Modellen)
- Animierte, zoombare Übergänge zwischen Modellen fördern Verständnis der Akteure
- Zusammenspiel von UI Repräsentation und zugrundeliegenden Modellen schaffen Transparenz
- Auf agiles Vorgehensmodell abgestimmte Auswahl von Modellen erleichtert ein strukturiertes, interdisziplinäres Vorgehen

- Die evolutionäre Entwicklung kann nachvollzogen werden
- Alle Modelle können serialisiert und dem AN übergeben werden

#### 5.0 Zusammenfassung

Die von uns vorgeschlagene Methode zur Gestaltung interaktiver Systeme verbindet Akteure aus SE, UE und BE unter dem Schirm agiler Prinzipien und Praktiken. Das als Basis vorgestellte AUCSL Vorgehensmodell kann auch für Bedingungen angepasst werden, in denen insbesondere das Design und damit die Emotionalität des UI im Vordergrund steht (Mommel et al. 2007c). Die von uns implementierten Entwicklungswerkzeuge bieten signifikante Mehrwerte in unseren Entwicklungsprozessen. Während MAXpro bereits im aktiv im Feld eingesetzt wird, arbeiten wir an weiteren Konzepten zur Verbesserung einer werkzeuggestützten, visuellen Spezifikation interaktiver Systeme. Unsere bisherigen Erfahrungen haben gezeigt, dass neben einer visuellen Schnittstelle auch Modelle der Anforderungsanalyse in ein adäquates Werkzeug integriert sein müssen. Insgesamt ist es für die Anwendung der von uns vorgeschlagenen Methoden und Werkzeuge erforderlich, dass sich neben etablierten Vorgehensmodellen auch die fachlichen Ausrichtungen der Experten ändern. So sehen wir vor, dass sich das Profil eines Usability- bzw. UI Design Experten für Herausforderungen moderner Entwicklungsprozesse öffnet. Schließlich ergibt sich daraus auch die Forderung nach mehr *generalisierten Spezialisten* in innovativen Entwicklungsteams (Ambler 2002).

#### 6.0 Literatur

- Ambler, W. S. (2002): Agile Modeling, New York: John Wiley & Sons
- Beck, K. (1999): Extreme Programming Explained, Addison-Wesley

Capgemini (2007): IT-Trends 2007. Online: <http://www.de.capgemini.com>

Göransson, B., Lif, M., Gulliksen, J. (2003): Usability Design - Extending Rational Unified Process with a New Discipline. In Interactive Systems. Design, Specification, and Verification, Volume 2844/2003, Springer Berlin/Heidelberg

Gundelsweiler, F., Memmel, T., Reiterer, H. (2004): Agile Usability Engineering. R. Keil-Slawik, H. Selke, G. Szwillus (Hrsg.): Mensch & Computer 2004: Allgegenwärtige Interaktion, München: Oldenbourg Verlag, S. 33-42

Gundelsweiler, F., Memmel, T., Reiterer, H. (2007): ZUI Konzepte für Navigation und Suche in komplexen Informationsräumen. In i-com, Zeitschrift für interaktive und kooperative Medien, Oldenbourg Wissenschaftsverlag, Vol. 6, Nr. 1, S. 38-48

Jarzabek, S., Huang, R. (1998): The case for user-centered CASE tools, Communications 41(8), ACM Press, 1998, S. 93-99.

Kruchten, P. (2003): Der Rational Unified Process – Eine Einführung (3rd Edition). Addison Wesley

Lin, J., Landay, J. A. (2002): Damask: A Tool for Early-Stage Design and Prototyping of Multi-Device User Interfaces. In Proceedings of The 8th International Conference on Distributed Multimedia Systems (2002 Interna-

tional Workshop on Visual Computing), San Francisco, CA, September 26-28, 2002, S. 573-580

Mayhew, D. J. (1999): The usability engineering lifecycle - A Practicioners Handbook for User Interface Design, San Francisco: Morgan Kaufmann

Memmel T., Bock, C., Reiterer, H. (2007a): Model-driven prototyping for corporate software specification, In Proceedings of the Engineering Interactive Systems Conference (EIS 2007), Salamanca, Spain

Memmel T., Gundelsweiler, F., Reiterer, H. (2007b): Prototyping Corporate User Interfaces – Towards A Visual Specification Of Interactive Systems. In Proceedings of the IASTED-HCI 2007, Chamonix, France

Memmel, T., Gundelsweiler, F., Reiterer, H. (2007c): Agile Human-Centered Software Engineering. In Proceedings of the 21st BCS HCI Group conference, Lancaster, UK

Newman, M. W., Jason, J. L., Hong, I., Landay, J. A. (2003): DENIM: An Informal Web Site Design Tool Inspired by Observations of Practice. In Human-Computer Interaction, 18(3), S. 259-324.

Offergeld, M., Oed, R. (2006): Usability Engineering als Auftraggeberkompetenz Proceedings Usability Professionals 2006 Tim Bosenick, Marc Hassenzahl, Matthias

Müller-Prove, Matthias Peissner German Chapter der Usability Professionals Association e.V., Stuttgart

Potts, C., Takahashi, K., Anton, A.I. (1994): Inquiry Based Requirements Analysis. IEEE Software, Volume 11, Issue 2, S. 21-32.

Rashid, A., Meder, D., Wiesenberger, J., Behm, A. (2006): Visual Requirement Specification In End-User Participation. In First International Workshop on Multimedia Requirements Engineering, 14th IEEE International Requirements Engineering Conference (RE'06), Minnesota, USA

Rudd, J., Stern, K., Isensee, S. (1996): Low vs. high fidelity prototyping debate, Interactions, vol. 3(1), S. 76–85. ACM Press, New York

Standish Group, Chaos Reports (2003): <http://www.standishgroup.com>.

Zave, P., Jackson, M. (1997): Four Dark Corners of Requirements Engineering. ACM Transactions on Software Engineering and Methodology. Volume 6, Issue 1, 1997, 1-30

## 7.0 Danksagung

Wir danken den Studenten Herrn Florian Geyer, Herrn Johannes Rinn und Herrn Daniel König für Ihre Mitwirkung zum Thema visuelle Spezifikation.

UE Modell	SE Modell	BE Modell	Hybrid Modell
User Roles, Personas	User Story, Personas	Personas	Personas
User, Task, Information, Interaction Scenario	Usage Scenario	Business Rule, Business Vision	Interaction Scenario
Flow Chart	Activity Diagram, Sequence Diagram	Business Object Model (Data Flow Diagram)	Flow Chart
(Essential) Use Case / Task Case	(Essential) Use Case	Business Use Case	Essential Use Case
Task Map	Use Case Diagram	Business Use Case Diagram	Task Map
(Use Case) Storyboard	UI Flow Diagram	Robustness Diagram	Storyboard
Low-Fi UI Prototype	Presentation Prototype	Mockup, Slides	Abstract UI Prototype
Hi-Fi UI Prototype	Functional Prototype, Pilot System	Images	Detail UI Prototype

Tabelle 1: Ausgewählte Modelle aus UE, SE und BE bzw. RUP Prozessen für einen agilen, interdisziplinären Entwicklungsprozess mit detaillierten Prototypen als oberste Schicht einer visuellen Spezifikation