M. Herczeg & M. C. Kindsmüller (Hrsg.): Mensch & Computer 2008: Viel Mehr Interaktion.
München: Oldenbourg Verlag, 2008, S. 357-366

User Interface Entwicklung mit interaktiven Spezifikationen

Thomas Memmel, Harald Reiterer

Arbeitsgruppe Mensch-Computer Interaktion, Universität Konstanz

Zusammenfassung

User Interface (UI) Spezifikationsprozesse involvieren unterschiedliche Akteure mit jeweils eigenen Ausdrucksmitteln. Dadurch ergeben sich Herausforderungen bei der Umsetzung von Anforderungen in gutes UI Design. Durch einen Mangel an interdisziplinären und kollaborativen Methoden und Werkzeugen dominieren dabei vor allem textbasierte Spezifikationsdokumente. Diese reichen jedoch mangels Interaktivität nicht aus, um innovative und kreative Prozesse zu unterstützen. In diesem Beitrag stellen wir eine Spezifikationstechnik vor, mit der Benutzer-, Aufgaben- und Interaktionsmodelle mit unterschiedlich detailliertem UI Design verbunden werden. Dadurch entsteht eine erlebbare UI Simulation, die im Vergleich zu interaktiven UI Prototypen zusätzlich den visuellen Drill-Down zu Artefakten der Anforderungsermittlung erlaubt. Das Resultat bezeichnen wir als interaktive UI Spezifikation, mit der eine höhere Transparenz und Nachvollziehbarkeit im Spezifikationsprozess möglich ist.

1 Einleitung

Die Modellierung, Gestaltung und Spezifikation von benutzerfreundlichen UIs ist heute bereits in vielen Industriezweigen ein wichtiger Wettbewerbsfaktor. Am Beispiel der deutschen Automobilindustrie haben wir erforscht, dass innovative Navigations- und Visualisierungskonzepte für Fahrerinformationssysteme eine ebenso wichtige Rolle spielen, wie die einfache Bedienbarkeit von Anwendungen im digitalen Vertriebskanal (Gundelsweiler et al. 2007). Ein strukturiertes und ineinander übergehendes Usability Engineering (UE) und Software Engineering (SE) ist daher unbedingt erforderlich (Metzker & Reiterer 2002). In Untersuchungen bei der Daimler AG und der Dr. Ing. h. c. F. Porsche AG haben wir jedoch herausgefunden, dass dies keineswegs der gelebten Praxis entspricht (Memmel et al. 2007). Der Automobilhersteller (Auftraggeber) spezifiziert ein UI in der Regel unter Mitwirkung von vielen unterschiedlichen Akteuren, darunter Domänen- bzw. Fachexperten sowie Marketingfachleute. Dabei werden überwiegend informale, textbasierte Office-Dokumente verwendet, da Office-Werkzeuge das im Unternehmen am weitesten verbreitete Kommunikationsmittel darstellen. Da nicht alle Akteure die gleichen Anwendungen favorisieren, erschweren viele

unterschiedliche Formate, wie z.B. Word, Excel oder PowerPoint, die Zusammenarbeit (Memmel et al. 2007). Medienbrüche verhindern eine ausreichende Transparenz, Nachvollziehbarkeit und Visualisierung von Anforderungen (Zave & Jackson 1997), sowie deren adäquate Übersetzung in ein gutes UI Design. Da textuelle Beschreibungen zudem häufig erheblichen Interpretationsspielraum lassen, wird das Erzeugen von adäquaten UI Prototypen erschwert. So fehlt oft die Möglichkeit, jederzeit zwischen abstrakten Artefakten und detaillierten UIs umschalten zu können. Die Auswirkungen von Designentscheidungen können dann nur mit Verzögerung identifiziert werden und Usability Probleme bleiben ebenso verborgen, wie Chancen für Innovation und Alleinstellung. Auch für den IT-Dienstleister (Auftragnehmer) ist es in der Regel unmöglich, das UI anhand textueller Spezifikationsdokumente in wenigen Iterationen richtig umzusetzen. Gerade das gewünschte interaktive Verhalten kann nur schwer aus abstrakten Beschreibungen extrahiert werden. Der Auftragnehmer verwendet zudem vollkommen andere Werkzeuge und formalere Methoden. Doch auch diese können nicht entlang der gesamten UI Wertschöpfungskette eingesetzt werden. Die aktive Beteiligung von Akteuren des Auftraggebers, die spezielle Terminologien oder Programmiersprachen nicht verstehen, würde verhindert. Formale Modellierungssprachen wie UML sind oft viel zu detailliert und können Benutzerinteraktionen mit dem UI schlecht abbilden (Ambler 2004). Innovatives UI Design und Gebrauchstauglichkeit bleiben dann hinter den Erfordernissen zurück (Campos & Nunes 2004).

In diesem Forschungsbeitrag stellen wir daher Methode und Werkzeug für einen semiformalen interdisziplinären UI Spezifikationsprozess vor, der durch die Auswahl geeigneter Modellierungssprachen und deren enger Verzahnung mit UI Design gekennzeichnet ist. Dadurch entstehen interaktive UI Spezifikationen, die gleichermaßen *Look* und *Feel* sowie die gesamte Design Rationale transportieren können. Dabei ist der von uns gewählte Ansatz trotz seiner Motivation aus der Automobilwirtschaft heraus so entwickelt, dass ein vielfältiger Einsatz möglich ist. Überall dort wo unterschiedliche Disziplinen gemeinsam für die Entwicklung verantwortlich sind, können interaktive UI Spezifikationen einen Mehrwert im Designprozess darstellen. Im 2. Abschnitt nehmen wir auf ähnliche Forschungsarbeiten Bezug. Schließlich erläutern wir im 3. Abschnitt die methodische Grundlage für das im 4. Abschnitt vorgestellte Werkzeug namens INSPECTOR. Der 5. Abschnitt beinhaltet bisherige Evaluationsergebnisse. Der Beitrag endet mit einer Zusammenfassung.

2 Ähnliche Forschungsarbeiten

In (Memmel et al. 2007) haben wir eine Modell-getriebene UI Spezifikationsmethode vorgestellt, die in späteren Phasen des UI Entwicklungsprozess erfolgreich bei der Spezifikation von Grafik-, Informations- und Interaktionsdesign eingesetzt werden konnte. Jedoch waren wichtige Artefakte aus frühen Phasen der UI Entwicklung (z.B. Benutzer- oder Aufgabenmodell) nicht integriert. Mit INSPECTOR stellen daher ein Methoden- und Werkzeugkonzept vor, welches in allen Phasen des UI Spezifikationsprozesses für Modellierung, Design und Simulation eingesetzt werden kann. Dabei setzen wir auf eine Modell-basierte Vorgehensweise, die nicht unmittelbar auf die Erzeugung von Code aus den Modellen fokussiert ist, sondern vielmehr auf die Unterstützung von kreativen Prozessen. Auch Constantine und

Campos (2005) haben mit den Werkzeugen CanonSketch und TaskSketch eine Modell-basierte Unterstützung von UI-Entwicklungsprozessen entwickelt. CanonSketch verwendet abstrakte Prototypen und eine vereinfachte UML Notation als Zwischenschicht zu einer Externalisierung des Designs als HTML Prototyp. TaskSketch ist ein Modellierungswerkzeug und bietet eine Kombination aus (Essential) Use-Case- bzw. Task-Case-Notationen und Aktivitätsdiagrammen. Alle Modelle lassen sich miteinander verbinden, so dass aus verknüpften Use-Cases ein Usage-Storyboard entsteht. Mit INSPECTOR bieten wir mehr Möglichkeiten zum UI-Prototyping und mehr Modellierungssprachen, die an einfache Notationen aus dem Agile Modeling (AM; Ambler 2004) angelehnt sind. Anknüpfend an das Werkzeug DAMASK (Lin & Landay 2002) und unsere eigenen Erfahrungen mit Zoom-basierten UIs (Gundelsweiler et al. 2007), verwenden wir ein Zoomable User Interface (ZUI), um unterschiedliche Detailstufen von Modellen und UI Design zu visualisieren.

3 Interdisziplinäre UI Modellierung

Es zählt zu den wichtigsten und häufigsten Aufgaben im UE, Bedürfnisse und Anforderungen in gebrauchstaugliche UIs zu übersetzen. Um schließlich ein UI spezifizieren zu können ist es wichtig, konzeptuelle Ideen zu externalisieren. Ein häufiger Wechsel zwischen abstrakten Beschreibungen und Designvisionen prägt dabei den UE Prozess. Dazu werden Konzepte aus dem Problem- in den Lösungsraum transferiert und evaluiert. Auch der umgekehrte Weg ist im UE wichtig: kann ein UE Experte nicht auf eine profunde Design Rationale zurückgreifen, werden Lösungen oft in Frage gestellt und verworfen. Daher müssen getroffene Designentscheidungen transparent und nachvollziehbar sein. Konzepte, die in die UI Spezifikation einfließen, müssen von allen Akteuren verstanden und auch von ihnen modelliert werden können. In unserer Forschung haben wir herausgefunden, dass UE Aktivitäten im UI Spezifikationsprozess durch eine bedachte Auswahl geeigneter und verbundener Modellierungssprachen und UI Designkonzepte unterstützt werden müssen. Dabei müssen die Fähigkeiten und Gewohnheiten der beteiligten Akteure berücksichtigt werden, aber dennoch eine Ablösung textbasierter Spezifikationsdokumente stattfinden. Dies kann durch Artefakte erreicht werden, die eine visuelle (graphische) Aufbereitung von Benutzerbedürfnissen und Nutzungsanforderungen erlauben. Die Identifikation geeigneter Modellierungssprachen ist mit Forschungsfragen zur Überbrückung der Unterschiede zwischen SE und UE verwandt. Beide Disziplinen finden sich auch in von uns untersuchten UI Spezifikationsprozessen wieder und sind bezüglich der verwendeten Ausdrucksmittel und der fachlichen Ausrichtung der Akteure recht unterschiedlich. Softwareentwickler sind für die Implementierung verantwortlich, während UE Experten in derartigen Konstellationen in der Regel eine vermittelnde Position einnehmen und die UI Qualität garantieren müssen. Wir erweitern die interdisziplinäre Betrachtungsweise zusätzlich um die Disziplin der Geschäftsprozessmodellierung (engl. Business Process Modelling, BPM). Akteure aller drei Disziplinen müssen gut zusammenarbeiten, damit benutzer- und aufgabengerechte UIs entwickelt werden können.

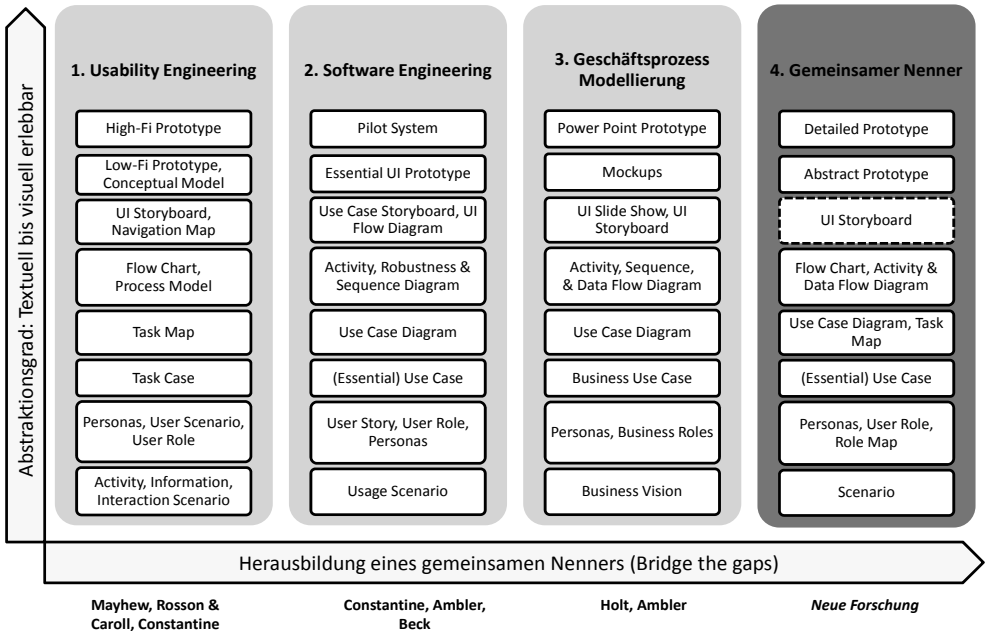

Abbildung 1: Bestimmung eines gemeinsamen Nenners in der interdisziplinären UI Modellierung und Spezifikation

Um für alle drei Disziplinen einen gemeinsamen Nenner zu definieren, haben wir zunächst Modelle und Designmethoden des UE betrachtet (s. Abb. 1, links). Anschließend haben wir diese anhand ihres Grads der visuellen Externalisierung kategorisiert. Das Ergebnis ist eine hierarchische Ordnung, beginnend mit textbasierten Artefakten (z.B. Szenarien), bis hin zu detaillierten Spezifikationsprototypen. Dazwischen bestimmen wir eine Auswahl von Benutzer-, Aufgaben- und Interaktionsmodellen aus dem UE (z.B. Rosson & Carroll 2002; Constantine & Lockwood 1999), die die Übersetzung von Erfordernissen in Anforderungen und schließlich in das UI Design unterstützt. Wir haben dabei agile UE Modelle (Gundelsweiler et al. 2004) priorisiert, da diese den Brückenschlag zu agilen Vorgehensweisen des SE und BPM (Ambler 2004; Holt 2005) erleichtern (s. Abb. 1). Agile Methoden ermöglichen bei gleichzeitiger Erhöhung der Kommunikation (AM: Model to communicate) und Zusammenarbeit (AM: Model with others; Active stakeholder participation) eine Vereinfachung und Deformalisierung, die für die von uns vorgefundenen Prozesslandschaften im Gegensatz zu formalen Entwicklungskontexten unbedingt erforderlich ist (AM: Apply the right artifacts). Durch eine Auswahl geeigneter Modelle eignen sich agile Methoden dennoch dazu, Anforderungen und UI Design zu spezifizieren. Dazu fokussiert sich die agile (UI) Entwicklung auf das Wesentliche, fordert Feedback und lässt dieses in Entscheidungen einfließen.

4 Interaktive UI Spezifikationen mit INSPECTOR

Das von uns entwickelte experimentelle Werkzeug INSPECTOR unterstützt die Akteure bei der agilen Benutzer-, Aufgaben- und Interaktionsmodellierung (s. Abschnitt 3) und der darauf basierenden Erzeugung einer interaktiven UI Spezifikation. Interaktiv bedeutet in diesem Zusammenhang, dass im Unterschied zu textbasierten Spezifikationsdokumenten einerseits und reinen UI Prototypen andererseits, eine erlebbare UI Spezifikation entsteht. Diese integriert die dem UI Design zugrundeliegenden Modelle sowie auch frühe und abstrakte Designstufen, um alle zum Verständnis der Spezifikation notwendigen Informationen bereitzustellen. Durch die Verwendung eines Zoom-Konzepts werden alle Artefakte zu einer interaktiven UI Spezifikation zusammengefügt. Dabei verwendet INSPECTOR geometrisches Zooming zur Vergrößerung von Objekten und semantisches Zooming für eine Veränderung der Darstellung und Informationsgranularität je nach Zoom-Faktor (Ware 2004). Automatisches und animiertes Zooming hilft neben einer Overview-Komponente (Abb. 3) dabei, die Topologie des Informationsraums zu verstehen. Auch Verknüpfungen zwischen Artefakten werden durch eine zielgerichtete Zoom-Operation traversiert. Dabei steuert das ZUI durch automatisches *Panning* und *Zooming* zum gewünschten Artefakt. Der Sprung zu anderen Artefakten kann aber auch durch ein ins ZUI integriertes Navigationsmenü (Pook 2001) ausgelöst werden. Durch das ZUI-Paradigma ermöglicht INSPECTOR so insgesamt eine visuelle Erlebbarkeit des UI Spezifikationsraums. Wie im Folgenden näher erläutert, wird dieser zunächst aus Modellen und UI Design aufgebaut. Die interaktive UI Spezifikation ermöglicht später den visuellen Drill-Down von der Simulation des UI zu den entstandenen Modellen.

Für die Beschreibung von Erfordernissen sind Szenarien ein interdisziplinär eingesetztes Ausdrucksmittel (Barbosa & Paula 2003). Szenarien eignen sich zur Erfassung von Systemeigenschaften, Handlungssträngen und Problemstellungen (SE), sowie zum Beschreiben von Endbenutzern und der damit verbundenen Vision von der Interaktion mit dem UI (UE). Geschäftsprozessmodellierer (BPM) verwenden Szenarien für die Beschreibung von Business Visionen und Business Rules (Abb. 1). Entsprechend der von uns definierten Modellierungshierarchie verwenden wir Szenarien als Einstieg in INSPECTOR. Da in der Regel mehrere, auch verbundene, Szenarien ausgearbeitet werden, bietet INSPECTOR eine „Scenario Map“. Dazu werden auf der Zeichenfläche zwei unterschiedliche graphische Elemente arrangiert. Die Sprechblasen dienen dabei als Platzhalter zur Erfassung der Szenario-Texte (optional mit vorgegebenen Templates) mit einem Texteditor (s. Abb. 2, rechts). Zusätzlich können auch vorhandene Dokumente (z.B. PDF), die etwa ein Problem-Szenario beschreiben, direkt integriert werden. Die Szenario-Formen (s. Abb. 2) dienen schließlich dazu, dass der Benutzer über einen Doppelklick auf die Storyboard-Ebene zoomen kann (s. Abb. 3). An der aus dem UE bekannten Storyboard-Notation (Beyer & Holtzblatt 1998, Ambler 2004) halten wir dabei fest. Aufgabe des UI-Storyboard ist es einerseits, Modelle und UI-Design in Relation zueinander zu setzen, sowie andererseits bereits den UI-Dialogfluss abzubilden.

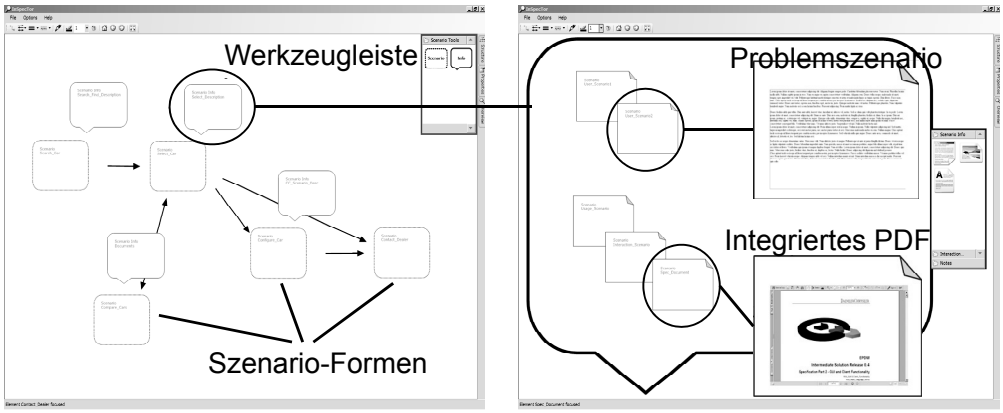

Abbildung 2: Scenario Map (links) und Informationsmodell für Problemszenario (rechts)

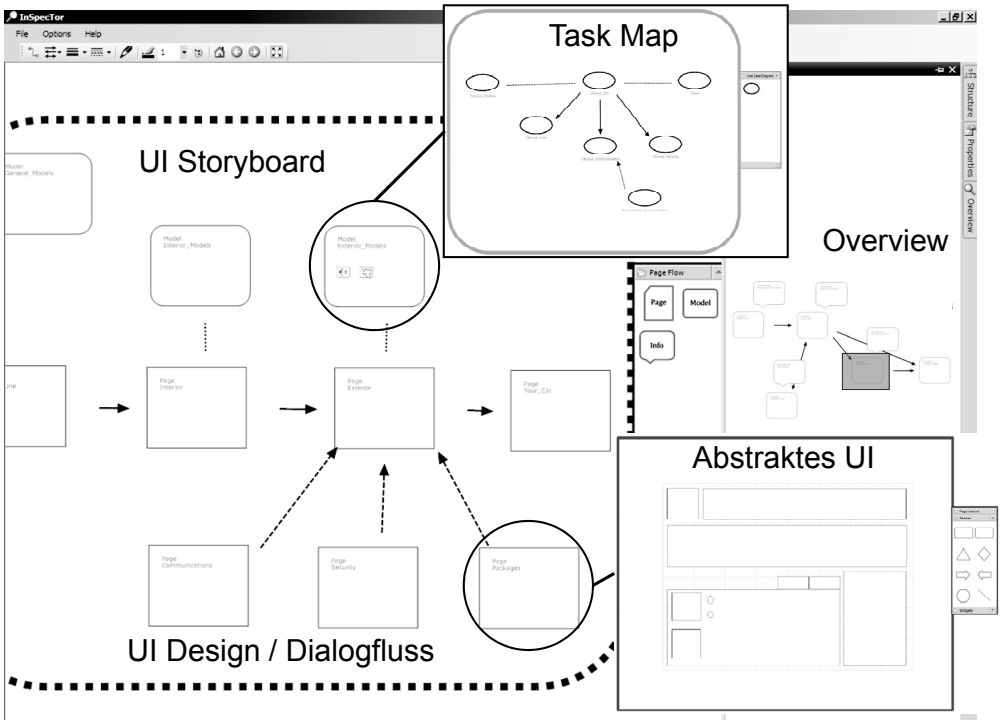

Abbildung 3: UI-Storyboard mit Platzhaltern für UI-Design und Modelle

Für Benutzermodellierung eignen sich User Profiles, User Scenarios (Rosson & Carroll 2002), Role Models und Role Maps (Constantine & Lockwood 1999) sowie Personas (Beyer & Holtzblatt 1998). Letztere dienen als gemeinsamer Nenner, da sie auch im SE und BPM bekannt sind. Durch den Zoom in einen Modellbereich (s. Abb. 3), kann der Akteur mit

deren Modellierung beginnen (s. Abb. 4, links). Trotz ihrer unterschiedlichen disziplinären Herkunft, können auch Task Cases (UE) und Business Use Cases (BPM) in einer für alle Disziplinen kompatiblen Form dargestellt werden: Essential Use Cases (UE, SE) eignen sich sehr gut, um die Interaktion des Benutzers mit dem UI zu beschreiben (Constantine & Lockwood 1999). Use Case Diagramme (SE, BPM) sind Use Case bzw. Task Maps (UE) ähnlich (Constantine & Lockwood 1999) und eignen sich daher für die interdisziplinäre Modellierung (s. Abb. 4, Mitte). Von diesen können Verbindungen zu den Personas (s. Abb. 4, links), Essential Use Case Notationen oder Aktivitätsdiagramme (nach Ambler 2004; Abb. 4, rechts) hergestellt werden. Letztere sind das objektorientierte Äquivalent zu Flow Charts (UE) und im Bereich der Prozessmodellierung (BPM) ebenso bekannt, wie in der Interaktionsmodellierung eines einzelnen Use Case Scenarios (SE). Data Flow Diagrams (nach Ambler 2004) integrieren wir zur Modellierung von Datenflüssen zwischen UI und Back-End System, die Einfluss auf das UI Design haben.

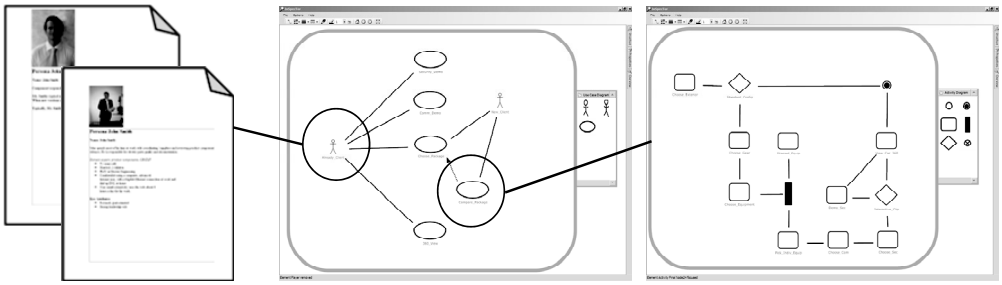

Abbildung 4: Personas (links), Use Case Diagramm (Mitte) und Aktivitätsdiagramme (nach Ambler 2004; rechts)

Für frühe Phasen des UI Entwurfs sind abstrakte Prototyping Techniken in allen drei Disziplinen (Sutcliffe 2005, Blomkvist 2005) als ein verbreitetes Mittel für konzeptuelles Design und Evaluation bekannt. Abstrakte Prototypen helfen, die im Spezifikationsprozess erforderliche Transparenz und Nachvollziehbarkeit bereits in frühen Phasen herzustellen. Dazu stellen wir, angelehnt an die Gewohnheiten der Akteure, einfache Formen (vgl. PowerPoint) zur Verfügung und auch Freihandzeichnen ist möglich. Um dem Anspruch einer visuellen UI Spezifikation gerecht zu werden, reichen sie jedoch nicht aus. Erst mit detaillierten und verlinkten UI Designs (s. Abb. 5, links) entsteht schließlich die interaktive UI Spezifikation, die den visuellen Drill-Down zu den Modellen ermöglicht. INSPECTOR stellt zu diesem Zweck typische UI Elemente bereit und erlaubt darüber hinaus auch die Integration von ActiveX Komponenten. Obwohl wir keinen Modell-getriebenen Ansatz verfolgen, haben wir Exportmöglichkeiten der gestalteten UIs nach XAML und usiXML integriert. Während XAML vor allem für eine Simulation des UI im Browser und eine Implementierung in Microsoft Werkzeugen beim Auftraggeber dient, ermöglicht usiXML (www.usixml.org) die Integration von INSPECTOR in eine Werkzeugkette zur UI Modellierung. Für ein Review der UI Spezifikation, stellt INSPECTOR auch eine Annotationsfunktion bereit, mit der Feedback direkt im ZUI integriert werden kann.

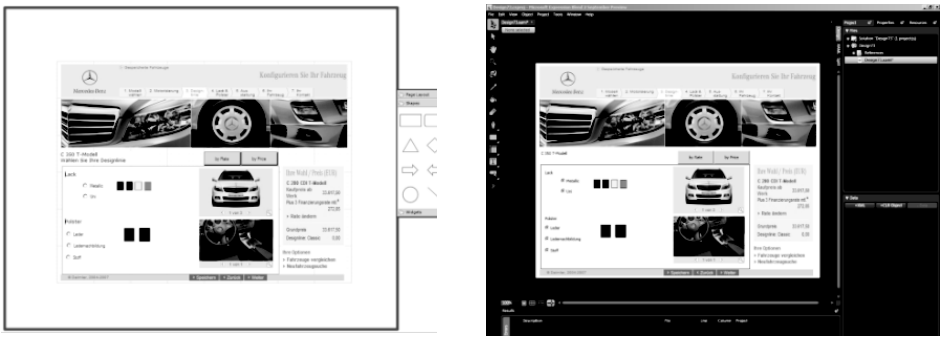

Abbildung 5: Detaillierter Spezifikationsprototyp in INSPECTOR (links) und die XAML Darstellung (rechts)

5 Evaluation

Wir haben damit begonnen, UE und SE Experten (n = 12) aus der Industrie (z.B. Daimler AG, ProContext GmbH) zu Vollständigkeit, Einsatzfähigkeit und Gebrauchstauglichkeit von INSPECTOR zu befragen. Wir haben bisher n = 6 der dazu eingesetzten Fragebögen zurückerhalten. Bisher haben alle Experten angegeben, dass die vorhandene Prozesslandschaft durch INSPECTOR verbessert werden kann. Dies ergibt sich für die Experten insbesondere aus der Kombination von informaler Modellierung und einfach zu bedienenden Designwerkzeugen (ø 4,8 Punkte; Skala 1-5 Punkte). Als innovative Unterstützung wird dabei deren Zoom-basierte Verbindung bewertet, worin eine Erhöhung von Transparenz und Nachvollziehbarkeit erkannt wird. In unserer Befragung kamen jedoch auch konzeptuelle Schwachstellen auf, so dass das Potential für den Spezifikationsprozess insgesamt noch niedrig bewertet wurde (ø 3,6 Punkte). Zum Beispiel stört ein zu häufiges Wechseln zwischen Modellen und Design den Arbeitsfluss. Wir integrieren daher eine kontextuelle Ebene (Pook 2001) zur zeitweisen Einblendung von Artefakten, die zum Fokus einen etwa durch Verlinkung hergestellten Bezug haben. Weitere Probleme betreffen vor allem die Interaktion mit dem ZUI und gleichen denen, die wir in einer zweiten Usability Studie herausgefunden haben.

In einer Tagebuch-Studie haben wir INSPECTOR über einen Zeitraum von 3 Wochen als UI Spezifikationswerkzeug in der Lehre eingesetzt. Insgesamt 8 Informatikstudenten haben auf einer wöchentlichen Basis ihre Projektarbeit und Erfahrungen mit INSPECTOR in einem Nutzertagebuch dokumentiert, darunter: (1) Erzeugte Modelle, (2) zusätzlich verwendete Werkzeuge, (3) festgestellte Probleme, (4) die erlebte User Experience und (5) eine generelle Bewertung. Wir haben uns bewusst für diese Art von Studie entschieden, um die Ergebnisse und deren Veränderung über einen längeren Zeitraum zu evaluieren. Im Vergleich zu einer Laborstudie konnten wir auf diese Weise auch Probleme identifizieren, die erst durch einen intensiven Einsatz von INSPECTOR attestiert werden konnten. So erkannten wir, dass nach einem mehrstündigen Einsatz des Werkzeugs Schwierigkeiten mit der Stabilität des ZUI und der Konsistenz der XML-basierten Projektdateien entstanden. Darüber hinaus hatten einige Teilnehmer Probleme mit dem richtigen Einsatz der Modelle und Designmethoden. Durch eine Lösung der Probleme und die Bereitstellung neuer Funktionalitäten, konnten die Stu-

denten zunehmend besser mit Modellen und UI Design Methoden umgehen. Während die Teilnehmer INSPECTOR zu Beginn noch mit durchschnittlich 1,75 Punkten (s = 0,46; auf einer 5-Punkte-Skala) bewerteten, erreichte die Bewertung so in der zweiten Woche einen Durchschnitt von 3,0 Punkten (s = 0,00) und am Ende der Studie schließlich 4,25 Punkte (s = 0,46). Die Varianzanalyse zeigte einen signifikanten Haupt-Effekt für die wöchentlichen Bewertungen ($F(2,14) = 105{,}00$, $p<0{,}001$), Auch die Unterschiede zwischen den einzelnen Wochen waren statistisch signifikant (Woche 1/Woche 2: $F(1,7)=58{,}33$; $p<0{,}001$; Woche 2/Woche 3: $F(1,7)=58{,}33$; $p<0{,}001$).

Zusammenfassung

Die Gestaltung gebrauchstauglicher und innovativer UIs beginnt mit dem Einsatz eines problemadäquaten Spezifikationswerkzeugs. Dieses muss die Zusammenarbeit von Akteuren aus unterschiedlichen Disziplinen vereinfachen und ein gemeinsames Verständnis des Problemraums ermöglichen. Benutzer-, Aufgaben- und Interaktionsmodelle müssen in einer einheitlichen und von allen Akteuren verstandenen Notation erfasst werden. Auch das Gestalten von Designvisionen und detaillierten UI Simulationen muss jederzeit leicht möglich sein. Mit INSPECTOR verbinden wir Modelle und UI Design mit einem ZUI Konzept zu einer interaktiven UI Spezifikation. Diese erlaubt im Gegensatz zu textbasierten Spezifikationsdokumenten ein visuelles Erleben der Design Rationale, wodurch die Transparenz und Nachvollziehbarkeit in einem kreativen und innovationsorientiertem Prozess erhöht wird.

Danksagung

Wir danken Florian Geyer und Johannes Rinn für die Mitwirkung am Projekt INSPECTOR.

Literaturverzeichnis

Ambler, S. W. (2004). *The Object Primer – Agile Model-Driven Development with UML 2*. Cambridge, MA: Cambridge University Press.

Barbosa, S. D. J. & Paula, M.G. (2003). Interaction Modelling as a Binding Thread in the Software Development Process, In *Proc. of the workshop on bridging the gaps between software engineering and human-computer interaction*, Oregon, USA.

Beyer, H. & Holtzblatt, K. (1998). *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann.

Blomkvist, S. (2005). Towards a model for bridging agile development and user-centered design. In: Seffah, A., Gulliksen, J., Desmarais, M. C. (Hrsg), *Human-centered software engineering – integrating usability in the development process*. Springer, S. 219-244.

Campos, P. & Nunes, N. (2004). Canonsketch: a User-Centered Tool for Canonical Abstract Prototyping. In *Proc. of 11th International Workshop on Design, Specification and Verification of Interactive Systems*. Springer, S. 146-163.

Constantine, L. & Campos, P. (2005). CanonSketch and TaskSketch: Innovative Modeling Tools for Usage-Centered Software Design. In *OOPSLA'05 Demonstrations*, San Diego, CA

Constantine, L. L. & Lockwood, L. A. D. (1999). *Software for Use: A Practical Guide to Models and Methods of Usage-Centered Design*. Addison-Wesley.

Gundelsweiler, F., Memmel, T. & Reiterer H. (2004). Agile Usability Engineering . In: Keil-Slawik, R., Selke, H. & Szwillus, G.: *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. *i-com, Zeitschrift für interaktive und kooperative Medien*, S. 38-48.

Holt, J. (2005). *A Pragmatic Guide To Business Process Modelling*. British Computer Society, UK.

Lin, J. & Landay, J. A. (2002). Damask. A Tool for Early-Stage Design and Prototyping of Multi-Device User Interfaces. In *Proc. of the 8th International Conference on Distributed Multimedia Systems, San Francisco*, S. 573-580.

Memmel, T., Bock, C. & Reiterer, H. (2007). Model-driven prototyping for corporate software specification. In *Proc. of the Engineering Interactive Systems Conference (EIS 2007), Salamanca, Spain*. Veröffentlichung steht aus (siehe http://www.se-hci.org/).

Metzker, E. & Reiterer, H. (2002). Evidence-Based Usability Engineering. In *Proc. of the CADUI 2002*, S. 323-336.

Pook, S. (2001). Interaction and Context in Zoomable User Interfaces. École Nationale Supérieure des Télécommunications. Paris, France.

Rosson, M. B. & Carroll, J. M. (2002). *Usability Engineering: scenario-based development of human computer interaction*. San Francisco, CA: Morgan Kaufmann.

Sutcliffe, A. G. (2005). Convergence or competition between software engineering and human computer interaction. In Seffah, A., Gulliksen, J., Desmarais, M. C. (Hrsg.): Human-centered software engineering – integrating usability in the development process, Springer, S. 71-84.

Ware, C. (2004). *Information Visualization: Perception for Design*. San Francisco, CA: Morgan Kaufmann.

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

Kontaktinformationen

Universität Konstanz, Arbeitsgruppe Mensch-Computer Interaktion
Universitätsstrasse 10, Box 73
D-78457 Konstanz