

Protoreto: Interaktive und automatisch auswertbare Papierprototypen

Daniel Herding, Ulrik Schroeder, Mostafa Akbari

Lehr- und Forschungsgebiet Informatik 9 (Computerunterstütztes Lernen),
RWTH Aachen University

Zusammenfassung

Prototyping ist bei der Entwicklung von Anwendungen mit grafischen Oberflächen ein wesentliches methodisches Werkzeug. Mit dem Prototyping-Werkzeug *Protoreto* und dem Capture-and-Replay-Tool *Jacareto* lassen sich GUI-Prototypen auf unterschiedlichen Abstraktionsniveaus schnell und einfach erstellen und leicht variieren. In Benutzerstudien lassen sich die Interaktionen der Versuchsperson mit dem Prototyp erfassen und später wiedergeben. *Jacareto* strukturiert die erhobenen Interaktionsdaten automatisch, z.B. für eine Auswertung in Statistiksoftware. Weiterhin beschreiben wir einige Features von *Protoreto*, mit denen sich Vorteile von Papierprototypen auf den Computer übertragen lassen.

1 Einleitung

Ziel des Prototyping in einem benutzerorientierten Entwurf (user-centered design) ist es, die Wahrscheinlichkeit, dass die Benutzenden das jeweilige Tool in ihrem Alltag mit Zufriedenheit einsetzen, zu erhöhen. Zu diesem Zweck werden die Funktionalität und die Bedienschnittstelle im Entwicklungsprozess in mehreren Iterationen mittels Benutzerstudien erprobt.

Im Folgenden stellen wir das Prototyping-Tool *Protoreto* vor, das für den benutzerzentrierten Entwurfsprozess entwickelt wurde, um die Kluft zwischen *Papier-* und *Displayprototypen* zu schließen. *Protoreto* unterstützt die schnelle Entwicklung von Prototypen, die sich leicht variieren und schrittweise konkretisieren lassen. Außerdem erlaubt es, die Interaktionen der Benutzenden mittels des Capture-and-Replay-Tools *Jacareto* aufzuzeichnen und automatisch strukturierte Daten zu erheben. *Protoreto* wird derzeit in verschiedenen wissenschaftlichen Disziplinen genutzt.

In Abschnitt 2 werden wir verschiedene Arten des Prototyping beschreiben. In Abschnitt 3 erläutern wir die Motivation für die Entwicklung von *Protoreto*. In Abschnitt 4 und 5 stellen

wir *Jacareto* und *Protoreto* vor. Ein Anwendungsbeispiel folgt in Abschnitt 6. Im Schlussteil geben wir einen Ausblick auf aktuelle Projekte zur Weiterentwicklung von *Protoreto*.

2 Prototyping

Mit der Schwierigkeit, den benutzerorientierten Entwurf in den Software-Engineering-Prozess einzugliedern, geht das Problem einher, dass der Begriff „Prototyping“ nicht eindeutig definiert ist.

“An analysis of publications shows that there is a lack of consensus in software engineering on the meaning of the terms prototype and prototyping. The same is true of the prototyping-oriented development approach.” (Bischofberger & Pomberger 1992, 15)

Daran hat sich seit 1992 nichts geändert. In der Literatur und in der Praxis werden der Prozess des Prototyping sowie die verschiedenen Arten von Prototypen unterschiedlich bezeichnet und verstanden. Die Definition *“prototype. A preliminary type, form, or instance of a system that serves a model for later stages or for the final, complete version of the system.”* (IEEE standard glossary of software engineering terminology, 1990) lässt Spielraum für Interpretationen. Seit der Festlegung der Definition haben sich neue Techniken und Methoden des Prototyping entwickelt, besonders im Bereich der Mensch-Maschine-Interaktion. Aus dieser Vielfalt heraus werden das Prototyping und die dabei erstellten Prototypen im nächsten Abschnitt neu kategorisiert und zusammengefasst.

2.1 Prototyping-Prozesse

Prototyping kann auf zwei verschiedene Weisen durchgeführt werden (Sommerville 2003, 184ff.). Wegwerf-Prototypen (throw-away prototypes) werden insbesondere am Anfang eines Projekts erstellt. Die Untersuchung der Prototypen durch Benutzerstudien ist gerade in dieser Phase sehr wichtig, da auf diese Weise z.B. Fehler in der Anforderungsanalyse auffallen, noch bevor viel Entwicklungsarbeit geleistet wurde. Wegwerf-Prototypen werden schnell und möglichst günstig erstellt und dienen ausschließlich dem Testen der Usability von Interaktionen und Funktionalitäten. Exploratives Prototyping und Experimentelles Prototyping nach Floyd werden hier unter Wegwerf-Prototyping zusammengefasst (Ludewig & Lichter, 2007). Balzert (1998), Dahm (2006), Bischofberger & Pomberger (1992) und Kappel (2004) erwähnen Wegwerfprototypen, gehen aber nicht weiter auf diese ein. Low-, Medium- und High-Fidelity-Prototypen nach Van Duyne et al. (2003) und Snyder (2003) werden unter Wegwerf-Prototypen eingeordnet, da sie nach der Evaluation nicht weiterverarbeitet werden. In einigen anderen Quellen findet sich auch die Bezeichnung „low-cost prototype“.

Beim Evolutionären Prototyping wird hingegen derselbe Prototyp stets weiterentwickelt. Dieser ist eine in Funktionalität und Design eingeschränkte Version des zu entwickelnden Systems. Vom Prototyp ausgehend über ein Pilotsystem bis hin zum fertigen System wird im

Extremfall ein einziger Prototyp durchgehend weiterentwickelt. In den verschiedenen Evaluationsstufen werden als Testpersonen Benutzer, Experten und Entwickler eingebunden.

Funktionalität und Design lassen sich auf verschiedenen Ebenen erproben. Um mit einer klaren Struktur Prototypen zu bilden, sollte entschieden werden, ob ein Horizontaler Prototyp (viele Funktionen, die aber jeweils nicht vollständig nutzbar sind) oder ein Vertikaler Prototyp (wenige Funktionen, die jedoch detailliert umgesetzt werden) erstellt wird (Balzert, 1998; Dix, 2006). Des Weiteren können Unterscheidungen nach dem *Level of Visual Refinement*, der *Richness of Interactivity* und der *Richness of Data Model* getroffen werden (McCurdy et al., 2006).

2.2 Arten von Prototypen

Es gibt verschiedene Techniken und Technologien, mit denen sich Prototypen für den Entwurf und Test von Benutzerschnittstellen herstellen lassen. Grob kann man zwischen computerunterstützten und nicht-digitalen Prototypen unterscheiden.

Papierprototypen können mit Stift und Papier oder am Computer, beispielsweise mit PowerPoint, erstellt werden. Sie bilden die Interaktionsmöglichkeiten mit einem System ab, können in Benutzerstudien allerdings von der Versuchsperson nicht eigenständig bedient werden. Stattdessen präsentieren die Entwickler dem Probanden den Papierprototyp und simulieren die Aktionen des Systems, indem sie zum Beispiel durch das Aufkleben von Post-Its verschiedene Systemzustände darstellen (Snyder, 2003; Van Duyne et al., 2003; Bischofberger & Pomberger, 1992; Kappel, 2004).

Die Ergebnisse aus der Evaluation der Papierprototypen werden in späteren Iterationsphasen zur Entwicklung von *Displayprototypen* herangezogen. Darunter verstehen wir Prototypen, die auf Standard-Hardware ausgeführt, auf einem grafischen Display dargestellt und mit Eingabegeräten wie Tastatur, Maus oder Touchscreen bedient werden. Die Bezeichnung Displayprototyp wird hier neu eingeführt, da in der von uns betrachteten Literatur kein adäquater und einheitlich verwendeter Begriff existiert. Die Vorgehensweisen und Techniken von Displayprototypen sind unter anderem in Sommerville (2003), Dix (2006), Van Duyne et al. (2005), Dahm (2006), Ludewig & Lichter (2007) und Balzert (1998) benannt, werden aber jeweils anders zusammengefasst.

Zur Entwicklung von Displayprototypen lassen sich verschiedene Technologien einsetzen. So lassen sich mit GUI-Buildern, die Bestandteil gängiger Integrierter Entwicklungsumgebungen (IDEs) sind, Bedienoberflächen systemnah nachstellen. Alternativ können Auszeichnungs-, Beschreibungs- und Skriptsprachen wie HTML, CSS, JavaScript, TCL/TK verwendet werden, um Displayprototypen mit einfachen Interaktionsmöglichkeiten zu erstellen. Auch Anwendungen wie Adobe Flash, Microsoft PowerPoint oder Microsoft Visio können

zum Einsatz kommen. Schließlich existieren einige spezialisierte Prototyping-Tools wie z.B. DENIM (Lin et al. 2000), RapidRabb.it¹ und Axure².

3 Anforderungen und Stand der Technik

Aus unseren Erfahrungen haben wir Vorgehensweisen bei der Erstellung und Auswertung unserer Prototypen identifiziert, die durch vorhandene Werkzeuge nicht ausreichend unterstützt wurden.

- Um alternative Designvorschläge bewerten zu können, müssen mehrere Prototypen parallel entwickelt werden, von denen letztendlich nur einer im Endprodukt umgesetzt wird. Darum sollten sich der Aufwand und die zu investierenden Zeit für die Erstellung der Wegwerf-Prototypen in Grenzen halten.
- Um Aussagen über den fachlichen Nutzen und die Benutzerfreundlichkeit eines Systems machen zu können, müssen genaue Daten über das Benutzerverhalten erhoben werden. Ein Werkzeug sollte deshalb die Möglichkeit zur automatisierten Erfassung und Auswertung dieser Daten bieten.
- Damit der entwickelte Prototyp bei Bedarf von allen Mitgliedern des meist interdisziplinären Teams angepasst werden kann, sollte das Werkzeug benutzerfreundlich und für alle Mitglieder des Teams leicht zugänglich sein. Dazu gehört insbesondere die Möglichkeit, den Prototypen ohne bzw. mit nur geringen Programmierkenntnissen anpassen zu können.

Diese Anforderungen konnte kein von uns untersuchtes Tool erfüllen. Spezielle Prototyping-Tools wie Axure sowie generische Anwendungen wie Microsoft PowerPoint, Adobe Flash und das (nicht mehr weiterentwickelte) HyperCard sind zwar für die Erstellung von Wegwerf-Prototypen geeignet, bieten aber keine einfache Möglichkeit, Daten über die Interaktion der Benutzenden mit dem System zu sammeln (vgl. Wagner 1990, 83).

Dumas et al. schlagen vor, Videokameras auf den Bildschirm und auf die Eingabegeräte auszurichten; allerdings weisen sie selbst darauf hin, dass beim späteren Abspielen die Schwierigkeit besteht, die Videoaufzeichnungen für die Evaluation synchronisieren zu müssen (Dumas et al. 2003, 226). Zudem ist eine manuelle Transkription, Strukturierung und Auswertung nötig. Weitere Nachteile sind anfallende Anschaffungskosten, Aufwand für Auf- und Abbau der Geräte und hoher Speicherbedarf.

Alternativ gibt es die Möglichkeit, Screencast-basierte Werkzeuge wie beispielsweise Camtasia Studio oder Morae³ einzusetzen. Beispielsweise erstellt Morae ein statisches Video der

¹ RapidRabb.it ist ein kommerzielles Online-Werkzeug zum kollaborativen Rapid Prototyping. <http://www.rapidrabb.it>

² Axure ist eine kommerzielle Anwendung zur Erstellung von Wireframes, Prototypen und Spezifikationen. <http://www.axure.com/>

Interaktion und ergänzt es um Informationen über Tastatur- und Mausbenutzung (Göpel 2006). GUI-Ereignisse auf höherer Ebene (z.B. die Auswahl aus einem Dropdown-Menü) werden ebenso wenig aufgezeichnet wie anwendungsspezifische, semantische Informationen. Die Morae-Aufzeichnung muss also, ebenso wie die von Dumas et al. vorgeschlagenen Videoaufzeichnungen, nachträglich vollständig angesehen, manuell interpretiert und strukturiert werden.

Da kein Tool unsere Anforderungen an Prototyping und Datenerfassung erfüllte, war es unser Ziel, ein Prototyping-Toolkit zu entwickeln, mit dem Displayprototypen schnell und einfach erstellt werden können und das die Interaktionsdaten der Benutzenden automatisch aufzeichnen, strukturieren und wiedergeben kann.

4 Aufzeichnung und Analyse mit Jacareto

Um die Datenerfassung bei Benutzerstudien zu automatisieren, kann das *Jacareto*-Framework eingesetzt werden. Das Akronym Jacareto steht für Java Capture and Replay Tool (Schroeder, 2000; Spannagel, 2003; Spannagel et. al, 2005), eine Open-Source-Software, die an der TU Darmstadt, der PH Ludwigsburg und der RWTH Aachen entwickelt wurde. Mit Jacareto lässt sich die Interaktion eines Benutzers mit einer GUI-basierten Java-Anwendung aufzeichnen und wieder abspielen.

Bei der Aufzeichnung wird ausgenutzt, dass in Java sämtliche Ereignisse einer AWT- oder Swing-Oberfläche eine Event-Queue durchlaufen. Jacareto registriert sich selbst als Event-Listener und wird dadurch über alle Events informiert, die innerhalb der Java Virtual Maschine ausgelöst werden. Dies erlaubt die Interaktionsaufzeichnung bei allen Java-Anwendungen, ohne deren Quellcode zu verändern. Liegt der Quellcode vor, können optional anwendungsspezifische, semantische Events ausgelöst werden, die mithilfe eines Moduls von Jacareto ebenfalls erkannt und aufgezeichnet werden können. Die Jacareto-Aufzeichnung ist also sehr viel detaillierter und aussagekräftiger als z.B. bei Morae.

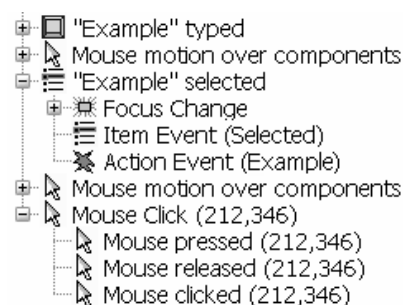


Abbildung 1: Strukturierung der Interaktions-Ereignisse in Jacareto

³ Camtasia Studio und Morae sind Produkte der Firma TechSmith. <http://www.techsmith.com/>

Sämtliche Ereignisse werden automatisch gruppiert und strukturiert. Mehrere aufeinanderfolgende Tastenanschläge werden zum Beispiel zu einer Tastensequenz gruppiert. Durch diese Gruppierung entsteht eine Baumstruktur, in der die Aufzeichnung übersichtlich dargestellt wird (siehe Abb. 1).

Die aufgezeichneten Datensätze lassen sich für die quantitative, statistische Auswertung exportieren. Zugleich ist aber auch eine qualitative Analyse möglich, da sich die Aufzeichnungen abspeichern und zu einem späteren Zeitpunkt wieder abspielen lassen. Die aufgezeichneten Aktionen können in Echtzeit oder im „Schnelldurchlauf“ wiedergegeben werden. Es besteht die Möglichkeit, eine beliebige Stelle der Interaktion zu markieren und von dieser aus die Wiedergabe fortzuführen.

5 Prototyping mit Protoreto

Protoreto ist eine Open-Source-Anwendung zur Erstellung von Displayprototypen. Sie wurde in Java entwickelt, um eine Kompatibilität mit Jacareto zu gewährleisten. Sie besteht aus einem WYSIWYG-Editor, in dem der Prototyp erstellt, und einer Laufzeitumgebung, in der dieser anschließend ausgeführt werden kann (siehe Abb. 2). Des Weiteren gibt es ein Jacareto-Modul, um Protoreto-spezifische, semantische Ereignisse aufzuzeichnen.

Ein Protoreto-Prototyp besteht aus mehreren sogenannten Bildschirmen, die das simulierte System jeweils in einem bestimmten Zustand zeigen (z.B. Login-Bildschirm, Hauptmenü, Optionen-Übersicht etc.). Der Benutzer des Protoreto-Editors kann, um einen neuen Bildschirm zu erstellen, ein Hintergrundbild festlegen (z.B. das Foto eines Mobilgeräts). Anschließend kann er klickbare Bereiche definieren, die bei der späteren Benutzung des Prototyps zu einem anderen Bildschirm führen, damit ein anderer Systemzustand simuliert wird. Entsprechend können auch Standard-Bedienelemente wie Schaltflächen oder Ankreuzfelder platziert werden. Auch bei diesen Elementen können Transitionen zu anderen Bildschirmen festgelegt werden; diese werden ausgelöst, wenn z.B. auf eine Schaltfläche geklickt oder in einem Texteingabefeld die Eingabetaste betätigt wurde. Ebenso können Bilder, Animationen und Audioclips einfach im Prototyp eingebettet werden.

Ein Vorteil des WYSIWYG-Editors ist, dass auch Benutzer ohne Programmierkenntnisse einfache Prototypen erstellen können. Sie müssen lediglich Bilder und Bedienelemente per Drag & Drop platzieren und anschließend die Zielbildschirme von Schaltflächen etc. festlegen. Zugleich können erfahrene Anwender Transitionen mit Skriptcode versehen (unterstützt werden z.B. JavaScript und Ruby) und damit komplexe und dynamische Prototypen in Protoreto entwickeln. Durch Skriptcode können etwa die Eigenschaften von Bedienelementen ausgelesen oder verändert werden. Beispielsweise kann bei Klick auf einen Button die Wahl des Zielbildschirms davon abhängen, ob in einem Ankreuzfeld ein Häkchen gesetzt ist oder nicht. Außerdem können globale Variablen deklariert werden, in denen Informationen über den Zustand des Prototyps gespeichert werden können. Dies ist in anderen Prototyping-Werkzeugen wie DENIM oder RapidRabb.it nicht möglich, bei denen der Systemzustand allein dadurch kodiert ist, welche Seite momentan angezeigt wird. Somit können in Protoreto

komplexe Prototypen erstellt werden, die mit anderen Tools prinzipbedingt nicht oder nur mit extrem großem Aufwand realisierbar wären.

Ein weiteres Feature, das Protoreto von alternativen Tools absetzt, ist der nahtlose Übergang (seamless migration) von Papier- zu Displayprototypen. Die Überbrückung dieses Medienbruchs ist interessant, weil Papierprototypen durch ihre visuelle Unvollkommenheit der Versuchsperson das frühe Entwicklungsstadium verdeutlichen und sie so zu Kritik ermutigen (Snyder 2003, 58). DENIM ermöglicht es zwar, beispielsweise auf einem Tablet-PC direkt in den Displayprototyp zu schreiben oder zu zeichnen, ähnlich wie das bei Papierprototypen möglich ist; allerdings gibt es keine Möglichkeit, Zeichnungen aus den Papierprototypen der vorangegangenen Iterationsphasen in den Displayprototyp zu übernehmen. DENIM ist also konzeptuell eher ein Ersatz für Papier-Prototyping und keine Ergänzung. Dagegen kann Protoreto Blätter des Papierprototyps einscannen und in den Displayprototyp integrieren.

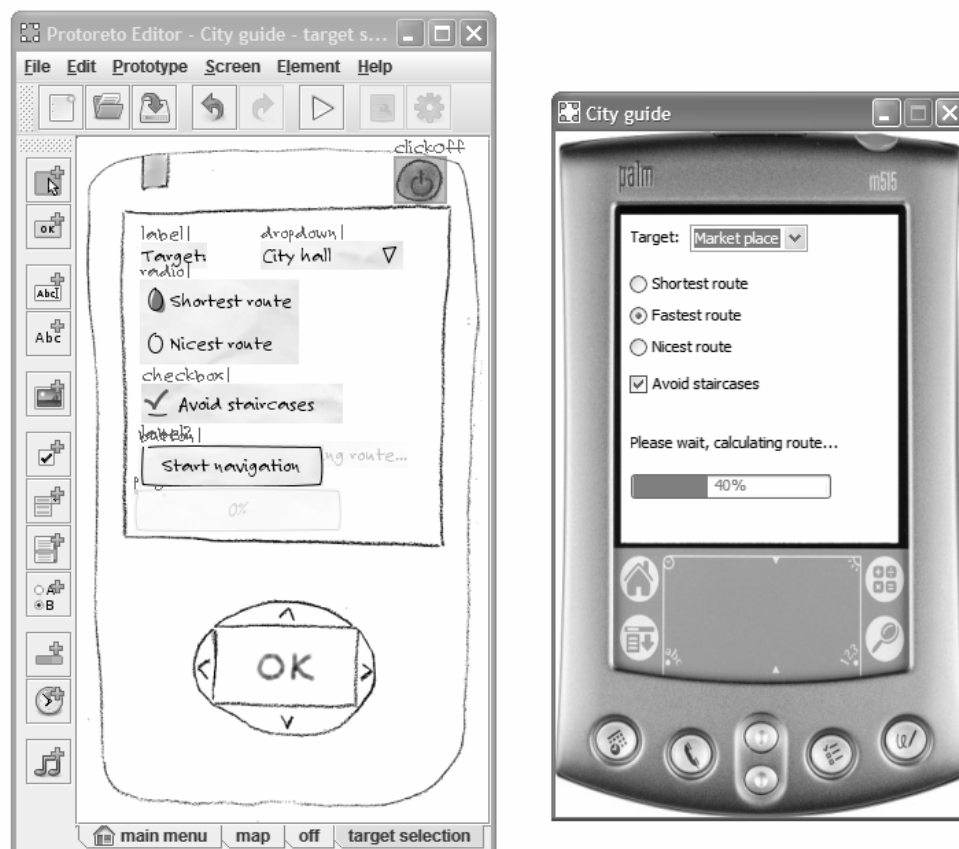


Abbildung 2: Links der Protoreto-Editor mit einem Prototyp im Look-and-Feel „NapkinLAF“, rechts die Protoreto-Laufzeitumgebung mit demselben Prototyp in einer späteren Iteration

Außerdem ist es möglich, Bedienelemente wie handgezeichnet erscheinen zu lassen. Möglich wird dies durch das von Java unterstützte "Pluggable Look and Feel", eine Technologie, welche die Darstellung und das Verhalten der GUI-Komponenten anpassbar macht. Ein solches Look and Feel ist *NapkinLAF*⁴, das, wie der Name schon andeutet, GUIs so erscheinen lässt, als wären sie auf eine Serviette gekritzelt (siehe Abb. 2). Dieser Effekt wird unter anderem durch die Verwendung von Handschrift-Fonts und die zufällige Transformation von Schaltflächen-Umrandungslinien ausgelöst.

Durch Verwendung der Scan-Funktion von Protoreto und NapkinLAF ist es somit möglich, die eingescannten Zeichnungen des Papierprototyps mit dynamischen Bedienelementen zu kombinieren und dabei eine kohärente Darstellung beizubehalten. Damit wird der Medienbruch zwischen Papier- und Displayprototypen überbrückt; wir kombinieren dabei die Hauptvorteile von Papierprototypen (Möglichkeit der Benutzerpartizipation, gutes Feedback durch visuelle Unvollkommenheit) und Displayprototypen (selbständige Interaktion der Versuchsperson, Interaktionsaufzeichnung mit Jacareto). In späteren Iterationen, wenn auch Feedback zum visuellen Erscheinungsbild benötigt wird, kann dann problemlos von NapkinLAF auf ein Standard-Look-and-Feel umgeschaltet werden (siehe Abb. 2).

6 Anwendungsbeispiel von Protoreto und Jacareto

In einer psychologischen Studie an der RWTH Aachen wurde der Umgang jüngere und ältere Menschen mit Webbrowsern auf Mobilgeräten untersucht (Ziefle et al. 2007). Da keine Werkzeuge zur Verfügung standen, die eine Interaktionsaufzeichnung auf Handhelds erlaubten, wurde eine auf Standard-PCs lauffähige PDA-Simulation in Java programmiert. Zudem wurde ein Jacareto-Modul entwickelt, das die Aufzeichnung PDA-spezifischer Ereignisse (z.B. das Wechseln der Seite im simulierten Webbrowser) ermöglichte. Damit konnte die Benutzerstudie erfolgreich durchgeführt werden. Die Forschenden empfanden die Evaluation der Aufzeichnungen gegenüber Video- oder Logfile-basierter Auswertung als besonders einfach. Allerdings waren die Entwicklung des Simulators und des zugehörigen Moduls aufwändig, sie hatten den Umfang einer Diplomarbeit (Michel 2006).

Die Entwicklung von Protoreto zielte darauf ab, derartige Simulationen bzw. Prototypen mit weitaus geringerem Aufwand zu erstellen. Der PDA-Simulator wurde als Proof-of-Concept funktionsgleich in Protoreto nachgebaut, hierfür benötigte eine Person zwei Arbeitstage statt, wie beim ursprünglichen Simulator, mehrere Monate; zudem waren keine Programmierkenntnisse notwendig. Außerdem musste kein spezielles Jacareto-Modul mehr programmiert werden, da das Protoreto-Modul für Jacareto alle relevanten Informationen liefert.

Aktuell wird das Protoreto-Jacareto-Gespann am Institut für Psychologie der RWTH Aachen eingesetzt, das in einem eHealth-Projekt die Benutzbarkeit von Small-Screen-Geräten für Diabeteserkrankte untersucht. Dazu werden einerseits existierende Diabetes-Mobilgeräte

⁴ Napkin Look & Feel. <http://napkinlaf.sourceforge.net/>

rekonstruiert, andererseits Prototypen für Systeme mit neuartigen Bedienelementen und Interaktionsschemata kreiert. In beiden Fällen wird auf Protoreto für die Entwicklung und auf Jacareto für die Aufzeichnung, Strukturierung und Analyse der Interaktion gesetzt.

7 Ausblick

Wenngleich sich mit Hilfe von Protoreto und Jacareto unter anderem Simulationen für Mobilgeräte erstellen und evaluieren lassen, gibt es einige Aspekte der Usability, die sich durch eine auf einem PC ablaufende Simulation nicht gut untersuchen lassen (beispielsweise die Schwierigkeit, Text über sehr kleine Tastaturen einzugeben). Dazu kommt, dass gerade bei Mobilgeräten eine Studie unter Laborbedingungen kaum mit Alltagssituationen vergleichbar ist (Lindroth et al. 2001). Wünschenswert wäre deshalb, Prototypen direkt auf einem Mobilgerät testen zu können. Momentan ist eine auf Java Mobile Edition lauffähige Version von Jacareto in der Entwicklung.

Eine weitere Möglichkeit, Prototypen außerhalb des Labors zu evaluieren, bieten Online-Studien. Diese hätten mehrere Vorteile: Feldstudien haben mehr Aussagekraft über eine spätere Nutzung in Alltagssituationen; den Versuchspersonen bliebe die Anreise zum Labor erspart, wenn eine Teilnahme von zu Hause aus möglich wäre; und schließlich würde dank der zunehmenden Verbreitung von Mobiltelefonen und PDAs mit vollwertigen Webbrowsern eine weitere Möglichkeit eröffnen, Prototypen auf Mobilgeräten zu evaluieren. Geplant ist deshalb eine Exportfunktion, um einen Protoreto-Prototyp in eine Website zu konvertieren.

Andere Features, die in Protoreto umgesetzt werden sollen, sind die Erweiterung um weitere Interaktionsformen wie Drag & Drop und eine Funktion, um direkt in den Displayprototyp hinein zu zeichnen, was die Verschmelzung von Papier- und Displayprototypen abrunden würde.

Literaturverzeichnis

- Balzert, H. (1998). *Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*. Heidelberg; Berlin: Spektrum Akademischer Verlag
- Bischofberger, W. & Pomberger, G. (1992). *Prototyping-Oriented Software Development: Concepts and Tools*. Berlin: Springer
- Dahm, M. (2006). *Grundlagen der Mensch-Computer-Interaktion*. München: Pearson Studium
- Dix, A., Finlay, J., Abowd, G. D. & Beale, R. (2006). *Human-computer interaction*. 3. Auflage. Harlow: Pearson Prentice-Hall
- Dumas, J. S. (2003). *User-based evaluations*. In *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*. Hillsdale: L. Erlbaum Associates, S. 1093-1117.

- Göpel, S. (2006). *Remote Usability Testing – Evaluierung der Testsoftware Morae in verschiedenen Anwendungsszenarien*. Masterarbeit, Stiftung Universität Hildesheim.
- Kappel, G. (2004). *Web Engineering: Systematische Entwicklung von Web-Anwendungen*. 1. Auflage. Heidelberg: dpunkt-Verlag.
- Lin, J., Newman, M. W., Hong, J. I. & Landay, J. A. (2000). DENIM: finding a tighter fit between tools and practice for Web site design. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York: ACM, S. 510-517.
- Ludewig, J. & Lichter, H. (2007). *Software Engineering: Grundlagen, Menschen, Prozesse, Techniken*. 1. Auflage. Heidelberg: dpunkt-Verlag.
- McCurdy, M., Connors, C., Pyrzak, G., Kanefsky, R. & Vera, AH (2006). Breaking through the fidelity barriers: An examination of our current characterization of prototypes and an example of a mixed-fidelity success. *Proceedings of CHI 06*.
- Michel, T. (2006). *Analyse kognitiv komplexer Systeme mittels automatisierter Interaktionsaufzeichnung*. Diplomarbeit, RWTH Aachen.
- Schroeder, U. (2000). PTAH: validation driven software development. *5th International Conference on Integrated Design & Process Technology 2000*, Dallas, Texas.
- Sommerville, I. (2003). *Software Engineering*. 6. Auflage. München: Pearson Studium.
- Spannagel, C. (2003). *Qualitative und quantitative Analyse von Interaktionsaufzeichnungen*. Diplomarbeit, Technische Universität Darmstadt.
- Spannagel, C., Gläser-Zikuda, M. & Schroeder, U. (2005). Application of Qualitative Content Analysis in User-Program Interaction Research. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 6(2), Art. 29, <http://nbn-resolving.de/urn:nbn:de:0114-fqs0502295>.
- Snyder, C. (2003). *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. New York: Morgan Kaufmann.
- Van Duyne, D. K., Landay, J. A., & Hong, J. I. (2003). *The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience*. Boston: Addison-Wesley.
- Wagner, A. (1990). Prototyping: A Day in the Life of an Interface Designer. In *The Art of Human-Computer Interface Design*. Addison-Wesley, S. 79-84.
- Ziefle, M., Schroeder, U., Strenk, J. & Michel, T. (2007). How younger and older adults master the usage of hyperlinks in small screen devices. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York: ACM, S. 307-316

Danksagung

Wir danken allen Beteiligten des Open-Source-Projekts Jacareto, insbesondere Dr. C. Spannagel.

Kontaktinformationen

E-Mail: {herding,schroeder,akbari}@informatik.rwth-aachen.de
Tel.: +49 241 - 80 21 932