

# Eine Werkzeugkette zur systematischen Unterstützung der Useware-Entwicklung

Nancy Thiels, Daniel Görlich

Zentrum für Mensch-Maschine-Interaktion (ZMMI)

TU Kaiserslautern

Postfach 3049, Kaiserslautern

thiels@mv.uni-kl.de

goerlich@mv.uni-kl.de

**Abstract:** Although the user-orientation of user interfaces has become a decisive factor for the acceptance of technical devices, their development still poses an enormous challenge to both industrial companies and researchers. The analysis of the users and their tasks, as well as the devices' technical functionalities, and ranges and fields of applications, is instrumental in the development of user-oriented user interfaces. In a Useware development process, data gained from such an analysis can be applied systematically to the conception and structuring of the user interface to be designed; furthermore, rules and guidelines strongly influence all steps of this design process. The Useware development process in its entirety, however, is not yet supported by computer-based tools. At its best, dedicated tools for individual phases or steps accompany development teams, which is not good enough by half for holistic data and knowledge management spanning the whole development process. This paper therefore introduces a tool chain that can support development teams in a systematic Useware development process, spanning all its phases from the initial analyses to the actual implementation.

## 1 Systematisches Useware-Engineering

Während die Nutzerorientierung von Bediensystemen längst zu einem entscheidenden Faktor für die Akzeptanz technischer Geräte geworden ist, stellt deren Entwicklung heutzutage noch immer eine enorme Herausforderung dar. Der steigende Aufwand bei der Bediensystementwicklung auf Grundlage wissenschaftlicher Methoden hat zu einem Entwicklungsanteil geführt, der mindestens ebenso groß ist wie die traditionellen Felder des Hard- und Software-Engineerings. So wurde 1999 in Analogie zu den Begriffen Hard- und Software der Begriff *Useware* als Sammelbegriff für alle Komponenten, die der Benutzung dienen, eingeführt [Züh99]. Mit ihm verbindet sich eine Fokussierung der Technikgestaltung auf menschliche Fähigkeiten und Bedürfnisse.

Maßgeblich für die Entwicklung nutzerorientierter Bediensysteme ist heute eine ganzheitliche Analyse der Nutzer, ihrer Aufgaben, der technischen Funktionalitäten des Gerätes und der Umgebungsbedingungen. Aufgrund der Personalstrukturen fehlen in

vielen Unternehmen aber häufig die Kompetenzen, die eine ganzheitliche Bediensystementwicklung aufgrund umfassender Analysen zulassen würden. Daher werden heute von den Marketingabteilungen vieler Unternehmen zwar routinemäßige Marktanalysen vorgenommen, doch beschäftigen sich diese meist mit der Vermarktung des Produktes, anstelle seiner ganzheitlichen, systematischen Entwicklung und Nutzerorientierung.

Die im Folgenden vorgestellte Werkzeugkette wird – nicht nur Unternehmen – die gewünschte ganzheitliche, systematische Entwicklung von Bediensystemen ermöglichen bzw. vereinfachen. Bereits für die bisher so gut wie gar nicht durch computerbasierte Werkzeuge unterstützte Analysephase wird ein solches für die Datenerhebung, -analyse und -aufbereitung, für die Definition von Anforderungen und die Modellierung von Aufgaben, Datenstrukturen und Gerätefunktionen zur Verfügung gestellt werden.[Gör06]

Ebenso wird die eigentliche Gestaltung des Bediensystems und seiner Benutzungsschnittstellen im weiteren Verlauf der Entwicklung systematisch unterstützt, indem den Entwicklern ein zweites Werkzeug für die Gruppierung und Priorisierung von Bedientätigkeiten (Aufgaben) an die Hand gegeben wird. Weiterhin wird dieses Werkzeug Möglichkeiten zur Beschreibung des Verhaltens des Bediensystems bereitstellen. Zuletzt wird damit das Systemumfeld modelliert werden können, angefangen von den Hard- und Software-Plattformen, auf denen das Bediensystem aufbaut, bis hin zur Struktur seiner Benutzungsschnittstelle.

Ein drittes Werkzeug wird dem Entwickler bei der graphischen Ausgestaltung des Bediensystems behilflich sein und ihm automatisiert generierte Prototypen zum Testen und Evaluieren zur Verfügung stellen.

Eine Werkzeugkette, die so gezielt die frühen Phasen der Entwicklung durchgängig unterstützt, bietet einen bis dato nicht gekannten Grad der computerbasierten Unterstützung der Entwickler. Sie hilft dadurch, Fehler frühzeitig zu erkennen und zu vermeiden, die ohne eine systematische Entwicklung und prozessbegleitende Evaluation erst sehr spät erkannt würden – eine Problematik, die heutzutage häufig Verteuerungen und Verzögerungen von Entwicklungen verursacht und nicht selten zum Scheitern von Projekten beiträgt.

Die zu erwartende Verringerung von Fehlentwicklungen und Projektfehlschlägen wird durch ständige, semi-automatische Evaluation in den frühen Phasen der Entwicklung, sowie durch die Eliminierung der häufigen Medienbrüche erreicht. Die einheitliche, strukturierte Entwicklung dürfte überdies die Arbeit von Projektteams erleichtern, Entwicklungszeiten verkürzen, dadurch Kosten einsparen und insbesondere Qualitätsansprüche durchsetzen und sichern.

## **2 Architektur der Werkzeugkette**

Die systematische Bediensystem-Entwicklung nach dem am Zentrum für Mensch-Maschine-Interaktion (ZMMI) erarbeiteten Useware-Entwicklungsprozess (Abb. 1)

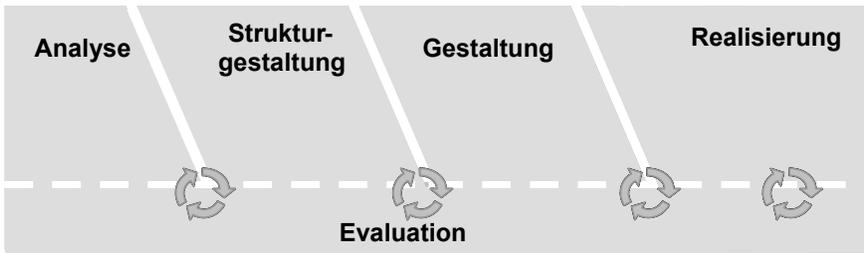


Abbildung 1: Ueware-Entwicklungsprozess nach [Züh99]

erfordert den Einsatz mehrerer Werkzeuge. Dabei überschneiden sich einerseits Arbeitsschritte, wodurch eine Entwicklung gemäß der Wasserfall-Methodik nicht mehr anwendbar ist; die Abhängigkeiten zwischen den Arbeitsschritten erfordern andererseits wiederum eine gewisse Sequenzialität. Die Flexibilität, sowohl Parallelisierbarkeit von Entwicklungsschritten als auch die Konsistenz der Zwischenergebnisse zu gewährleisten, stellt die Architektur der Werkzeugkette mittels einer integrierten „Produktdatenbank“ mit produkt- bzw. projektspezifischen Daten sicher. Die unterschiedlichen Werkzeuge können darin zu beliebiger Zeit ihre (Zwischen-)Ergebnisse ablegen, die wiederum als Eingangsgrößen für andere Werkzeuge dienen können. Greift ein Werkzeug schreibend auf die Produktdatenbank zu, so werden bei Bedarf alle anderen, die jene geänderten Daten als Eingangsgrößen verwenden, benachrichtigt, was wiederum automatische Anpassungen in allen Projektphasen zur Folge haben kann.

Zusätzlich wird eine Wissensbasis die Anwendung allgemeingültigen Wissens durch die Entwicklungswerkzeuge ermöglichen. Dabei handelt es sich etwa um Vorschriften, Vorlagen oder Erfahrungen, die nicht projektspezifisch sind, und von Werkzeugen z.B. als Anleitung oder zu Konsistenzprüfungen (Evaluation) genutzt werden.

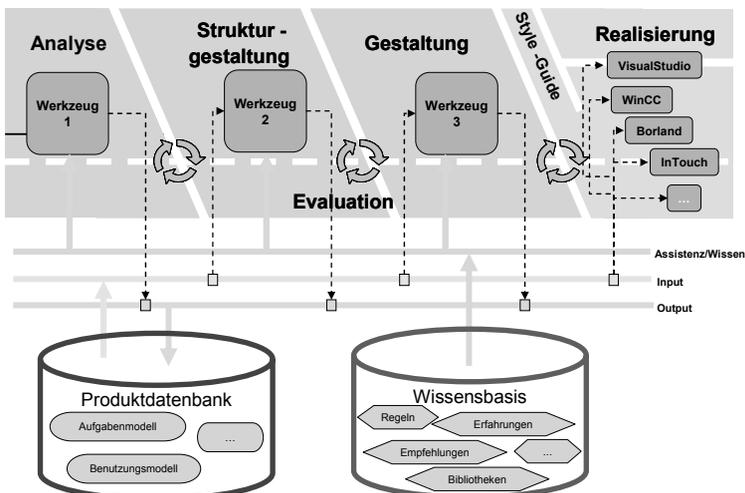


Abbildung 2: Architektur der Werkzeugkette

Die Gesamtheit der Werkzeuge ähnelt einem Expertensystem. Die Architektur der Werkzeugkette sowie die Anbindung der Wissensbasis und der Produktdatenbank an die einzelnen Werkzeuge ist in Abbildung 2 dargestellt.

## **2.1 Produktdatenbank und projektübergreifende Wissensbasis**

Bei der Entwicklung ergonomischer Bediensysteme müssen zahlreiche Richtlinien und Vorschriften, aber auch Erfahrungen beachtet werden. Heutigen Entwicklungswerkzeugen fehlen Möglichkeiten, derartiges Wissen und Erfahrungen aufzunehmen und sie an- bzw. wieder zu verwenden. Das theoretische Fundament wurde jedoch gelegt und hat verschiedene Repräsentationsformen hervorgebracht, bspw. logikbasierte Methoden wie Prädikatenlogik erster Stufe oder Fakten-Regel-Systeme [Got90], Frames [Mis74], semantische Netze [Win93], Constraints [Pup91], Description Logics [Nar03] und Feature Logic [Smo92]. Auch Arten von Wissen wurden bereits detailliert klassifiziert. So unterscheidet etwa [Thu89] zwischen Wissen über Objekte, über Beziehungen, über Vorgehensweisen, sowie über Regeln & Vorschriften.

Die Anforderung, eine Wissensbasis für die Werkzeugkette zu definieren, gliedert sich also in die beiden Schritte der Evaluation und Auswahl erstens von Arten des zu modellierenden und abzubildenden Wissens, und zweitens der Strukturierung und Repräsentation jenes Wissens. Parallel dazu werden geeignete Speicher- und Zugriffsverfahren sowie Schnittstellen und schnelle Assoziationsverfahren ausgewählt oder entwickelt sowie implementiert werden, über die die einzelnen Werkzeuge der Werkzeugkette später auf das hinterlegte Wissen zugreifen können.

Nachdem das ZMMI einen Grundstock an Wissen in die Wissensbasis angelegt hat, kann dieser Grundstock durch den Einsatz der Werkzeugkette in Projekten sukzessive erweitert werden. Er kann als Grundlage für spätere Projekte herangezogen werden.

Zusätzlich zur projektübergreifenden Wissensbasis, die allgemeingültiges Wissen und häufig wieder verwendbare Daten bereithält, wird eine Produktdatenbank die produkt- bzw. projektspezifischen Informationen vorrätig halten. Ebenso wie die Wissensbasis wird sie allen Werkzeugen der Werkzeugkette zur Verfügung stehen und über definierte Schnittstellen Rohdaten und interpretierte Information liefern.

So wird die Werkzeugkette als ganzheitliches Entwicklungssystem ihren Anwendern beispielsweise erlauben, zu jeder Zeit während des Entwicklungsprozesses Regeln (z.B. Gestaltungsregeln, Anforderungen etc.) zu definieren bzw. anzupassen. Sie wird die Anwender bei der Erstellung der jeweils projektspezifischen Datenbasis unterstützen und auf Verstöße, Unvereinbarkeiten, Unregelmäßigkeiten und ähnliche Probleme beispielsweise mit den zusammengetragenen Daten, den gestellten Anforderungen, der entworfenen Bediensystemstruktur usw. hinweisen. Um Rückschritte im Entwicklungsprozess zu vermeiden, wird die Datenbasis jederzeit veränderlich sein, ohne dass die aus späteren Entwicklungsphasen gewonnenen Ergebnisse dadurch obsolet werden.

## 2.2 Werkzeuge

Die Werkzeugkette wurde als eine solche konzipiert, da ihre Unterteilung in mehrere dedizierte, aber kooperierende Werkzeuge der Tatsache Rechnung trägt, dass die Integration aller Teilschritte in einem einzigen Werkzeug dessen Komplexität stark erhöhen und somit den Entwicklungsprozess beeinträchtigen würde. Zudem sehen fast alle wissenschaftlichen wie auch kommerziell/industriellen Entwicklungsprojekte die Erreichung von Zwischenzielen, Meilensteinen oder Teilergebnissen vor, die sowieso aus der Werkzeugkette heraus generiert bzw. gewonnen werden müssen. Die Einzelwerkzeuge ermöglichen daher (auch durch Assistenzfunktionen) das zielgerichtete Erreichen solcher Teilergebnissen, die wiederum in die Datenbasis eingehen und somit für weitere Entwicklungsschritte sowie zur Evaluation der bisherigen Teilergebnisse und des Projektverlaufs zur Verfügung stehen.

Entsprechend des in Abbildung 1 dargestellten Useware-Entwicklungsprozesses ist je ein Werkzeug für jede der ersten drei Phasen (Analyse, Strukturgestaltung und Gestaltung) vorgesehen; für die Realisierung ist aufgrund der umfangreichen Unterstützung durch Programmierwerkzeuge kein eigenes Werkzeug geplant. Statt dessen erlauben Export-Filter und Konverter eine Kopplung der Werkzeugkette an jene Programmierwerkzeuge.

Das Werkzeug für die Analysephase wird die strukturierte Aufnahme aller qualitativen, quantitativen und strukturellen Daten einer ganzheitlichen Analyse (Erstellung, Erhebung, Aus- und Bewertung) ermöglichen sowie Funktionalitäten und Schnittstellen zur Interpretation, Präsentation und Weitergabe jener Daten bieten. Deren Aufbereitung umfasst unter anderem die Definition von Anforderungen an das Bediensystem, insbesondere die Erarbeitung von Aufgabenmodellen und Workflows der Anwender ebenso wie der Datenstrukturen und Funktionsmodelle des Bediensystems [Böd06].

In der darauf folgenden Phase der Strukturgestaltung wird das entsprechende Werkzeug die Modellierung von Benutzungsstrukturen, die den Nutzer durch eine konsequente Aufgabenorientierung und die Entwicklung hardwareunabhängiger Benutzungsmodelle in den Vordergrund stellen, ermöglichen. Die Modellierung basiert dabei auf einer umfassend semantisch definierten Auszeichnungssprache, der am ZMMI entwickelten Useware Markup Language (useML) [Reu03].

Zuletzt wird das dritte Werkzeug die Gestaltung der späteren Bediensysteme aufgrund von Plattformspezifikationen unterstützen. Es verfolgt dabei einen Top-Down-Ansatz, also die Gestaltung vom Groben zum Detail, um insbesondere auch die Entwicklung von Geräte- und Softwareproduktfamilien lange Zeit gemeinsam durchzuführen und dadurch sowohl Entwicklungsaufwand zu sparen als auch Konsistenz zu gewährleisten. Dieses Werkzeug wird bereits eingeschränkt funktionsfähige Prototypen zum Testen und Evaluieren generieren.

Letztendlich wird die Werkzeugkette für die abschließenden Phasen des Entwicklungsprozesses Möglichkeiten zum Export der erzielten Ergebnisse zwecks Weiterverarbeitung in ausgewählten Fremdanbieter-Werkzeugen (wie etwa ECLIPSE für UML & Java, SQL-Datenbanken etc.) offerieren. Die Unterstützung des Anwenders

durch die Werkzeugkette endet nach der Gestaltungsphase, da die Realisierung (Programmierung) des entwickelten Bediensystems von zahlreichen – etablierten, wenn auch proprietären – Werkzeugen verschiedenster Anbieter aus Forschung und Industrie bereits umfassend unterstützt wird.

### 3 Ausblick

Mit der Definition der Useware Markup Language [Reu03], der Spezifikation eines systematischen Useware-Entwicklungsprozesses [Züh99], der Entwicklung erster Tools zur systematischen Entwicklung nutzergerechter Maschinenbediensysteme [Wah00] sowie zur semi-automatischen Generierung graphischer Benutzungsschnittstellen (GUI's), und ebenso mit der Definition von Methoden zur Datenerhebung und -strukturierung in der Analysephase wurden am Zentrum für Mensch-Maschine-Interaktion bereits bedeutende Vorarbeiten geleistet.

Langfristig soll die Werkzeugkette die nutzerzentrierte Entwicklung von Bediensystemen fördern und unterstützen, da aufgrund der prognostizierten zukünftigen Interaktionsparadigmen wie beispielsweise dem der Ambient Intelligence die Komplexität sowohl bei der Entwicklung als auch der Nutzung technischer Geräte noch weiter steigen wird. Traditionelle Herangehensweisen und Techniken werden aufgrund der Unbestimmtheit der An- und Verwendung von Technologien und Geräten nicht mehr praktikabel sein. Nur eine generische, modellbasierte Herangehensweise, die auch eine Online-Generierung der Bediensysteme gestattet, wird in diesen Umgebungen eine effiziente und intuitive Entwicklung von Bediensystemen ermöglichen.

### Literaturverzeichnis

- [Böd06] Bödcher, A.: Strukturierte Analyse als Grundlage für die USEWARE-Entwicklung. In: USEWARE 2006, VDI-Verlag, Düsseldorf, 2006. (Im Druck.)
- [Got90] Gottlob, G. et. al.: Expertensysteme. Springer Verlag, Wien, 1990.
- [Gör06] Görlich, D.; Thiels, N.: Eine Werkzeugkette zur Unterstützung der USEWARE-Entwicklung. In: USEWARE 2006, VDI-Verlag, Düsseldorf, 2006. (In Arbeit.)
- [Mis74] Minsky, M. A.: A framework for representing knowledge. Artificial Intelligence Memo 306, MIT AI Lab, 1974.
- [Nar03] Nardi, D.; Brachman, R. J.: An Introduction to Description Logics. In: The Description Logic Handbook. Baader et al. Cambridge University Press, 2003.
- [Pup91] Puppe, F.: Wissensrepräsentation mit Regeln. In: Struß, P. (Ed.): Wissensrepräsentation. Oldenbourg, München, 1991.
- [Reu03] Reuther, A.: useML – Systematische Entwicklung von Maschinenbediensystemen mit XML. Dissertation, Technische Universität Kaiserslautern, Fachbereich Maschinenbau und Verfahrenstechnik, 2003.
- [Smo92] Smolka, G.: Feature-constrained logics for unification grammars. Journal of Logic Program. Vol 12, 1992, pg. 51-87.
- [Thu89] Thuy, N. H. C.; Schnupp, P.: Wissensverarbeitung und Expertensysteme, Oldenbourg, München, 1989.

- [Wah00] Wahl, M.: Systematische Entwicklung nutzergerechter Maschinenbedienoberflächen. Fortschritt-Berichte pak, Band 3. Kaiserslautern: Verlag Universität Kaiserslautern, 2000.
- [Win93] Winston, P. H.: Artificial Intelligence. Addison-Wesley, Reading, 1993.
- [Züh99] Zühlke, D.; Wahl, M.: Hardware, Software – Useware. In: Elektronik, Nr. 23, 1999, S. 54-62.
- [Züh04] Zühlke, D.: Useware Engineering für technische Systeme. Springer Verlag, 2004.