



German Chapter der Usability Professionals' Association e.V.



Spezifikation von Benutzeroberflächen

Thesen zum aktuellen Industriestand der Entwicklung von
Oberflächen für kommerzielle Anwendungssoftware

UP-Track der Konferenz Mensch & Computer 2005
4. – 7. September 2005, Johannes Kepler Universität Linz



Referenten



Paul Chlebek
UML • OOD • HCI/GUI • MDA • XSLT

Diplom-Informatiker (FH) Paul Chlebek ist selbständiger HCI/UML/OOD Spezialist mit den Arbeitsschwerpunkten Methodenentwicklung, Usability Engineering und UI-Konzeption.

Er hat internationale praktische Projekterfahrung als leitender Entwickler, Internationalisation Engineer, Projektleiter, Entwicklungsleiter und Softwarearchitekt. Derzeit ist er für Banken und Automotive-Kunden tätig.



Markus Hillebrand
MDSD • MDA • CASE • UML • XML

Markus Hillebrand, Diplom-Informatiker (FH), hat zusätzlich Philosophie und Soziologie studiert und ist selbständiger Software-Entwickler und Software-Engineer.

Er ist Mitinitiator des offenen Software-Labors „www.eos-project.org“ mit dem Teilprojekt „www.benutzeroberflaeche.de“. Derzeit ist er für die BMW-Group als Methodiker für HCI & MDSD tätig.





Inhalt

1. Überblick
2. Thesen
3. Diskussion

Weitere Materialien finden Sie unter:

<http://www.benutzeroberflaeche.de>

www.benutzeroberflaeche.de ist ein Open Knowledge
Projekt für Entwicklung von Benutzeroberflächen





Rahmenbedingungen: Um diese HCIs geht es

- Vertrag über Erstellung eines Werks.
- Vertragsgegenstand ist nicht nur die Funktion sondern auch die Art und Weise der Bedienung.
- Auftragnehmer und/oder Auftraggeber hat eine Organisation mit arbeitsteiligen Prozessen.
- Industrielle Produkte, kommerzielle Anwendungen (z.B. Produktvarianten für verschiedene Märkte).





Oberflächen zu spezifizieren lohnt sich

1. Oberflächenspezifikationen sind unumgänglich, weil sie Grundlage für Test und Abnahme der beauftragten Anwendung sind. Sie ermöglichen die rechtssichere Zusammenarbeit.
2. Fachspezialisten beschreiben die beauftragte Anwendung oft durch Masken und Maskenabfolgen. Die Oberfläche steht damit im Mittelpunkt der Abstimmungsgespräche über die Anwendung.
3. Aus Oberflächeneigenschaften ergeben sich Hinweise auf Funktionen und Daten. Die Anforderungen, Randbedingungen und Wechselwirkungen der Bedienung von Funktionen gehen über das im Code Formulierbare und an der Oberfläche sicht- und erlebbare hinaus.
4. Eine formale Spezifikation erlaubt ihre frühzeitige Verifikation, eine Teilgenerierung der Implementierung und die Testautomatisierung.





Benutzeroberflächen können nicht aus der Funktionalität abgeleitet werden

1. Die Bedienung der Anwendung wird durch Informationen beschrieben, die nicht im Daten- und Funktionsmodell zu finden sind: z.B. Layout, Design, Bedienungsreihenfolge, Hervorhebung von wichtigen Bedienungselementen.
2. Für die Bedienung interaktiver Systeme genügt das EVA Prinzip nicht: Die Schnittstellen zwischen Nutzer und Anwendung bestehen nicht auf funktionaler Ebene, sondern erfolgen durch interpretative Wechselwirkung von Ein- und Ausgaben.





Mit einer Maschine kommuniziert man nicht, sondern man benutzt sie

1. HCI ist keine Kommunikation, sondern die Bedienung von Funktionen.
2. HCIs sind hochkomplexe Bedienertafeln, nicht mehr und nicht weniger. Deren Bau orientiert sich an Kommunikation, weil „Kommunikation“ die komplexeste Interaktionsdefinition liefert, zu der Menschen fähig sind.
3. Wir erkennen eine Maschine nicht als Gesprächspartner an: Adaptivität ist nur in engen Grenzen sinnvoll.





Informationsmodelle für Spezifikation und Programmierung sind verschieden

1. Eine Oberflächen-Spezifikation hat Planungscharakter und soll daher eindeutig, abstrakt und schnell zu ändern sein, um auf wechselnde Anforderungen und Kundenwünsche einzugehen, vergleichbar den Architektur-Skizzen und Innenarchitekturplänen von Gebäuden.
2. Der Quellcode der Implementierung ist eine detaillierte Verfahrensanweisungen mit IO-Anweisungen für eine spezielle Ablaufmaschine.
3. Einigkeit der Stakeholder und detaillierte Umsetzung der Anforderungen sind zwei verschiedene Ziele, das Informationsmodell der Spezifikation soll aber in der Implementierung weiterverwendet werden können.





Unterspezifikation ist in Ordnung

1. Eine totale Detaillierung ist speziell im Software-Bereich ein Grenzfall zur Implementierung und ökonomisch nicht sinnvoll.
2. Wie tief spezifiziert werden muss, hängt also vom Projektkontext und dem Interpretationsvertrauen in die Umsetzer ab.
3. Problem ist derzeit die mangelnde Standardisierung von Oberflächenbestandteilen, Werkzeugen und Ausbildung in Verfahren.





State-Charts sind für die Modellierung von UIs nicht genug

1. Trotz der Möglichkeit, Zustandshierarchien zu bilden, werden State-Charts im HCI-Bereich schnell groß und unübersichtlich.
2. Über State-Charts hinausgehende Möglichkeiten zur Komplexitätsreduktion werden zu wenig beachtet: Verbesserungen sind z.B. Sequenzbildungen, prototypische Vererbung, Vermeidung von Transitionen.
3. State-Charts sind allgemeine Ablaufbeschreibungen, im HCI-Bereich sind spezialisierte Konstrukte sinnvoll: z.B. Ablaufschritte, Bildschirme, Bedienungselemente.





UI Spezifikationen funktionieren besser mit Steps statt States

1. In State-Charts sind Übergänge und Zustände nicht gewichtet und daher alle gleich priorisiert.
2. Ergonomie unterscheidet jedoch die wichtigeren Pfade und die weniger wichtigen Pfade, um z.B. die Zahl der Klicks zu wichtigen „Zielen“ minimieren zu können.
3. In HCIs gibt es Haupt- und Nebenpfade. Dies sollte sich in der Spezifikation wiederfinden.
4. In HCI werden strukturierte Bedienungsabläufe und die daran hängenden Bedienungselemente beschrieben, mit States sind nur unstrukturierte Abläufe möglich.





Eine eindeutige Spezifikationsprache für UIs ist notwendig und möglich

1. Eine eindeutige Spezifikationsprache für Benutzeroberflächen ist notwendig, um UI Entwicklung missverständnisarm abstimmen und koordinieren zu können.
2. Sie ist möglich, weil das UI programmiert werden kann. Eine Spezifikationsprache entsteht, wenn man die Details technischer UI-Umsetzungen weglässt (=Abstraktion der Technik).
3. Sie muss aus der Erfahrung gebildet werden, da sie nicht aus einer nicht vorhandenen allgemeinen HCI-Theorie ableitbar ist.





Fazit: Rückblick und Ausblick

- Methoden und Prozessunterstützung für durchgängige Spezifikation, Implementierung und Test sind not only nice to have.
- Werkzeuge müssen auf einem gemeinsamen Informationsmodell zusammenarbeiten.
- Spezifizieren schafft Prozess-Sicherheit.
- Sicherheit braucht eindeutige Formen.
- „Konkreto“: Eine Spezifikationssprache für Benutzeroberflächen.





Diskussion der Thesen

1. Oberflächen zu spezifizieren lohnt sich.
2. Benutzeroberflächen können nicht aus der Funktionalität abgeleitet werden.
3. Mit einer Maschine kommuniziert man nicht, sondern man benutzt sie.
4. Informationsmodelle für Spezifikation und Programmierung sind verschieden.
5. Unterspezifikation ist in Ordnung.
6. State-Charts sind für die Modellierung von UIs nicht genug.
7. UI Spezifikationen funktionieren besser mit Steps statt States.
8. Eine eindeutige Spezifikationssprache für UIs ist notwendig und möglich.





Danke



Please visit our HCI homepage at
<http://www.benutzeroberflaeche.de>
<http://www.eos-project.org>

