

Adaptive Benutzerschnittstellen

David Hannappel

Veranstaltung: Adaptive Systeme

Wintersemester 2005

Dozent: Prof.Dr.-Ing. J. Ziegler

Tutor: Dipl.-Inform. Tim Hussein

Februar 2006

Inhaltsverzeichnis

1	Einführung	3
1.1	Motivation	3
1.2	Geschichte der Benutzerschnittstellen	3
2	Grundlagen adaptiver Benutzerschnittstellen	4
2.1	Begriffsdefinition	4
2.2	Voraussetzungen	4
2.3	Adaptierbare und Adaptive Systeme	4
2.4	Anforderungen an adaptive Benutzerschnittstellen	5
2.5	Was wird adaptiert?	5
2.6	Automatische und Teilautomatische Adaption	5
2.7	Adaptionstechniken	5
2.8	Nutzen und Anwendungsszenarien	6
3	Adaptionstechniken	6
3.1	Idee	6
3.2	Benutzermodell	6
3.3	Generische Benutzermodellierung durch Beobachtung von Handlungssequenzen	7
3.4	Aktionsvorhersage-Algorithmen	7
4	XUL als deklarative Beschreibungssprache für Benutzerschnittstellen	8
5	Anwendungsbeispiele für Adaptive Benutzerschnittstellen	8
6	Zusammenfassung und Ausblick	9
	Literatur	11

1 Einführung

1.1 Motivation

In den letzten Jahrzehnten gab es im Bereich der Mensch-Computer-Interaktion eine Vielzahl an Entwicklungen und Forschungen. Computersoftware ist heutzutage deutlich interaktiver gestaltet, als es früher der Fall war. Es findet mehr Kommunikation zwischen Anwender und Software statt. Das resultiert in besonders hohen Anforderungen an Benutzerschnittstellen, welche für die Interaktion zwischen Benutzer und Computerprogramm zuständig sind.

Adaptive Benutzerschnittstellen vereinfachen und verbessern die Interaktion des Anwenders mit der Software, indem sie sich automatisch an die Gewohnheiten und Bedürfnisse des Anwenders anpassen.

In dieser Ausarbeitung stelle ich die Entwicklung, Grundlagen sowie die Funktionsweise Adaptiver Benutzerschnittstellen vor.

1.2 Geschichte der Benutzerschnittstellen

Rechner der ersten Generation verfügten über Lochkarten-Benutzerschnittstellen. Der Nutzer konnte seine Eingaben über Lochkarten machen, auf denen Binärkombinationen eingestanzt waren. Die Verarbeitung der Karten dauerte in der Regel mehrere Stunden und die Ausgabe erfolgte über einen Ausdruck.

In den Siebziger Jahren folgten Betriebssysteme wie z.B. MS-DOS, die eine interaktive Verarbeitung über Kommandozeile ermöglichten.

Die Benutzerschnittstellen wurden im Laufe der Zeit immer mehr mit grafischen Elementen versehen, um die Interaktion zu vereinfachen. Anfangs wurde dies noch über Textbausteine und Symbole erreicht, um das Text-Terminal übersichtlicher zu gestalten. Mitte der Achtziger Jahre kamen grafische Benutzeroberflächen wie Apple Macintosh oder Microsoft Windows auf den Markt. Durch die graphischen Elemente soll die Interaktion zwischen Benutzer und Maschine möglichst intuitiv und selbsterklärend ablaufen. Mit den grafischen Benutzeroberflächen wurde die Computermaus als neues Eingabegerät eingeführt.

Im Laufe der Zeit kamen neue Entwicklungen wie akustische sowie haptische Benutzerschnittstellen auf den Markt. Akustische Benutzerschnittstellen interagieren über Geräusch- und Spracherkennung mit dem Benutzer. Haptische Benutzerschnittstellen erkennen Bewegungen des Nutzers und setzen diese in digitale Eingabedaten um.

Heutzutage implementieren die meisten Betriebssysteme eine graphische Benutzerschnittstelle (GUI).

Ansätze adaptiver Benutzerschnittstellen existieren seit Mitte der neunziger Jahre. Adaptivität bzw. Personalisierung spielen heutzutage in Zeiten des World Wide Webs eine immer größere Rolle.

2 Grundlagen adaptiver Benutzerschnittstellen

2.1 Begriffsdefinition

Mit dem Wort „Adaption“ bezeichnet man die Fähigkeit, sich oder ein Objekt an die Umgebung und externen Faktoren anzupassen. Adaptive Benutzerschnittstellen (Adaptive User Interfaces) passen sich an die Ziele und Bedürfnisse des Anwenders an. Dabei kommen Techniken der Künstlichen Intelligenz zum Einsatz, um die Bedürfnisse des Nutzers „vorherzusagen“ [Die93].

Eine andere Definition stammt von Langley [Lan97]:

„An adaptive user interface is an interactive software system that improves its ability to interact with a user based on partial experience with that user.“

Verhaltensmuster, Ziele und Präferenzen des Benutzers werden üblicherweise in einem sogenannten Benutzermodell gespeichert.

2.2 Voraussetzungen

Adaptive Benutzerschnittstellen, die sich eigenständig an den Nutzer anpassen können, erfüllen im Prinzip immer die Voraussetzungen für den Einsatz in Softwaresystemen. Dennoch lohnt sich deren Einsatz in manchen Softwaresystemen mehr als in Anderen. Natürlich ist es unmöglich, dass ein Software System jemals alle Bedürfnisse, Ziele und Fähigkeiten eines Nutzers antizipiert. Adaptive Benutzerschnittstellen sollten dann implementiert werden, wenn die Software nicht auf eine bestimmte Anwendergruppe angepasst werden kann, sondern sich an einem breiten Benutzerkreis mit unterschiedlichen und schwer voraussehbaren Vorkenntnissen orientieren soll. Hier kann Adaptivität eine höhere Akzeptanz und Zufriedenheit bei dem Anwender bewirken, sodass Aufgaben effizienter bewältigt werden können.

Dieser positive Effekt stellt sich jedoch nur ein, wenn gewisse Voraussetzungen gegeben sind [SK]:

Die automatische Anpassung muss transparent sein, d.h. sie muss einen erkennbaren und nachvollziehbaren Nutzen für den Anwender bringen. Veränderungen, deren Grund der Nutzer nicht durchschauen kann, irritieren und stören den Anwendender.

2.3 Adaptierbare und Adaptive Systeme

Ein System heißt adaptierbar, wenn es vom Anwender manuell auf seine Bedürfnisse angepasst werden kann. Die meisten modernen Software-Anwendungen lassen sich manuell konfigurieren und sind damit adaptierbar.

Adaptiv nennt man ein solches System, das sich automatisch an den Benutzer anpasst. Der Anwender muss sich nicht mit der Konfiguration des Systems beschäftigen, da dies automatisch und im Hintergrund geschieht.

2.4 Anforderungen an adaptive Benutzerschnittstellen

Eine adaptive Benutzerschnittstelle sollte Ziele und Bedürfnisse des Anwenders antizipieren. Die Schnittstelle sollte aus der Interaktion mit dem Anwender lernen und sich mit Hilfe der gesammelten Erfahrungen immer besser an das Nutzerverhalten anpassen. Die wichtigste und schwierigste Funktion der adaptiven Software ist die Auswertung des Nutzerverhaltens durch adaptive Algorithmen. Die Auswertung sollte automatisch, unsichtbar für den Nutzer, und effizient zur Laufzeit geschehen.

Die Änderungen an der Schnittstelle tragen nur dann zu einer Verbesserung der Mensch-Maschine-Kommunikation bei, wenn der Anwender die Änderungen nachvollziehen kann bzw. einen wirklichen Nutzen daraus ziehen kann.

2.5 Was wird adaptiert?

Wenn wir von adaptiven Benutzerschnittstellen sprechen, kann man zwischen der Adaption der Inhalte und einer Anpassung des Layouts unterscheiden.

Langley [Lan97] differenziert zwischen „informative interfaces“ und „generative interfaces“ .

„Informative interfaces“ filtern Informationen, indem sie nur die Teile zeigen, die der Anwender nützlich und interessant finden würde. Ein typisches Beispiel hierzu sind Artikelempfehlungen, wie sie z.B. vom Onlineshop Amazon generiert werden.

Die zweite Klasse Benutzerschnittstellen, die „generative interfaces“ strukturieren und gliedern Daten für den Nutzer, wie es z.B. in Tabellenkalkulationen der Fall ist.

2.6 Automatische und Teilautomatische Adaption

Die Adaption der Benutzerschnittstelle sollte automatisch oder teilautomatisch erfolgen. Bei der teilautomatischen Durchführung werden dem Anwender Vorschläge zur Adaption präsentiert. Der Anwender kann wählen, ob er die Adaption durchführen möchte, oder nicht. Diese Art der Adaption wird vom Anwender laut [SK] besser angenommen, als eine „Zwangsadaption“ .

2.7 Adaptionstechniken

Adaptive Benutzerschnittstellen werden häufig durch Nachbildung menschlicher Denkmuster implementiert. Dabei kommen Techniken wie Künstliche Intelligenz, Neuronale Netzwerke und diverse Algorithmen zur Mustererkennung zum Einsatz.

Neuronale Netze zeichnen sich besonders durch ihre hohe Lernfähigkeit, parallele und fehlertolerante Datenverarbeitung aus. Sie können eine Aufgabe anhand von Trainingsbeispielen erlernen, ohne dazu explizit programmiert werden zu müssen.

Die Komplexität derartiger Techniken ist der Grund dafür, dass adaptive Benutzerschnittstellen bisher nur selten bzw. nur ansatzweise umgesetzt wurden.

2.8 Nutzen und Anwendungsszenarien

Das übergeordnete Ziel adaptiver Benutzerschnittstellen ist, durch die automatische Anpassung an die Bedürfnisse des Anwenders einen erkennbaren Zusatznutzen zu schaffen, sowie die Effektivität der Nutzung zu steigern. Hierzu muss die Software das Nutzungsverhalten protokollieren, wiederholt analysieren und Benutzerprofile speichern und anpassen.

Als Anwendungsszenarien sind beispielsweise Anpassungen der Menüs, eine Reduktion von selten benutzten Funktionen oder die Bereitstellung individueller Zusatzfunktionen denkbar. Oft genutzte Funktionalitäten sollen so effizienter zur Verfügung gestellt werden. Derartige Hilfestellungen und Anpassungen können dem Nutzer ein effizienteres und intuitiveres Benutzen der Software ermöglichen.

Beispiele für typische Anwendungsszenarien:

- Personalisierte GUI:
Auf Basis von Benutzerprofilen und Benutzerverhalten können Menüs einer Software im Laufe der Benutzung an die Präferenzen des Anwenders angepasst werden.
- Dynamische Menüs:
Die Menüs eines Anwenders passen sich daran, welche Elemente besonders häufig bzw. zuletzt verwendet wurden. Diese Art der Adaption findet man z.B. im Startmenü von Microsoft Windows XP.
- Favoriten:
Die Views können - nach Seitenaufrufen und Verweildauer sortiert - als Favoriten dargestellt werden.

3 Adaptionstechniken

3.1 Idee

Die Grundidee bei adaptiven Systemen besteht darin, dass das System den Benutzer bei seiner Arbeit „beobachtet“ und über ihn ein Benutzermodell bzw. Benutzerprofil anlegt. Dieses Vorgehen ist natürlich nur dann sinnvoll, wenn der Nutzer einen wesentlich längeren Zeitraum mit dem System interagiert, als das Erlernen des Nutzerverhaltens beansprucht.

3.2 Benutzermodell

Ein Benutzermodell wird in [SK] definiert als eine „Repräsentierung von Benutzereigenschaften, z.B. Annahmen, Interessen, Präferenzen, Skills, Aufgaben, Pläne, kognitive Stile“.

Diese Eigenschaften können entweder direkt vom Benutzer spezifiziert werden (Benutzerprofil) oder automatisch aus dem Benutzerverhalten abgeleitet werden. Das Benut-

zermodell basiert auf einer History von Benutzerkontexten, z.B. einer Liste der Benutzeraktionen.

3.3 Generische Benutzermodellierung durch Beobachtung von Handlungssequenzen

In vielen Projekten wurden bereits intelligente, adaptive Benutzerschnittstellen erforscht. Im jeweiligen Anwendungsgebiet wurden beeindruckende Ergebnisse erzielt. Die Übertragung auf andere Anwendungsfelder gestaltet sich meist jedoch äußerst kompliziert [AK04], denn viele der manuell modellierten Ansätze sind zu aufwändig und unflexibel. Es wurden daher einige generische Ansätze entwickelt, die von dem Anwendungsfeld unabhängig sind und allein aus den bisherigen Aktionen des Benutzers lernen können.

3.4 Aktionsvorhersage-Algorithmen

Mit einem Aktionsvorhersage-Algorithmus (AVA, engl. Action Prediction Algorithm) lassen sich die möglichen nächsten Benutzeraktionen samt der Wahrscheinlichkeiten prognostizieren. Die letzte Aktion des Benutzers dient dabei als Eingabe. Als Nachteil kann angesehen werden, dass viele Lernfälle benötigt werden, um die Modelle adäquat anlernen zu können.

AVA-basierte Ansätze bieten sich besonders für Mensch-Maschine-Systeme mit einem hohen Interaktionsanteil an (z.B. Interaktion auf Webseiten).

In [AK04] werden einige generische Ansätze in Form von sogenannten Aktionsvorhersage-Algorithmen vorgestellt.

- IPAM-Algorithmus:
Der IPAM-Algorithmus (Incremental Probabilistic Action Modeling) ist ein einfacher AVA, der gute Ergebnisse liefert und auf einem einfachen Markov-Modell basiert, welches durch ein spezielles Online-Lernverfahren aktualisiert wird. Der Algorithmus wurde 1998 von Davison und Hirsh [Dav98] in einer Studie eingeführt, in der es darum ging, UNIX-Kommandos vorherzusagen. Dabei wurden die Kommandos eines Nutzers aufgezeichnet. Wurde ein neues Kommando eingegeben, so erkannte der Algorithmus, ob dieses bereits früher ausgeführt wurde. Das nachfolgende Kommando in der Aufzeichnung würde mit einer gewissen Wahrscheinlichkeit so auch nun auf die Eingabe folgen. Dem Nutzer wurde die Vorhersage des folgenden Kommandos auf dem Bildschirm präsentiert. Mithilfe der Funktionstasten war es möglich, die Vorhersage anzunehmen, also das Kommando auszuführen. In durchschnittlich 3 von 4 Fällen wurde so das richtige Kommando vorhergesagt.
- PPM Algorithmen:
Der PPM-Algorithmus (Prediction by Partial Matching) basiert eigentlich auf einem Kompressionsalgorithmus (Ziv & Lempel 1977), der auch zur Vorhersage von Benutzeraktionen verwendet werden kann. Der Algorithmus versucht, eine Vorhersage aufgrund von exakten Übereinstimmungen mit Sequenzen aus den Trai-

ningsdaten zu treffen. Der Algorithmus ist ähnlich zum IPAM Algorithmus. Der Unterschied besteht darin, dass der IPAM-Algorithmus eine Ereignissequenz und kein einzelnes Ereignis zur Vorhersage nutzt.

- LEV-Algorithmus:

Der LEV-Algorithmus basiert auf der Ähnlichkeit von Sequenzen. Als Ähnlichkeitsmaß wird die Levenshtein-Distanz genutzt. Grundidee hierbei ist, auch dann gültige Vorhersagen zu treffen, wenn der Benutzer Fehler bzw. unnötige Schritte während der Interaktion mit dem System macht. Die Levenshtein-Distanz wird auch häufig zur Ähnlichkeitssuche in Suchmaschinen (z.B. Google) und Wörterbüchern genutzt.

4 XUL als deklarative Beschreibungssprache für Benutzerschnittstellen

Die XML User Interface Language, kurz XUL, ist eine auf XML basierende Beschreibungssprache für GUIs. XUL ist aus dem Open-Source-Projekt Mozilla mit dem Ziel entstanden, die Benutzerschnittstelle möglichst unabhängig von der Implementierung der Applikation definieren zu können.

Dieser Ansatz hat den Vorteil der einfacheren Wartbarkeit der Applikation.

Bekannte Mozilla-Anwendungen sind z.B. Netscape, Firefox und Thunderbird, bei denen die gesamte Benutzeroberfläche aus XUL-Dateien erzeugt wird. Für die Darstellung der Nutzeroberfläche kommt die sogenannte „Gecko Rendering Engine“ zum Einsatz. Mozilla bringt nicht nur Browserfunktionalitäten mit sich sondern ausserdem ein vollwertiges Framework zur Applikations-Entwicklung. XUL-Applikationen sind auf nahezu allen modernen Plattformen lauffähig und nutzen bestehende offene Standards wie CSS, JavaScript, DTD und RDF. Die Trennung von Layout und Design der Benutzeroberfläche ist ein weiterer Vorteil, der die Verwendung von Skins ermöglicht, wie es z.B. im Mozilla Firefox der Fall ist.

Microsoft verfolgt ein ähnliches Konzept mit dem kommenden Betriebssystem Windows Vista und nennt diese Sprache XAML.

Es gibt bereits einige adaptive XUL-Ansätze, wie z.B. die „Adaptive HomePage“-Erweiterung für den Firefox-Browser, welche automatisch die Startseiten-URL an das erlernte Surfverhalten des Nutzers anpasst.

Angenommen ein Nutzer prüft für gewöhnlich morgens seine E-Mails, nutzt Google am Mittag und besucht ein Forum am Abend - öffnet er zu diesen Zeiten dann den Browser, so wird die entsprechende Homepage sofort aufgerufen.

5 Anwendungsbeispiele für Adaptive Benutzerschnittstellen

Adaptive Benutzerschnittstellen lassen sich in fast allen Bereichen sinnvoll einsetzen, in denen eine regelmäßige Interaktion zwischen Mensch und Maschine stattfindet.

6 Zusammenfassung und Ausblick

Intelligente adaptive Websites beobachten das Verhalten des Anwenders und passen sich fortlaufend dementsprechend an. Auf kommerziellen Sites kann diese Art der Adaptivität dazu dienen, die Kundenbindung zu verstärken und das Kaufverhalten anzuregen.

Im Bereich der Medizin können adaptive Benutzerschnittstellen dazu dienen bestimmte Krankheiten zu diagnostizieren. Als Eingabe können z.B. Gewebeproben oder die Beschreibung der Krankheitssymptome dienen. Auf diese Art und Weise können auch Nebenwirkungen von bestimmten Medikamentenkombinationen vorhergesagt werden. Bei der Untersuchung können intelligente Techniken wie Künstliche Neuronale Netzwerke angewendet werden, um eine Diagnose zu machen.

In Finanz- und Börsensoftware können adaptive Algorithmen dazu genutzt werden, um Börsenkurse zu beobachten und Schwankungen zu prognostizieren und diese je nach Nutzergewohnheiten zu evaluieren. Bei Börsenmaklern und großen Börsen-Unternehmen ist diese Art von Adaptiver Software äußerst populär.

Auch in der Automobilindustrie könnten adaptive Nutzerschnittstellen eine immer größere Rolle spielen. So wäre es z.B. möglich, eine Sitz-Steuerung zu entwerfen, die sich automatisch an den jeweiligen Fahrer und dessen Gewohnheiten anpasst.

Navigationssysteme könnten sich automatisch an die Fahr-Gewohnheiten des Anwenders anpassen, und so ein individuelles Modell erstellen, um Prioritäten in der Wegplanung zu erkennen. Ein Ansatz eines solchen Systems wurde von Langley in [Lan96] mit dem „Adaptive Route Advisor“ vorgestellt. Die Ergebnisse des Experiments zeigten, dass das Benutzermodell nach ca. 12 Trainingsversuchen so weit angelernt war, dass das System mit einer Wahrscheinlichkeit von über 75% die Route wählte, die den Gewohnheiten des Nutzers entsprach.

In Lernsoftware sowie Computerspielen können adaptive Nutzerschnittstellen dazu dienen, den Schwierigkeitsgrad dem Nutzer anzupassen.

6 Zusammenfassung und Ausblick

In dieser Ausarbeitung wurde auf die Bedeutung und die Funktionsweise adaptiver Benutzerschnittstellen eingegangen. Das besondere an adaptiven Benutzerschnittstellen ist, dass sie automatisch die Präferenzen des Anwenders erkennen, und diese Daten in einem Benutzermodell speichern. Das Benutzermodell zusammen mit den Benutzeraktionen dient dazu, die Adaption an der Benutzerschnittstelle durchzuführen. So soll erreicht werden, dass der Anwender mit einer auf ihn angepassten (und nicht von ihm anpassbaren) Benutzerschnittstelle arbeitet.

Zur Definition von Benutzerschnittstellen existieren unterschiedliche Beschreibungssprachen. Die in dieser Ausarbeitung vorgestellte deklarative Beschreibungssprache XUL eig-

6 Zusammenfassung und Ausblick

net sich durch die Trennung von Layout und Design besonders gut zur Implementation von komplexen, adaptiven Nutzerschnittstellen.

Wir haben verschiedene generische Ansätze kennengelernt, um Verhaltensmuster bei Nutzern erkennen - und daraus zukünftige Aktionen ableiten zu können. Die Vorhersage des Nutzerverhaltens wird häufig mit Techniken der künstlichen Intelligenz erreicht.

Daraus ergeben sich gewisse Grenzen der Vorhersagegenauigkeit für adaptive Algorithmen und eine hohe Komplexität in der Programmierung.

Adaptive Nutzerschnittstellen sind durch den hohen Abstraktionsgrad und damit verbundenen Kostenfaktor nur selten in Softwaresysteme implementiert. Wie wir anhand der Beispiele gesehen haben, sind Ansätze in verschiedenen Programmen, Betriebssystemen (siehe Startmenü in Windows XP) und bei e-Commerce Unternehmen wie Amazon umgesetzt.

Das Potenzial adaptiver Benutzerschnittstellen ist enorm, denn Adaptivität kann zu einer höheren Nutzerakzeptanz, verbesserten Arbeitseffizienz und einer stärkeren Kundenbindung führen.

Die Entwicklung adaptiver Nutzerschnittstellen kann besonders von Forschungsergebnissen aus den Bereichen der Künstlichen Intelligenz profitieren. Da adaptive Algorithmen mit Daten von Benutzermodellen arbeiten, kann auch eine Verbesserung des User-trackings zu einer verbesserten Adaption beitragen. Neue Unaufdringliche Sensoren, beispielsweise hochauflösende Minikameras, eröffnen neue Möglichkeiten zur automatischen Erfassung von Nutzerdaten. Es ist beispielsweise ein System denkbar, das die Gefühlslage eines Anwenders erkennt, indem es den Puls misst oder die Augenbrauenbewegung analysiert, und dementsprechend weiterführende Hilfe anbietet.

Es ist zu erwarten, dass personalisierte, intelligente Schnittstellen zukünftig eine große Rolle bei der Entwicklung neuer Systeme spielen werden.

Literatur

- [AK04] ALEXANDER KÜNZER, FRANK OHMANN & LUDGER SCHMIDT: *Antizipative Modellierung des Benutzerverhaltens mit Hilfe von Aktionsvorhersage-Algorithmen*. MMI-Interaktiv Nr. 7 Juni 04, 2004.
- [Dav98] DAVISON, B. & HIRSCH, H.: *Predicting sequences of user actions*. Notes of the AAAI/ICML 1998 Workshop on Predicting the Future: AI Approaches to Time-Series Analysis, 1998.
- [Die93] DIETERICH, MALINOWSKI, KÜHME & SCHNEIDER-HUFSCHMIDT: *State of the Art in Adaptive User Interfaces*. Amsterdam, North-Holland, 1993.
- [Lan96] LANGLEY, PAT: *The Experimental Study of Adaptive User Interfaces*. 1996.
- [Lan97] LANGLEY, PAT: *Machine Learning for Adaptive User Interfaces*. Proceedings of the 21st German Annual Conference on AI, 1997.
- [SK] SOFTWAREKOMPETENZ.DE: *Software-Kompetenz.de Glossareintrag Benutzermodell*. <http://www.software-kompetenz.de/?15954>. Stand: Januar 2006.