

WEFEMIS - ein Werkzeug zur Evaluierung interaktiver Geräte*

N. Hamacher, J. Marrenbach
Lehrstuhl für Technische Informatik, RWTH Aachen
Ahornstr. 55, D-52074 Aachen
Tel: +49 241 80-26105
e-mail: {hamacher, marrenbach}@techinfo.rwth-aachen.de

Zusammenfassung

Die Entwicklung neuer Geräte wird heute weitgehend durch Rapid-Prototyping-Methoden und -Werkzeuge unterstützt. Diese Methoden und Werkzeuge zur Definition von Funktionalität und Verhalten eines technischen Gerätes erlauben die Aufdeckung von Schwachstellen im Entwurf. Zur Bewertung der Benutzungsfreundlichkeit schon in frühen Phasen des Entwicklungsprozesses existieren jedoch keine geeigneten Werkzeuge. In diesem Beitrag wird die Entwicklungsumgebung WEFEMIS (**W**erkzeug zur **f**ormalen und **e**mpirischen **E**valuierung **i**nteraktiver **S**ysteme) vorgestellt, mit dessen Hilfe es für den Entwickler möglich ist, interaktive Systeme durch formale Modelle zu spezifizieren, Softwareprototypen zu erstellen, zugehörige Benutzermodelle zu integrieren und das System mit Hilfe geeigneter Verfahren zu evaluieren. Die Erstellung des Prototypen sowie des Benutzermodells ist für den Entwickler nahezu ohne Programmieraufwand möglich. Die verschiedenen Modelle können grafisch editiert und miteinander verknüpft werden. Abschließend wird die Anwendung der Entwicklungsumgebung anhand eines Beispiels (CD-Spieler) verdeutlicht.

1 Einleitung

Bei der Entwicklung und Verbesserung von technischen Geräten spielen Prototypen eine wichtige Rolle. Sie erlauben eine frühe Evaluierung des endgültigen Produkts und können in Versuchen von Endbenutzern getestet werden.

Während die eigentliche Funktionalität eines Geräts mit Hilfe dieser Tests gut überprüft werden kann, ist die Qualität der Mensch-Maschine Kommunikation, also der Interaktion des Bedieners mit dem Gerät hinsichtlich Erwartungskonformität oder Steuerbarkeit, nicht leicht zu beurteilen.

Diese Arbeit stellt eine Entwicklungsumgebung vor, die die Erstellung von formalen Beschreibungen des Benutzerverhaltens und der Systemfunktionalität erlaubt. Derartige Spezifikationen der Mensch-Maschine Interaktion sind bereits in einem frühen Stadium des Entwicklungsprozesses möglich. Dadurch können bereits sehr früh Aussagen bzgl. der Benutzungsfreundlichkeit interaktiver Geräte getroffen werden, wodurch der finanzielle und zeitliche Aufwand für die Entwicklung deutlich reduziert werden kann.

2 Systemkonzept

* In: H. Oberquelle, R. Oppermann, J. Krause (Eds.): Berichte des German Chapter of the ACM. Mensch & Computer 2001, Volume 55, pp. 345-353, 5.-8. März, Bad Honnef (Bonn), Teubner Stuttgart, ISBN 3-519-02748-8

Für die Evaluierung von Mensch-Maschine Schnittstellen hinsichtlich der Gebrauchstauglichkeit werden im Wesentlichen drei Kriterien berücksichtigt. Nach ISO 9241-11 (ISO 1998) werden diese mit Effektivität, Effizienz und Zufriedenheit bezeichnet. Die Effektivität beschreibt die Genauigkeit und die Vollständigkeit, mit der ein Operateur ein bestimmtes Ziel oder Teilziel erreicht. Die Effizienz bezeichnet das Maß, das die Effektivität in das Verhältnis zum benötigten Aufwand setzt. Dabei ist zwischen menschlicher, zeitlicher und wirtschaftlicher Effizienz zu unterscheiden. Zufriedenheit beschreibt die Beeinträchtigungsfreiheit und die Akzeptanz des Operators bei der Bedienung des technischen Gerätes. Diese Kriterien legen den allgemeinen Bewertungsrahmen für die Untersuchung von Mensch-Maschine Systemen fest. Darüber hinaus werden zur weiteren Spezifikation Forderungen, wie sie z.B. in der ISO-Norm 9241 Teil 10 (ISO 1996) aufgeführt sind, berücksichtigt.

Unter Berücksichtigung der obengenannten Kriterien wird eine Umgebung entwickelt, die es dem Entwickler ermöglicht, ein technisches Endgerät frühzeitig hinsichtlich den ergonomischen Anforderungen operationalisiert zu evaluieren (Marrenbach 2000). Basierend auf einer Systemspezifikation wird zunächst ein Benutzermodell auf Grundlage der GOMS-Theorie erstellt. Ausgehend von diesem Modell kann eine Analyse der zu erwartenden Gebrauchstauglichkeit durchgeführt werden. Als Voraussetzung werden für alle Bestandteile des GOMS-Modells Zeiten festgelegt, die für die Anwendung der entsprechenden Selektionsregeln, Operatoren und Methoden erwartet werden können. Darüber hinaus besteht die Möglichkeit, ein Handbuch für das interaktive System zu generieren. Auf Basis der Spezifikation kann des weiteren ein Prototyp erstellt werden, mit dem Versuchspersonen oder Endanwendern interagieren können. Dabei werden automatisch Handlungsprotokolle erstellt, durch deren Auswertung Rückschlüsse auf die Bedienbarkeit des Prototypen, wie z.B. Lernzeiten, Bearbeitungszeiten und Fehlerraten, gezogen werden können.

Beim Entwurf und der Implementierung der Entwicklungsumgebung wird das Ziel verfolgt, soweit wie möglich vorhandene Werkzeuge für die Schritte des Entwurfs- und Prototyping-Prozesses zu verwenden. Noch nicht vorhandene Komponenten werden als Erweiterungsmodule zu bestehenden Systemen konzipiert und implementiert, um die Anzahl der verwendeten Werkzeuge so gering wie möglich zu halten.

3 Entwurf der Entwicklungsumgebung

In diesem Abschnitt werden die Komponenten der Entwicklungsumgebung und ihre Verbindungen untereinander beschrieben. Wie bereits erläutert, besteht das System aus unterschiedlichen Programmen, die miteinander kommunizieren und als Gesamtheit die Erstellung eines simulierten, lauffähigen Prototypen erlauben. Abbildung 1 zeigt die einzelnen Systemkomponenten der Entwicklungsumgebung.

Im Rahmen der Entwicklungsumgebung wird von einem Softwareprototypen ausgegangen. Die Benutzungsschnittstelle wird als grafische Benutzungsoberfläche realisiert. Das hat den Vorteil, dass die Erstellung schnell durchgeführt werden kann und Änderungen am Entwurf später leicht möglich sind. Jedoch ist die Bedienung von Softwareprototypen nicht immer vollständig mit entsprechenden Hardwarelösungen vergleichbar. Für eine erste Evaluierung zum Test der generellen Handhabung eines Gerätes sind solche Versuche indes ausreichend.

Nach dem Seeheim-Modell (Green 1984) kann das Gerätemodell in die drei Komponenten Anwendungsschnittstelle, Dialogkontrolle und Präsentationskomponente aufgeteilt werden. Die Präsentationskomponente ist der Teil eines Gerätes, den der Benutzer als Erscheinungsbild erkennt und über den die Bedienung erfolgt. Dazu gehören alle Eingabelemente, mit deren Hilfe der Benutzer das Gerät steuert, sowie Elemente zur Visualisierung von Nutzerinformationen. Die Dialogkontrolle steuert die Präsentationskomponente. Sie stellt Daten zur Visualisierung bereit und reagiert auf Benutzereingaben. Während die Präsentationskomponente lediglich auf direkte Kommandos des Benutzers reagiert, besitzt die Dialogkontrolle zusätzliches Wissen über den Zustand des Gerätes und entscheidet, wie auf eine Aktion zu reagieren ist. Die Anwendungsschnittstelle bindet die Dialogkontrolle an die Programmlogik an. Sie legt fest, welche Funktionalität der Dialogkontrolle und darüber auch dem Benutzer zur Verfügung gestellt wird. Die Dialogkontrolle nutzt die Anwendungsschnittstelle, um Funktionen aufzurufen, die entsprechend den Eingaben des Benutzers Aktionen ausführen.

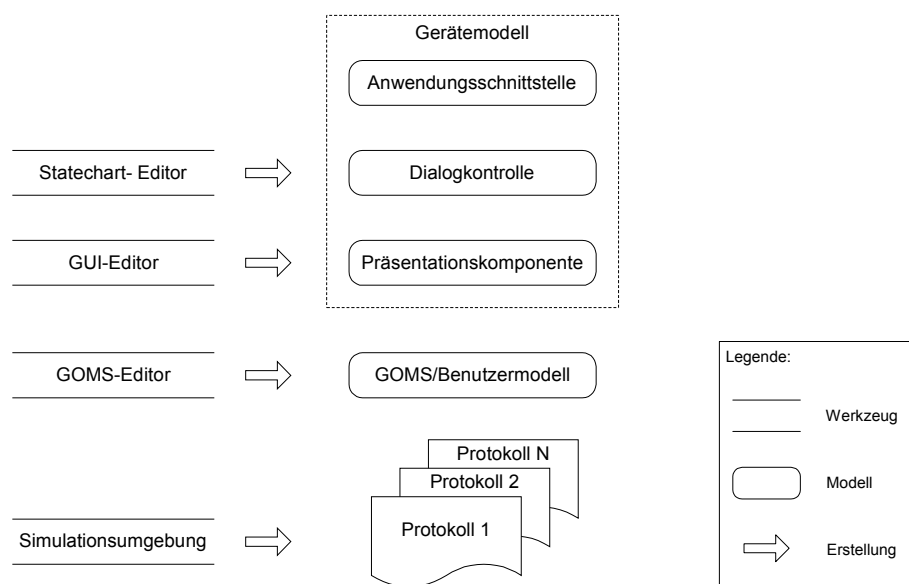


Abb. 1: Systemkomponenten der Entwicklungsumgebung WEFEMIS

Zur Gewinnung quantitativer Aussagen über Systemkomplexität und Modellkompatibilität werden, insbesondere im Hinblick auf die Werkzeugunterstützung, formale Beschreibungen der zu untersuchenden Objekte benötigt. Zur Formalisierung des Benutzermodells eignet sich unter anderem die GOMS-Modellierung, da GOMS sehr einfach und effizient in der Handhabung ist (John 1997). So misst sich der z.B. Aufwand zum Erlernen des Formalismus für Ingenieure ohne psychologische Fachkenntnisse lediglich in wenigen Tagen (Grey 1999). GOMS steht für die verschiedenen Komponenten des Modells, nämlich Goals, Operators, Methods und Selection Rules. Ziele charakterisieren die Zustände, die der Benutzer erreichen will. Die Operatoren sind die Grundfunktionen des Systems, d.h. die elementaren Interaktionstechniken. Methoden sind Folgen von Operatoren, die nacheinander angewendet werden, um ein Ziel zu erreichen. Die Auswahlregeln werden verwendet, wenn es mehrere Methoden zum Erreichen eines Zieles gibt.

3.1 Erstellung der Benutzermodelle

Das Analyse-Werkzeug soll anhand formaler Spezifikationen (Benutzermodell-Analysen) Parameter zur vergleichenden Bewertung verschiedener Entwurfsalternativen liefern. Der Schwerpunkt liegt dabei in der Analyse des dem Entwurf jeweils zugrunde liegenden Aufgabenmodells. Diese GOMS-Analyse ist zur summativen Evaluation von Systemwürfen und Prototypen geeignet. Die durch diese Analyse gewonnenen Daten können zur vergleichenden Bewertung verschiedener Alternativkonzepte herangezogen werden. Die Analyse liefert eine Vorhersage der Ausführungs- und Lernzeiten der Benutzermodelle auf Basis der Modellierung nach NGOMSL (Kieras 1988). Ferner wird eine Komplexitäts-Analyse vorgenommen. Dabei wird die Komplexität einer Aufgabe anhand der Tiefe der hierarchischen Zielstruktur, der Anzahl der Schritte (Teilziele und Operatoren) einer Methode und der Anzahl der unterschiedlichen Methoden in einem Aufgabenbaum abgeschätzt (Kieras 1988).

Den Benutzermodell-Editor zeigt Abbildung 2. Im rechten Bereich können die Ziele, Methoden und Operatoren ausgewählt bzw. eingegeben werden. Diese Strukturen werden im mittleren Bereich in Form eines Baumes dargestellt. Die Ergebnisse der Analyse lassen sich detailliert darstellen, z.B. wie häufig werden einzelne Operatoren verwendet oder wieviel Zeit wird für die gesamte Aufgabe benötigt. Bei der Analyse wird zwischen sensorischen, mentalen und motorischen Operatoren unterschieden. Für diese Operatoren werden die jeweilige Anzahl und die benötigte Zeit berechnet und grafisch dargestellt (Abbildung. 3). Ferner erfolgt eine Berechnung der Gesamtzahl der Operatoren und der entsprechende Zeitbedarf.

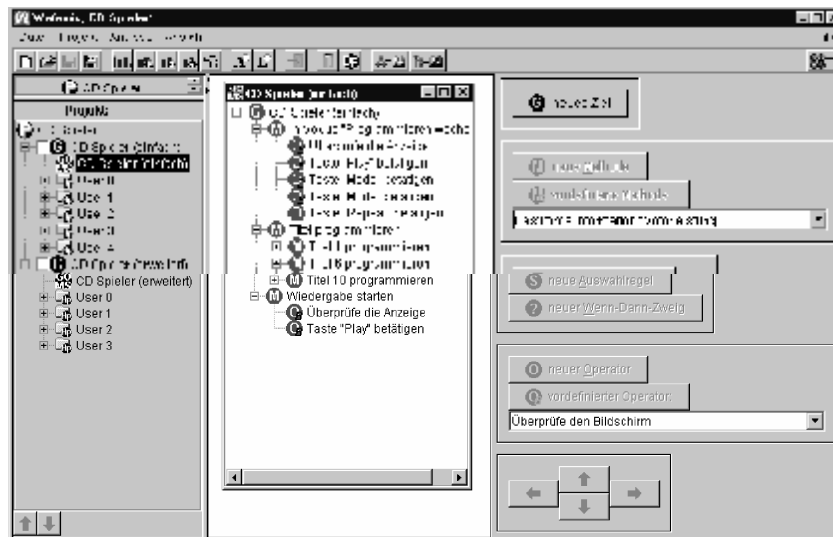


Abb. 2: Benutzermodell-Editor mit einer Beispielanwendung

3.2 Dialogkontrolle

Für die Spezifikation der Dialogkontrolle werden Zustandsübergangsdiagramme, sogenannte Statecharts, verwendet. Statecharts wurden von Harel als Erweiterung der Endlichen Automaten zur Spezifizierung von Gerätezuständen entwickelt (Harel 1987). Während Endliche Automaten nur aus gleichwertigen Zuständen und Übergängen bestehen, erlauben Statecharts die hierarchische Ineinanderschachtelung von Automaten und Zuständen. Zustände werden durch Übergänge miteinander verbunden, die den Aktionen des

Benutzers oder anderen Ereignissen entsprechen und dadurch eine Veränderung des Betriebszustandes des Gerätes bewirken. Dabei können Übergänge von jeder beliebigen Hierarchiestufe in jede andere erfolgen.

Zur Generierung der Zustandsübergangsdiagramme wird das Programm Statemate der Firma i-Logix eingesetzt. Aus einem vollständig spezifizierten Statechart-Modell kann Statemate Code u.a. in der Sprache C erzeugen, der als Kontrollkomponente in einen Prototypen übernommen werden kann. Mit Hilfe der Statecharts können sowohl einfache technische Geräte als auch sehr komplexe Systeme modelliert werden (Marrenbach 1998).

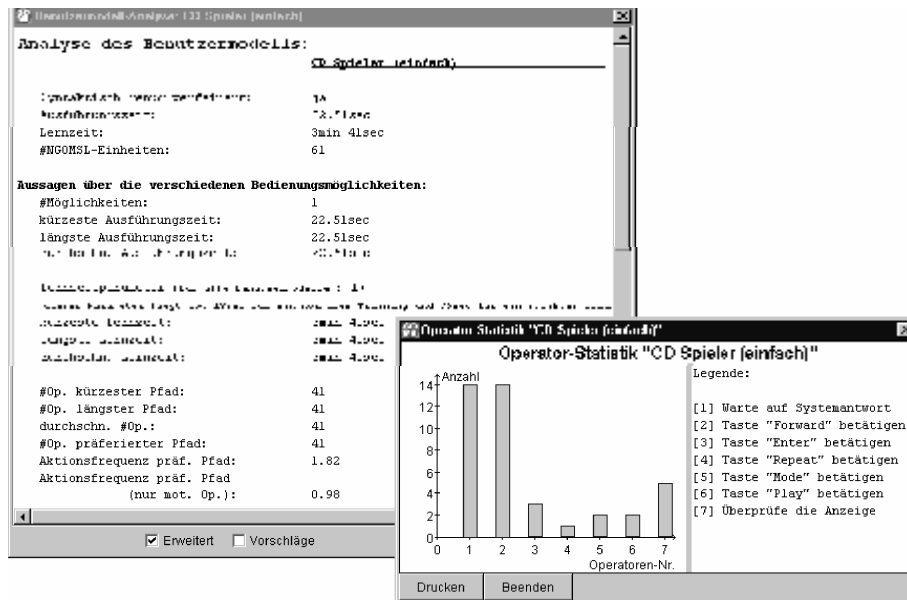


Abb. 3: Benutzermodell-Analyse - Übersicht einer Beispielanwendung

3.3 Erstellung der Benutzungsoberfläche

Als Interaktionspunkt mit dem Benutzer sieht das Seeheim-Modell die Präsentationskomponente vor. Sie enthält Elemente, die den Benutzer über den aktuellen Zustand des Gerätes informieren und Eingaben entgegennehmen. Aktionen werden an die Dialogkontrolle weitergereicht, Rückmeldungen an den Benutzer kommen ebenfalls aus dieser Komponente. Das entwickelte Werkzeug ermöglicht dem Entwickler, die grafische Benutzungsoberfläche (GUI) des Prototypen zu erstellen und zu bearbeiten. Die Erstellung einer GUI ist eine häufig durchgeführte Aufgabe, die von den meisten Software-Entwicklungsumgebungen, so auch Statemate, unterstützt wird.

3.4 Auswertung der Handlungsprotokolle

Die experimentelle Auswertung erfolgt mittels eines Handlungsinterpreters, der ähnlich wie ein Kommandosprachen-Interpreter Benutzeraktionen mit der zu Grunde liegenden Syntax, dem GOMS-Modell, vergleicht. Unerwartete Benutzeraktionen werden als Fehler interpretiert wobei anschließend eine geeignete Visualisierung der Fehlerraten erfolgt. Das System erlaubt ferner die gleichzeitige Auswertung von Protokollen mehrerer Benutzer.

3.5 Vorgehen bei der Benutzung von Wefemis

Der Einsatz des Evaluierungswerkzeugs ist unabhängig von der Entwicklung des Endgerätes und kann daher entwicklungsbegleitend in verschiedenen Phasen der Systementwicklung stattfinden. Die Benutzermodell-Analyse ist somit bereits durchführbar, wenn noch keine Spezifikation des zu evaluierenden Systems vorhanden ist. Die Spezifizierung des Gerätes durch den Entwicklungsingenieur dient als Grundlage für die darauf folgende Realisierung. Diese Spezifikation bildet zusätzlich den Ausgangspunkt für die Erstellung eines Prototypen, der von Versuchspersonen in empirischen Untersuchungen getestet werden kann. Wie bereits beschrieben lassen sich aus diesen Tests Handlungsprotokolle gewinnen, die für eine vergleichende Bewertung mit den Benutzermodellen ebenfalls in das Analyse-Modul integriert und analysiert werden können.

4 Anwendungsbeispiel

Im Folgenden wird anhand eines Beispiels die Anwendung der Entwicklungsumgebung WEFEMIS verdeutlicht. Um die Interaktion mit dem Benutzer und die Möglichkeiten des Benutzermodells besser darstellen zu können, wird ein Gerät modelliert, das einige Interaktionspunkte mit dem Benutzer aufweist, jedoch in seinem technischen Aufbau nicht allzu komplex ist. Es handelt sich dabei um einen CD-Spieler, für den zwei unterschiedliche Benutzungsoberflächen und die zugehörigen Benutzermodelle realisiert werden.

4.1 Gerätemodell

Der CD-Spieler bietet verschiedene Betriebsmöglichkeiten. Neben der Wiedergabe einer ganzen CD können auch einzelne Titel in beliebiger Reihenfolge programmiert und abgespielt werden. Darüber hinaus kann ein Titel alleine abgespielt werden. In allen Abspielmodi ist es möglich, eine wiederholte Wiedergabe zu wählen. Das Statechart des CD-Spielers ist in Abbildung 4 dargestellt. Dieses Statechart modelliert die einzelnen Zustände des CD-Spielers und die möglichen Übergänge.

Beim Anschalten durch die PLAY-Aktion wird die Wiedergabe unmittelbar gestartet. Der Benutzer kann dann noch andere Modi anwählen oder Titel programmieren. Die STOPP-Aktion schaltet das Gerät zurück in den STANDBY-Modus, wobei die Anzeige des Gerätes nur während der Wiedergabe leuchtet. Es kann weiterhin zwischen einfacher Wiedergabe und wiederholtem Abspielen gewählt werden. Eine weitere parallele Kontrolle steuert den PAUSE-Modus. Die Moduswahl und die Wiederholungssteuerung wird mit einem gedächtnisbehafteten Zustand modelliert, so dass auch im ausgeschalteten Zustand (STANDBY) die aktuelle Einstellung beibehalten wird. Ist das Gerät im PAUSE-Modus, so geht diese Zustandsinformation beim Wechsel in den STANDBY-Modus jedoch verloren.

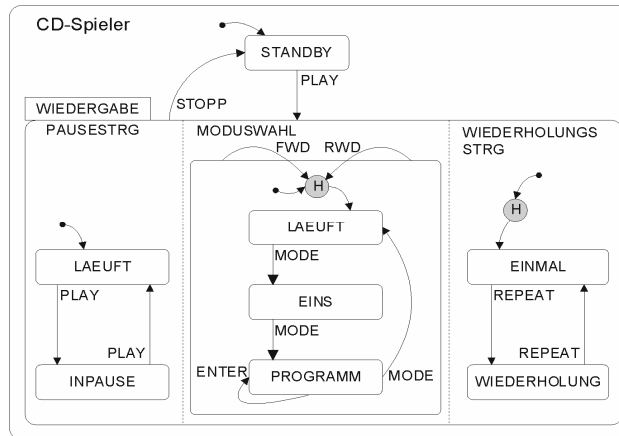


Abb. 4: Statechart zum CD-Spieler (Leuker & Marrenbach 1998)

4.2 Benutzungsoberfläche

Die Benutzungsoberflächen der beiden CD-Spieler unterscheiden sich im wesentlichen hinsichtlich der Menge der zur Verfügung stehenden Tasten zur Steuerung der Funktionalität der Prototypen.

Beim Prototyp 1, dargestellt in Abbildung 5 (links), sind die Schaltflächen mit den Anfangsbuchstaben der Aktionen STOP, ENTER, MODE und REPEAT beschriftet. Die gängigen Symbole für Abspielen, Vor- und Zurückspulen werden ebenfalls direkt auf die Schaltflächen geschrieben. Das Display besteht aus Anzeigen für die Modi sowie einem Feld, welches beliebige Texte darstellen kann, z.B. die Spielzeit oder die Titelanzeige. Im Gegensatz zum ersten Prototypen stellt der erweiterte Prototyp (Abbildung 5, rechts) dem Anwender ein numerisches Tastenfeld zur Verfügung über das Eingaben, z.B. zur Programmierung von Titeln, gemacht werden können. Darüber hinaus sind die Schaltflächen mit den Funktionen beschriftet. Ein Display steht ebenfalls zur Verfügung, um Spielzeit und Titelnummer anzuzeigen.

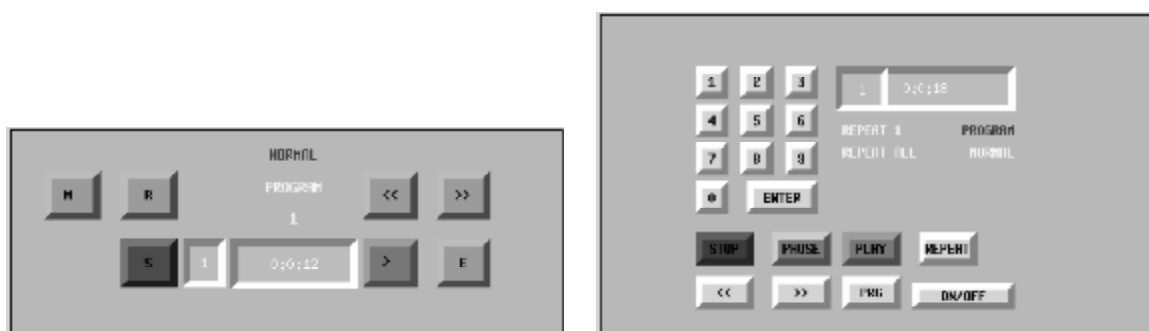


Abb. 5: Benutzungsoberfläche eines einfachen CD-Spielers (links: Prototyp 1) und eines erweiterten CD-Spielers (rechts: Prototyp 2)

4.3 Benutzermodell

Die Prototypen bieten eine Reihe von Möglichkeiten, um die Bedienung zu simulieren.

Beispielhaft werden die Benutzermodelle für das Szenario "Programmierung von Titeln" erzeugt. Die Aufgabe besteht jeweils darin, die Titelfolge 1, 6 und 10 zu programmieren und anschließend das Programm abzuspielen. Für den Prototyp 1 ergibt sich das in der folgenden Abbildung (Abbildung 6) aufgezeigte linke Benutzermodell während für den erweiterten Prototypen das rechte Modell in Abbildung 6 dargestellt ist. Abbildung 7 zeigt die nebeneinandergestellte Benutzermodell-Analyse der jeweiligen Prototypen.

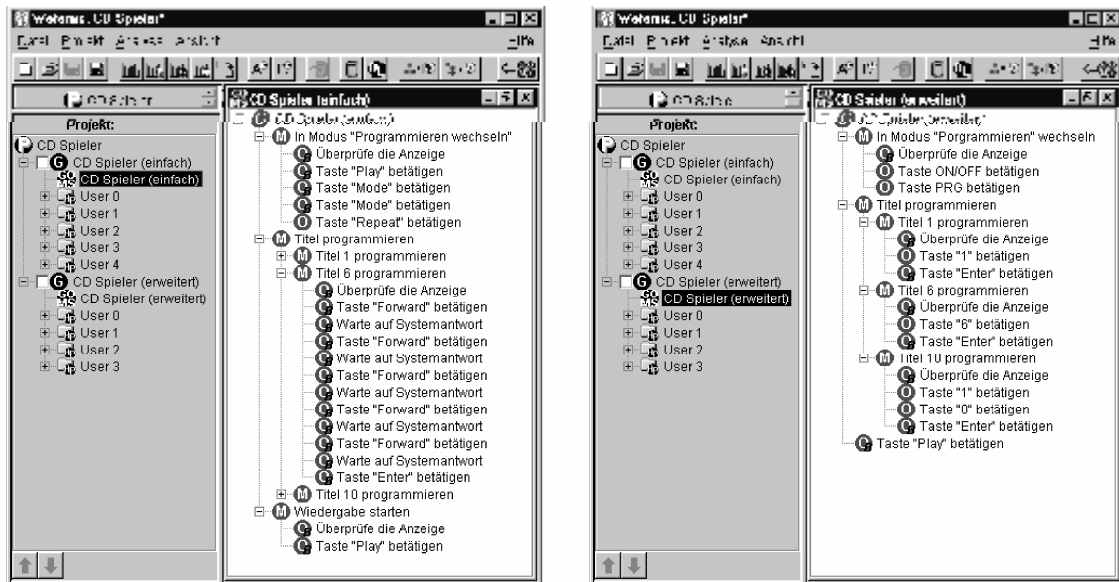


Abb. 6: Benutzermodelle des einfachen (links) und des erweiterten (rechts) CD-Spielers

	CD Spieler (einfach)	CD Spieler (erweitert)
Analyse der Benutzermodelle:		
Syntaktisch genug verfeinert:	ja	ja
Ausführungszeit:	22,51sec	8,65sec
Lernzeit:	3min 41sec	3min 41sec
#NGOMS-Einheiten:	61	31
Aussagen über die verschiedenen Bedienungsmöglichkeiten:		
#Möglichkeiten:	1	1
kürzeste Ausführungszeit:	22,51sec	8,65sec
längste Ausführungszeit:	22,51sec	8,65sec
durchschn. Ausführungszeit:	22,51sec	8,65sec
Ausführungszeit-Diagramm		
Lernzeitparameter (für alle Benutzermodelle): 17		
(Dieser Parameter liegt zw. 17sec für ein normales Training und 35sec für ein striktes Training.)		
kürzeste Lernzeit:	3min 41sec	3min 41sec
längste Lernzeit:	3min 41sec	3min 41sec
durchschn. Lernzeit:	3min 41sec	3min 41sec
Lernzeit-Diagramm		
#Op. kürzester Pfad:	41	14
#Op. längster Pfad:	41	14
durchschn. #Op.:	41	14
#Op. präferierter Pfad:	41	14
Aktionsfrequenz präf. Pfad:	1,82	1,62
Aktionsfrequenz präf. Pfad (nur not. Op.):	0,98	1,16
Anzahl der Komponenten im Benutzermodell:		
Methoden:	6	5
Operatoren:	41	14
Auswahlregeln:	0	0
WennDann-Bez.:	0	0
Summe:	47	19

Abb. 7: Benutzermodell-Analyse der CD-Spieler

Basierend auf dem GOMS-Modell kann für das Szenario für den Prototypen 1 eine mittlere Bearbeitungszeit von 22,51 Sekunden berechnet werden. Für Prototyp 2 ergibt sich eine Ausführungszeit von 8,65 Sekunden. Diese Ergebnisse werden im Rahmen einer Ver-

suchsreihe überprüft. Weiterhin ergibt die Analyse, dass die Anzahl der notwendigen Operatoren für den Prototypen 2 mit 14 mehr als 60 % niedriger als für Prototyp 1 (41) liegt.

4.4 Experimentelle Überprüfung der Benutzermodelle

Inwieweit die Benutzermodelle und die entwickelten Benutzungsoberflächen erwartungskonform sind, kann erst nach Durchführung einiger Experimente beurteilt werden. Im Rahmen einer empirischen Untersuchung standen insgesamt 6 Probanden pro Prototyp zur Verfügung. Jeder Proband führte das Szenario 25 mal durch (Maaßen 1998). Die analytisch gewonnenen Daten werden mit den Ergebnissen, die sich aus den Handlungsprotokollen ermitteln lassen, verglichen. Das Werkzeug ermöglicht hierfür zum einen die automatische Auswertung der Handlungsprotokolle und zum anderen die grafische Darstellung der Ergebnisse (Hamacher 2000). Beispielhaft wird die Analyse anhand der Bearbeitungszeiten vorgestellt.

Die folgende Grafik (Abbildung 8) zeigt die ermittelten Werte der Ausführungszeiten im Vergleich zur analytisch berechneten Ausführungszeit, die als durchgezogene Linie in dem Diagramm visualisiert ist. Der grau hinterlegte Balken markiert einen Toleranzbereich von $\pm 20\%$. Es ist deutlich erkennbar, dass die vorhergesagten Ausführungszeiten mit den experimentell ermittelten Werten gut übereinstimmen. Zusätzlich zeigt diese Analyse, dass der erweiterte CD-Spieler (Prototyp 2) einfacher und schneller zu bedienen ist, als der vergleichbare Prototyp 1. Die Bewertung der Komplexität bestätigt diese Aussage, da für die Bearbeitung der Aufgabe mit dem Prototypen 1 mehr Benutzeraktionen erforderlich sind als mit Prototyp 2 (vgl. Abbildung 6 und Abbildung 7).

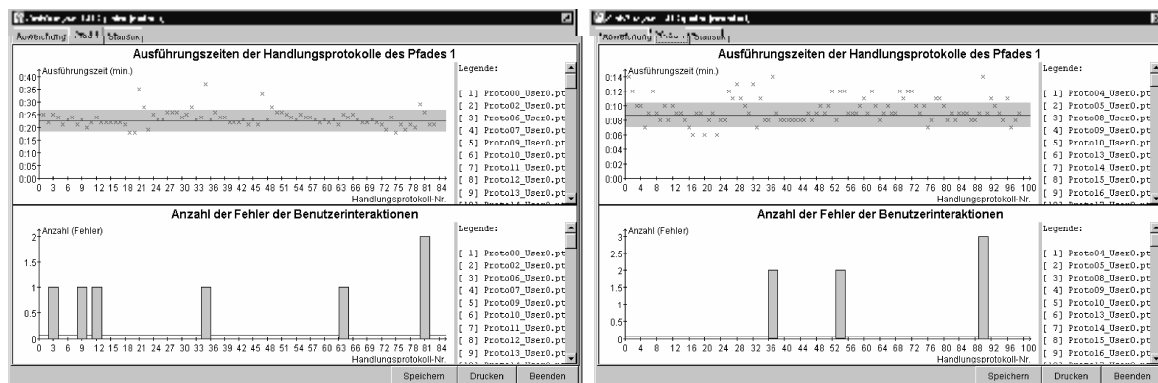


Abb. 8: Vergleich der Ausführungszeiten der jeweiligen Prototypen (analytisch vs. experimentell)

Dieses einfache Beispiel zeigt, dass die Modellierung des Benutzerverhaltens basierend auf der NGOMSL-Theorie geeignet ist, interaktive Systeme frühzeitig in der Entwicklungsphase mit der Entwicklungsumgebung formal bewerten zu können.

5 Zusammenfassung und Ausblick

In dieser Arbeit wurde die Entwicklungsumgebung WEFEMIS vorgestellt, die sich aus vorhandenen Werkzeugen (Statemate) und entwickelten Erweiterungen (Benutzermodell-

Editor, Analysewerkzeug) zusammensetzt. Die Umgebung bietet nach der Aufgabenanalyse und der Spezifikation der Funktionalität eine durchgängige Unterstützung bei der Erstellung der Dialogkontrolle und einer grafischen Benutzungsoberfläche entsprechend dem Seeheim-Modell sowie eines normativen Benutzermodells basierend auf einem GOMS-Modell. Gleichzeitig ermöglicht die Entwicklungsumgebung die formale Evaluierung der Gebrauchstauglichkeit des spezifizierten Systems bzgl. Ausführungszeit und Fehlerraten.

Die Dialogkontrolle wird durch Statecharts in einem grafischen Editor spezifiziert. Die Erstellung und Bearbeitung des Benutzermodells geschieht mit Hilfe eines Editors, der darüber hinaus charakteristische Größen, wie Ausführungszeit und Komplexität automatisch berechnet.

Bei der Interaktion von Versuchspersonen mit dem Prototypen werden automatisch Handlungsprotokolle erzeugt, aus denen die Zeiten und Aktionen einer Bedienung hervorgehen. Diese Protokolle können mit dem Benutzermodell verglichen werden, um die prognostizierten Ausführungszeiten zu validieren und typische Fehlbedienungen zu entdecken. Die Einsatzmöglichkeiten der Entwicklungsumgebung wurden anhand einer beispielhaften Anwendung (CD-Spieler) vorgestellt.

Zukünftig sind Erweiterungen des Systems geplant. Diese betreffen im Wesentlichen die weitere Automatisierung des Entwicklungs- und Bewertungsprozesses.

6 Literaturverzeichnis

- Card, S., Moran, T.; Newell, A. (1983): The psychology of human computer interaction. Lawrence Erlbaum.
- Gray, W.D., Boehm-Davis, D., John, B.E., Kieras, D.E. (1999). Cognitive analysis of dynamic performance: Cognitive process analysis and modeling. Department of Psychology at George Mason University
- Green, M. (1984): Report on Dialogue Specification Tools. In: Computer Graphics Forum. Vol. 3 1984, pp. 305-313.
- Harel, D. (1987): Statecharts: A Visual Formalism for Complex Systems. In: Science of Computer Programming. Amsterdam: North Holland. Vol. 8, pp. 231-274.
- ISO 9241 Part 11 (1998): Ergonomic requirements for office work with visual display terminals - Guidance on usability. International Organization for Standardisation, Genf.
- ISO 9241-10 (1996): Ergonomic requirements for office work with visual display terminals - Dialogue principles. International Organisation for Standardisation, Genf.
- John, B.E., Kieras, D.E. (1997). Using GOMS for user interface design and evaluation: Which technique?. In: ACM Transactions on Computer--Human Interaction. Vol. 3, pp. 287-319.
- Kieras, D. (1988): Towards a practical GOMS model technology for user interface design. In: M. Helander (ed.), Handbook of human-computer interaction. Amsterdam: North Holland, pp. 135-157.
- Hamacher, N. (2000): Entwicklung und Implementierung eines Werkzeugs zur Bewertung interaktiver Systeme basierend auf normativen Benutzermodellen. Lehrstuhl für Technische Informatik, Diplomarbeit, RWTH Aachen.
- Leuker, S.; Marrenbach, J. (1998): Integrating User Models into Device Prototype Design. In: 24th Conference of the IEEE Industrial Electronics Society IECON. Aachen, Germany, pp. 2532-2534.
- Maaßen, D. (1998): Untersuchung zur Ermittlung der Gebrauchstauglichkeit von Endgeräten. Lehrstuhl für Technische Informatik, Diplomarbeit, RWTH Aachen.
- Marrenbach, J. (2000): Konzept eines Werkzeugs zur formalen und empirischen Evaluierung der Gebrauchstauglichkeit interaktiver Endgeräte. In: Timpe, K.-P.; Willumeit, H.-P.; Kolrep, H. (Hrsg.): Bewertung von Mensch-Maschine-Systemen. Düsseldorf: VDI. S. 283-296.