

Einsatz formaler Methoden zur Evaluierung der Gebrauchsfähigkeit interaktiver Geräte

Utilization of Formal Methods for the Usability Evaluation of Interactive Devices

Nico Hamacher, Karl-Friedrich Kraiss, RWTH Aachen
 Jörg Marrenbach, AUDI AG, Ingolstadt

Gebrauchsfähigkeit ist ein wichtiges Attribut interaktiver Systeme. Sie kann empirisch erst ermittelt werden, wenn ein funktionsfähiger Prototyp zur Verfügung steht. Da sich die Entwicklungsprozesse ständig beschleunigen, kommen empirische Ergebnisse jedoch häufig zu spät, um noch in den Entwicklungsprozess einfließen zu können. Dieser Beitrag beschreibt als Alternative den Einsatz formaler Bewertungsmethoden auf Basis normativer Benutzermodelle. Diese erlauben die Bewertung der Gebrauchsfähigkeit bereits auf der Grundlage technischer Spezifikationen. Es wird das Werkzeug TREVIS vorgestellt, das normative Benutzermodelle semi-automatisch aus technischen Spezifikationen generiert und daneben vielfältige Analysemöglichkeiten bereit stellt.

Usability is an important attribute of interactive systems. It can only be examined empirically as soon as a functioning prototype is available. Since the development process is continuously accelerating, empirical results often come too late and cannot be used in the current process. In this contribution a formal evaluation method using normative user models is presented as an alternative. It allows an evaluation based on technical specifications. The tool TREVIS is introduced, which generates normative user models semi-automatically from such specifications, and additionally provides various options for data analysis.

1 Einleitung

Bei der Entwicklung interaktiver Geräte spielen Begriffe wie *Ergonomie* und *Gebrauchsfähigkeit*¹ eine immer wichtigere Rolle. Während der Fokus der Entwicklung Anfang der neunziger Jahre hauptsächlich auf der Realisierung der geforderten Funktionalität lag, ergibt sich heutzutage eine erhöhte Anforderung hinsichtlich einer einfachen und intuitiven Bedienbarkeit. Dies ist vor allem dadurch begründet, dass Kunden die Produkte, mit denen sie nicht zufrieden sind, in immer größerem Maße unberücksichtigt lassen [10]. Aufgrund des wachsenden Wettbewerbsdrucks verwenden inzwischen einige Firmen die *hohe* Gebrauchsfähigkeit ihrer Produkte als Werbe- und Verkaufsargument.

¹ In der Praxis werden die Begriffe *Benutzungsfreundlichkeit*, *Gebrauchstauglichkeit*, *Benutzerfreundlichkeit*, *Benutzbarkeit* und *Usability* häufig parallel und synonym verwendet, wenn es um die ergonomische Gestaltung interaktiver Geräte geht. In dieser Arbeit wird ausschließlich von Gebrauchsfähigkeit gesprochen.

Die übliche empirische Bewertung der Gebrauchsfähigkeit hat gravierende Nachteile. Sie erfordert einen Prototypen, kostet Zeit und Geld und liefert die Ergebnisse meist so spät, dass die laufende Entwicklung keinen Nutzen mehr daraus ziehen kann.

Immer kürzere Entwicklungszyklen haben dazu geführt, dass die Geräteentwicklung vermehrt durch Entwicklungswerkzeuge unterstützt wird. Hierzu gehören Programme zur Erstellung von Gerätespezifikationen, die z. B. die automatische Generierung von Programmcode ermöglichen. Dieses ohnehin vorhandene technische Know-How kann die Grundlage für formale ergonomische Analysen bilden und damit eine Synergie zwischen technischen und ergonomischen Bewertungsanstrengungen bewirken.

Dieser Beitrag beschreibt im Einzelnen die skizzierte Vorgehensweise und stellt ein Werkzeug zur formalen Bewertung der Gebrauchsfähigkeit auf der Grundlage technischer Spezifikationen vor.

2 Entwicklung und Bewertung interaktiver Geräte in verschiedenen Lebensphasen

Der Werdegang eines Gerätes gliedert sich in verschiedene voneinander abgrenzbare Lebensphasen (Bild 1). Zu Beginn wird in der Definitionsphase ein allgemeines Systemkonzept ausgearbeitet. Auf Basis dieses Konzepts sind in der Spezifikations- und Entwurfsphase die notwendigen (Unter-)Systeme und Funktionen zu definieren. In der anschließenden Entwicklungsphase erfolgt die detaillierte Realisierung der einzelnen Systemkomponenten, die in der Integrationsphase zu dem funktionsfähigen Gesamtsystem zusammengesetzt werden. Die abschließende Nutzungsphase stellt den Einsatz des Gerätes entsprechend des konzipierten Verwendungszwecks dar.

Die Einteilung in Lebensphasen ermöglicht eine genaue Zuordnung von Entwicklungswerkzeugen und Evaluierungsmethoden, die ebenfalls in Bild 1 dargestellt ist und im Folgenden erläutert wird.

2.1 Entwicklung des Gerätes

Bei der Entwicklung eines interaktiven Gerätes kommen in den frühen Lebensphasen Werkzeuge zur Spezifikation der Gerätefunktionalität und später zur Realisierung dieser Funktionalität zum Einsatz.

Spezifikation der Gerätefunktionalität

Die Spezifikation interaktiver Geräte wird heute noch häufig manuell mit Hilfe eines Pflichten- bzw. Lastenheftes durchgeführt. In den letzten Jahren kommen jedoch immer mehr automatisierte Spezifikationswerkzeuge zum Einsatz, die den Vorteil haben, dass die Eigenschaften des Gerätes eindeutiger festgelegt und teilweise automatisch ausgewertet werden können. Beispielhaft seien folgende Spezifikationswerkzeuge genannt:

- *SDL* (functional Specification- and Description-Language) ist eine Sprache zur Spezifikation und Beschreibung von Systemen [7]. Die Grundidee von SDL ist es, ein System in Form von kommunizierenden Prozessen zu beschreiben. Ein Prozess ist ein endlicher Automat und kann mit anderen parallel existierenden Prozessen über Verbindungswege mit Hilfe von Signalen kommunizieren.

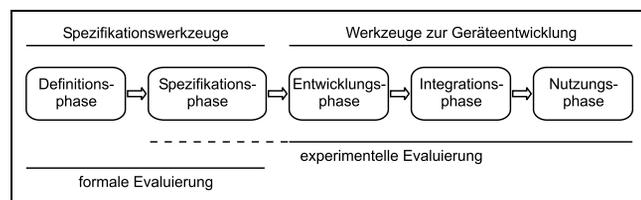


Bild 1: Die Lebensphasen eines interaktiven Gerätes.

- Statecharts bieten eine Hilfe zur Modellierung von Abläufen und Aktionen von Systemen. Ein Beispiel ist das Werkzeug *Statemate* der Firma i-Logix Inc., welches auf den von Harel entwickelten Statecharts aufbaut [6]. Statecharts erweitern endliche Automaten um die Möglichkeit der hierarchischen 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. Eine automatische Generierung von Programmcode aus den Statecharts ist möglich.
- Der Standard *UML* (Unified Modeling Language) bietet eine ganze Reihe grafischer Übersichten und Regeln zur Spezifizierung von Softwareprojekten. Dabei existieren nicht nur Diagramme für verschiedene Arten von Softwareprojekten, sondern auch für die verschiedenen Lebensphasen des Systems. Weitere Informationen zu UML gibt [15].

Erstellung der Benutzungsoberfläche (GUI)

Die Entwicklung der Benutzungsoberfläche und von lauffähigen Prototypen eines Gerätes wird von zahlreichen Werkzeugen unterstützt.

- Die Programmfamilie *VAPS* der Firma Virtual Prototypes [20] bietet eine objektorientierte Erstellung von Geräteoberflächen. Unterstützt wird dies durch Bibliotheken und zahlreiche grafische Import- und Exportfunktionen. Die exportierten Grafiken können u. a. in *BetterState* importiert und eingesetzt werden. Weiterhin unterstützt *VAPS* den Entwicklungsingenieur beim Requirements-Engineering und der Einhaltung der festgelegten Bedingungen.
- Zur Erstellung der grafischen Benutzeroberfläche ermöglicht das Werkzeug *RAPID* der Firma e-SIM [18] die Generierung der Oberfläche von Prototypen sowie die Modellierung des Verhaltens bei verschiedenen Eingaben mit Hilfe von Statecharts.
- Die Firma *Altia* [19] bietet mit ihrer Produktpalette (z. B. *Altia-Design*, *-FacePlate*) in vielfältiger Weise die Möglichkeit, die Oberfläche von Geräten zu gestalten. Eine Anbindung an gängige Spezifikationswerkzeuge (z. B. *Statemate*) sowie eine Generierung von *graphical code* ist möglich.

2.2 Bewertungskriterien der Gebrauchsfähigkeit

Für die Bestimmung der Gebrauchsfähigkeit sind nach ISO 9241-11 (Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 11: Anforderungen an die Gebrauchstauglichkeit [9]) die Kriterien *Effektivität*, *Effizienz* und *Zufriedenheit* zu berücksichtigen. Die Effektivität beschreibt die Ge-

Kriterien der Gebrauchsfähigkeit nach ISO 9241-11	Maße zur Evaluierung der Gebrauchsfähigkeit (Auswahl)
Effektivität	Anzahl der Fehler, Anzahl der gelernten Funktionen
Effizienz	Ausführungszeit, Lernzeit, Anzahl der Dialogschritte
Zufriedenheit	Benutzer-Rating (Skala)

Bild 2: Zusammenhang zwischen Kriterien der Gebrauchsfähigkeit und ausgewählten Maßen zu deren Evaluierung.

naugigkeit und die Vollständigkeit, mit der ein Benutzer bestimmte Ziele oder Teilziele erreicht. Die Effizienz setzt die Effektivität in das Verhältnis zum benötigten Aufwand. Zufriedenheit beschreibt die Beeinträchtigungsfreiheit und die Akzeptanz des Benutzers bei der Bedienung des technischen Gerätes.

Diese Kriterien legen einen allgemeinen Bewertungsrahmen zur Evaluierung der Gebrauchsfähigkeit von interaktiven Geräten fest. Davon ausgehend können Maße zur Bewertung der Gebrauchsfähigkeit ermittelt werden, wie beispielsweise die Anzahl der zu erlernenden Funktionen, die Lernzeit, die Ausführungszeit, oder das Benutzer-Rating. Die Analyse dieser Maße liefert eine Aussage darüber, ob das Gestaltungsziel *Gebrauchsfähigkeit* erreicht worden ist (s. Bild 2).

Die Effektivität und Effizienz kann durch die Simulation der Interaktion zwischen Benutzer und Gerät auf Grund einer formalen Evaluierung ermittelt werden. Im Gegensatz dazu ist Zufriedenheit nur durch Befragen der Benutzer im Rahmen einer experimentellen Evaluierung ermittelbar.

3 Bewertung durch Simulation des Benutzers bei der Aufgabendurchführung

Das Konzept der rechnergestützten Analyse von MMS stützt sich auf die Kombination einer Systemsimulation und eines normativen Verhaltensmodells des Benutzers. Die Simulation von Mensch-Maschine-Systemen erfordert die möglichst realistische Nachbildung der Aufgabe und des Gesamtsystems bestehend aus technischen, organisatorischen und personellen Komponenten [12]. Entsprechend enthält ein Mensch-Maschine-Modell drei Hauptkomponenten (Bild 3):

Das *Aufgabenmodell* ist eine Zusammenstellung von Aufgaben, die aus Sicht der Entwickler relevant für die Beurteilung der Gebrauchsfähigkeit sein könnten. Das *Systemmodell* beschreibt die vorgesehene Funk-

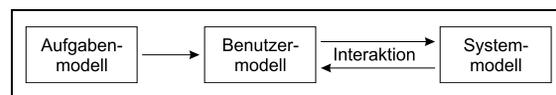


Bild 3: Simulation eines Mensch-Maschine-Systems.

tionalität und Benutzungsoberfläche des in Entwicklung befindlichen Systems. In der Spezifikationsphase enthält das Systemmodell die einzelnen Schritte der Funktionsabläufe in der Weise, wie sich die Designer die spätere, normative Benutzung vorstellen. Diese technischen Spezifikationen bilden die Grundlage für eine analytische Evaluierung. Für empirische Untersuchungen kann später eine Umsetzung der Spezifikation in einen lauffähigen Prototypen erfolgen.

Der Benutzer wird in der Mensch-Maschine Simulation durch ein *Verhaltensmodell* dargestellt, das sensorische, kognitive und motorische Aktivitäten und deren Dauer bei der Durchführung von Aufgaben beschreibt. Das Modell ist normativ, wenn es die vom Designer vorgesehene Aktionsfolge abbildet.

3.1 Erstellung normativer Benutzermodelle

Im Folgenden werden einige formale Ansätze zur Beschreibung kognitiver, sensorischer und motorischer Tätigkeiten vorgestellt, die sich zur Modellierung der Interaktion mit interaktiven Systemen eignen.

Theorie der kognitiven Architektur – ACT

Mit der ACT-Theorie von Anderson [1] können Schritte der menschlichen Wissensverarbeitung, insbesondere wesentliche Phänomene des Lernens und Behaltens von Informationen, erklärt werden. Es wird zwischen dem *sensorischen Speicher*, dem *Kurzzeitgedächtnis* (KZG) und dem *Langzeitgedächtnis* (LZG) unterschieden. Das LZG als Informationsspeicher wird weiterhin in einen deklarativen (faktenbeschreibenden) und einen prozeduralen (kodiert durch Bedingungs-Aktions-Paaren, den Produktionen) Teil zerlegt. Der Vorgang des Wissenserwerbs entspricht dem Lernen von Produktionen bzw. von Produktionsfolgen; dabei werden die Wissensseinheiten von dem deklarativen in den prozeduralen Teil des Langzeitgedächtnisses verschoben. Darüber hinaus liefert ACT durch die Akkumulation von Zugriffszeiten Aussagen über den Zeitbedarf einzelner Prozesse.

Allgemeine (Problemlösungs-) Architektur der Kognition – SOAR

Die SOAR-Theorie nach Laird [13] versucht komplexe Problemlösungsprozesse zu modellieren. Wissen wird als Bedingungs-Aktions-Regeln (Prozeduren) im Langzeitgedächtnis repräsentiert. Ein Lernmechanismus generiert automatisch neue Produktionen,

wobei der Kontext erfolgter Problembearbeitungen und deren Lösungen in generalisierter Form zusammengefasst werden. Die präferierten Produktionen zur Evaluierung möglicher Handlungsoptionen werden parallel in das Kurzzeitgedächtnis eingetragen, während die Abarbeitung der einzelnen Handlungsschritte als serielle Sequenz erfolgt.

Die formale Beschreibung des Benutzerverhaltens – GOMS-Theorie

Die GOMS-Theorie beruht auf der Beschreibung der Aktionen, die Menschen mit interaktiven Geräten durchführen, und wurde erstmals von Card vorgestellt [3]. GOMS steht als Akronym für die Komponenten des Benutzermodells, nämlich für Goals (Ziele), Operators (Operatoren), Methods (Methoden) und Selection Rules (Auswahlregeln). Dabei repräsentieren die *Ziele* die Zustände, die der Anwender mittels der Aufgabenbearbeitung erreichen will. Ein übergeordnetes Ziel besteht oft aus mehreren Teilzielen, die sukzessive ausgeführt werden müssen, um das Hauptziel zu erreichen. Die *Operatoren* sind die sensorischen, kognitiven und motorischen Grundelemente, durch die der Anwender mit dem interaktiven Endgerät interagiert. Sie repräsentieren die kleinsten Aktionseinheiten und werden nicht weiter zerlegt. Die *Methoden* repräsentieren Teilziele und setzen sich aus einer Abfolge von Operatoren und weiteren Methoden zusammen, die vom Anwender nacheinander angewendet werden müssen, um den vorgegebenen Zielzustand zu erreichen. Die *Auswahlregeln* werden verwendet, wenn durch das System mehrere Methoden alternativ zum Erreichen des Ziels möglich sind.

Die Wahl der Auswahlregel kann durch Aufgabenparameter, Training oder Gewohnheiten des Anwenders beeinflusst werden. Bild 4 verdeutlicht die Struktur eines GOMS-Modells. An oberster Stelle steht das Ziel der Aufgabe, welches erreicht werden soll. Die Teilaufgaben sind durch Methoden modelliert,

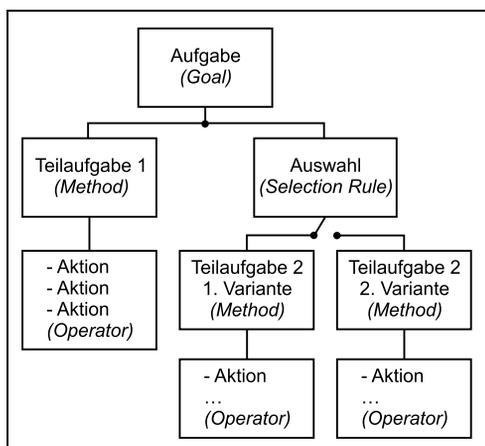


Bild 4: Die Struktur eines GOMS-Modells mit den 4 Komponenten *Goal*, *Methods*, *Operators* und *Selection Rules*.

die die Aktionen zum Erreichen der (Teil-)Aufgabe enthalten. Durch eine Analyse von GOMS-Modellen können sowohl qualitative Aussagen hinsichtlich der Umsetzung der Funktionalität als auch quantitative Vorhersagen zur Bedienbarkeit, z.B. Ausführungs- und Lernzeiten, gemacht werden.

Vergleich der vorgestellten Architekturen

Mit allen vorgestellten Ansätzen ist das Verhalten von Anwendern bei der Benutzung interaktiver Endgeräte beschreibbar. Allen Theorien liegt ein Modell der sequenziellen Informationsverarbeitung beim Menschen zugrunde. Ein wesentlicher Unterschied besteht darin, dass der Entwickler bei Verwendung der ACT- und SOAR-Theorie über fundierte psychologische Kenntnisse hinsichtlich der menschlichen Informationsverarbeitung verfügen muss, um die Theorien sinnvoll einsetzen zu können. Ferner können mit ACT und SOAR zwar eine Vielzahl von kognitiven Prozessen betrachtet werden, eine Modellierung der Wahrnehmung oder Motorik ist jedoch nur bedingt möglich. Der GOMS-Ansatz ist hingegen als ein *Technisches Modell (engineering model)* zur Beschreibung des Benutzerverhaltens anzusehen [4], das die Modellierung motorischer Aktionen erlaubt.

Generierung der Benutzermodelle in den verschiedenen Lebensphasen

Die Benutzermodelle lassen sich bereits sehr früh im Entwicklungsprozess generieren. Bild 5 gibt einen Überblick über die Möglichkeiten der Erstellung formaler Benutzermodelle.

In der Definitionsphase ist die manuelle Erstellung mit Hilfe eines Editors möglich. Die Werkzeuge *QGOMS* von Beard [2], *GOMSED* von Wandmacher [17] oder *GLEAN* von Kieras [11] ermöglichen diese manuelle Erstellung. Der Zeitaufwand für die Erstellung von Hand ist jedoch beträchtlich, sodass

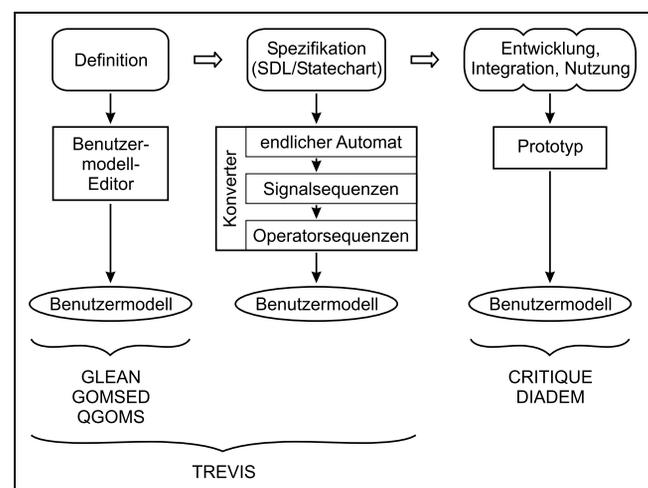


Bild 5: Möglichkeiten der Erzeugung normativer Benutzermodelle.

sie in der Systementwicklung nicht vorgenommen wird. Die in der Spezifikationsphase festgelegte technische Spezifikation kann mit Hilfe eines Konverters automatisch in Benutzermodelle umgewandelt werden. Die automatische Konvertierung, die in Kapitel 3.2 genauer erläutert wird, basiert auf SDL- oder Statechart-Spezifikationen und ist in dem Werkzeug TREVIS (s. Kapitel 4) realisiert.

Eine weitere Möglichkeit der Generierung von Benutzermodellen ergibt sich durch die fehlerfreie Bedienung eines Prototypen. Die dabei aufgezeichneten Aktionen der Bedienung entsprechen den motorischen Operatoren des Benutzermodells. Dieses Vorgehen wurde in den Werkzeugen *DIADEM* von Nirshl [16] und *CRITIQUE* von Hudson [8] umgesetzt. Eine solche formale Bewertung empirisch erhobener Daten büßt jedoch den Vorteil der Anwendbarkeit in frühen Phasen ein.

3.2 Automatische Erzeugung von normativen Benutzermodellen aus funktionalen Spezifikationen

Technische Spezifikationen beschreiben die gesamte Funktionalität eines Gerätes inklusive der Dialogkontrolle. Daher ist es möglich, die Aktionen, die zur Bearbeitung einer Aufgabe nötig sind, zum Teil automatisch aus der Spezifikation in der benötigten Reihenfolge zu extrahieren. Diese Extraktion formaler Benutzermodelle aus funktionalen Spezifikationen verspricht eine deutliche Zeitersparnis gegenüber der manuellen Vorgehensweise. Bild 6 zeigt die Schritte bei der Umwandlung einer Spezifikation im SDL-Format in ein normatives Benutzermodell (im TREVIS-Format, s. Kapitel 4). Beispielhaft ist die Konvertierung anhand der Aufgabe „Aktiviere den Infrarot(IR)-Betrieb eines Handys“ dargestellt. Eine SDL-Spezifikation der Menüfunktionalität des Handys steht als Eingabe für den Konverter zur Verfügung.

Die SDL-Spezifikation wird automatisch geparsed und in einen Endlichen Automaten (EA) umgewandelt. Der Startzustand der Spezifikation wird dabei als Startzustand des EA übernommen. Um das Ziel der Aufgabe festzulegen, muss ein EA-Zustand als Zielzustand manuell markiert werden. Es folgt die automatische Extraktion aller möglichen Pfade vom Start- zum Zielzustand und die Auflistung der jeweils vorkommenden Transitionen. Zyklen werden dabei automatisch erkannt und unterdrückt. Die identifizierten Transitionensequenzen enthalten alle Aktionen für die verschiedenen Bedienmöglichkeiten zum Erreichen des Aufgabenziels, wobei die Aktionen den Operatoren eines Benutzermodells entsprechen. Die Zuordnung der jeweiligen Aktionen zu den Operatoren erfolgt manuell. Daran schließt sich die automatische Umwandlung der Transitionensequenzen in

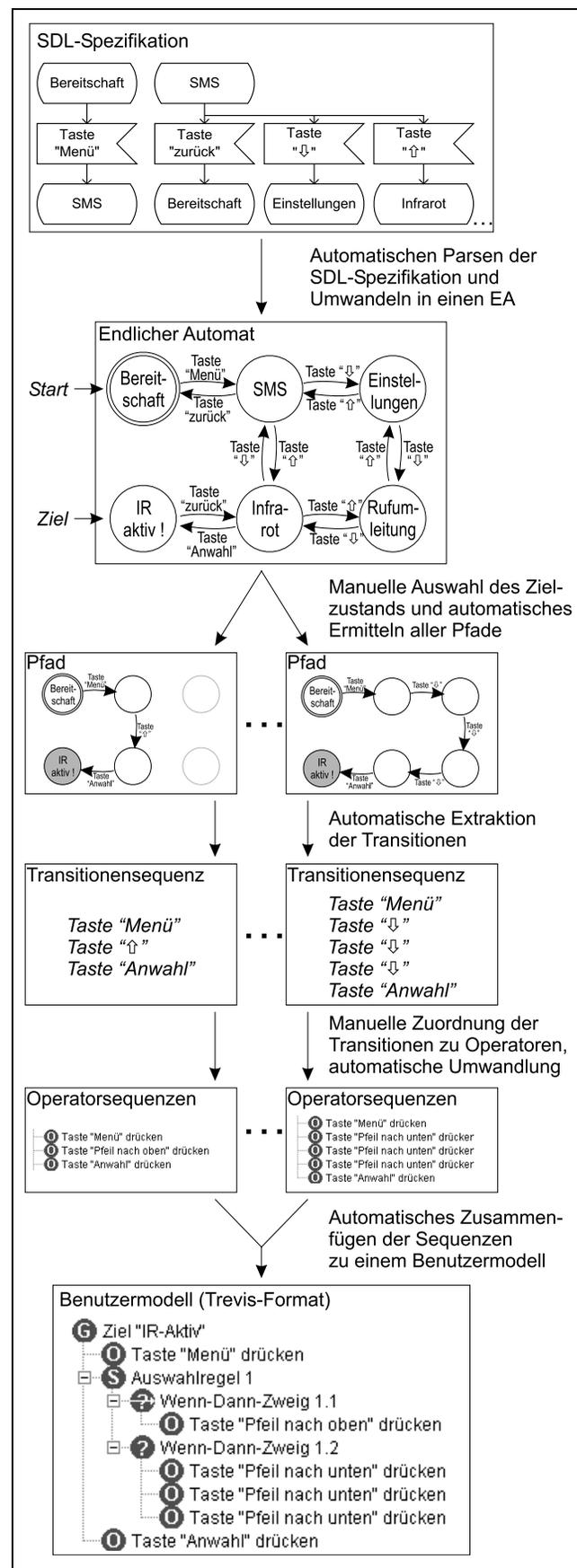


Bild 6: Die Schritte bei der Konvertierung einer SDL-Spezifikation in normative Benutzermodelle (im TREVIS-Format) am Beispiel der Aufgabe „Aktiviere IR-Betrieb eines Handys“.

Operatorsequenzen an. Ein letzter Schritt fasst die Operatorsequenzen zu einem vollständigen Benutzermodell zusammen, wobei eine automatische Herleitung der Selection Rules zur Auswahl verschiedener Ausführungsmöglichkeiten erfolgt.

Diese Vorgehensweise liefert teilautomatisch ein Grundmodell aller zur Durchführung einer Aufgabe nötigen motorischen Aktionen. Eine manuelle Nachbearbeitung des Benutzermodells, z. B. zum Einfügen von kognitiven Operatoren, ist in geringem Umfang erforderlich, da der Konverter den größten Teil der Arbeit zur Erstellung der Benutzermodelle automatisch durchführt.

Die beschriebene Methodik ist für alle Spezifikationen anwendbar, die auf Endlichen Automaten basieren (z. B. SDL oder Statecharts).

4 TREVIS – Werkzeug zur Evaluierung interaktiver Systeme

Zur Unterstützung der formalen Evaluierung interaktiver Systeme befindet sich am Lehrstuhl für Technische Informatik der RWTH Aachen das Werkzeug TREVIS (Tool for Rapid Evaluation of Interactive Systems) in Entwicklung [5, 14]. TREVIS verwendet formale Benutzermodelle auf Basis der GOMS-Theorie und kombiniert verschiedene Evaluierungsmethoden und Entwicklungswerkzeuge.

Den strukturellen Aufbau des Werkzeuges zeigt Bild 7, wobei der grau unterlegte Bereich die in TREVIS integrierten Module kennzeichnet.

Zentraler Bestandteil des Werkzeuges ist der Benutzermodell-Editor, der zum Erstellen und Bearbeiten der Benutzermodelle dient. Als Hilfestellung stehen dem Entwicklungsingenieur u. a. eine Bibliothek zur

Wiederverwendung von GOMS-Komponenten zur Verfügung. Aus den in der Definitionsphase ermittelten Aufgaben können entsprechende Benutzermodelle manuell mit Hilfe des Benutzermodell-Editors erstellt werden. Weiterhin kann eine Gerätespezifikation in SDL, wie in Kapitel 3.2 beschrieben, importiert und in Benutzermodelle umgewandelt werden.

TREVIS beinhaltet unterschiedliche Analyse-Module, die Maße für die Gebrauchsfähigkeit des zu untersuchenden Gerätes liefern: Die *Benutzermodell-Analyse* erzeugt u. a. Prognosen über die zu erwartende Ausführung- und Lernzeit. Eine Angabe über die Struktur des Modells und ein Diagramm der benutzten Operatoren und deren Häufigkeit ergänzt die Analyse.

Ein Vergleich verschiedener Designvarianten wird mit der *Design-Analyse* durchgeführt, die die Benutzermodell-Analysen der verschiedenen Varianten übersichtlich nebeneinander darstellt und so eine Grundlage für einen schnellen Vergleich der Analyseergebnisse liefert.

In der *Fehleranalyse* werden neben den Benutzermodellen auch Aktionssequenzen ausgewertet. Diese Sequenzen enthalten die Interaktionen von Probanden an einem Prototypen und können aus einer empirischen Evaluierung generiert werden. TREVIS stellt eine Funktion zum Einladen dieser Aktionssequenzen bereit. Die Fehleranalyse vergleicht die Benutzermodelle (als Repräsentation eines *perfekten* Benutzers) mit den importierten Aktionssequenzen (als Repräsentation *realer* Benutzer). Diese Analyse gibt Aufschluss über Anzahl und Art der Fehler der realen Benutzer, sowie über Aktionsfrequenzen und Fehlerhäufigkeit.

Weitere Informationen über TREVIS sowie eine lauffähige Demoversion finden sich unter:

www.techinfo.rwth-aachen.de/Forschung/MMI/Trevis

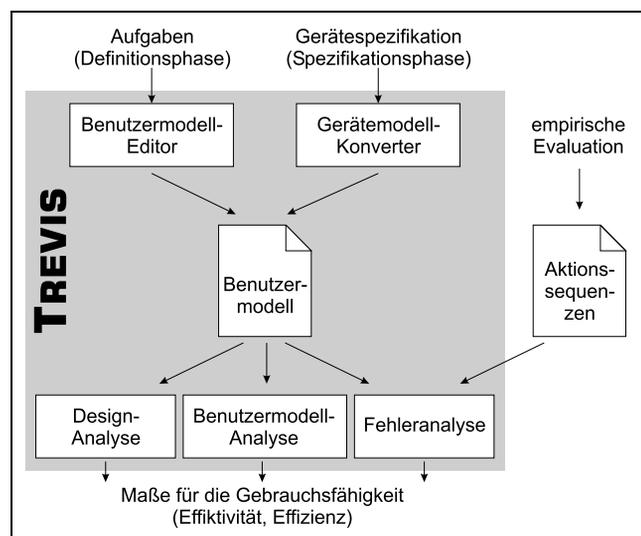


Bild 7: Eine Übersicht des strukturellen Aufbaus des Werkzeuges TREVIS.

5 Zusammenfassung und Ausblick

Der Einsatz formaler Methoden mit Hilfe von normativen Benutzermodellen ermöglicht die Evaluierung der Gebrauchsfähigkeit interaktiver Geräte schon in frühen Lebensphasen. Dadurch können die Evaluierungsergebnisse noch im laufenden Entwicklungsprozess berücksichtigt werden. Als besonders zuverlässig und einfach zu handhaben hat sich dazu die GOMS-Theorie herausgestellt. Mit Hilfe dieser Theorie können Aussagen über die Effektivität und Effizienz bei der Bedienung eines interaktiven Gerätes getroffen werden. Um den Zeitaufwand bei der Erstellung der Benutzermodelle so gering wie möglich zu halten, wurde ein Konverter entwickelt und in das Werkzeug TREVIS integriert, der Benutzermodelle semi-automatisch aus technischen Spezifikationen generiert.

Eine Kombination der einmal erzeugten Benutzermodelle mit weiteren Evaluierungsmethoden in spä-

teren Lebensphasen des Gerätes ermöglicht weitere Analysen, z. B. über Bedienungsfehler und das Benutzerverhalten.

GOMS erlaubt bislang nur die Bewertung isolierter Maße der Gebrauchsfähigkeit. Da der Funktionsumfang interaktiver Geräte jedoch rapide zunimmt, ist eine Betrachtung dieser Maße im Kontext zueinander nötig. TREVIS wird zur Zeit weiterentwickelt, um die Konsistenz bei der Bedienung sowie die Komplexität des Gerätes mit Hilfe normativer Benutzermodelle bewerten zu können.

Eine Anbindung von TREVIS an weitere Entwicklungswerkzeuge, z. B. zur Oberflächen- bzw. Prototypen-Erstellung, ermöglicht das Wiederverwenden der Benutzermodelle sowie der einmal generierten Analyseergebnisse. So ist eine automatische Überprüfung der Funktionalität sowie eine Analyse der Gestaltung der Geräteoberfläche durchführbar.

Literatur

- [1] *Anderson, J. R.*: Language, memory, and thought. Hillsdale, NJ USA: Lawrence Erlbaum Associates 1976.
- [2] *Beard, D.; Entrikin, S.; Conroy, P.; Wingert, N.; Schou, C.; Smith, D.; Denelsbeck, K.*: Quick GOMS: A visual software engineering tool for simple rapid time-motion modeling. In: ACM Interactions 4[3] (1997) S. 31–36.
- [3] *Card, S. K.; Moran, T. P.; Newell, A.*: The psychology of human-computer interaction. Hillsdale, NJ USA: Lawrence Erlbaum Associates 1983.
- [4] *Gray, W. D.; Boehm-Davis, D.; John, B. E.; Kieras, D. E.*: Cognitive analysis of dynamic performance: Cognitive process analysis and modeling. Department of Psychology at George Mason University 1999.
- [5] *Hamacher, N.*: Entwicklung und Implementierung eines Werkzeugs zur Bewertung interaktiver Systeme basierend auf normativen Benutzermodellen. Diplomarbeit. Lehrstuhl für Technische Informatik, RWTH Aachen 2000.
- [6] *Harel, D.; Sintzoff, M.*: Statecharts: A visual formalism for complex systems. In: Science of Computer Programming. Amsterdam: North-Holland (1987) S. 231–274.
- [7] *Hogrefe, D.*: Estelle, LOTOS und SDL. Berlin: Springer Verlag 1989.
- [8] *Hudson, S.; John, B. E.; Knudsen, K.; Byrne, M.*: A tool for creating predictive performance models from user interface demonstrations. In: Proceedings of 12th annual ACM Symposium on User Interface Software and Technology. New York, USA: ACM Press (1999) S. 93–102.
- [9] ISO 9241: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 1–17. Deutsches Institut für Normung 1999.
- [10] *Jones, T. O.; Sasser, W.*: Why satisfied customers defect? In: Harvard Business Report 4[1] (1995) S. 88–99.
- [11] *Kieras, D. E.*: A guide to GOMS model usability evaluation using GOMSL and GLEAN3. Electrical Engineering and Computer Science Department, University of Michigan, 1999.
- [12] *Kraiss, K.-F.*: Modellierung von Mensch-Maschine Systemen. In: Willumeit H.-P. (Eds.): Verlässlichkeit von Mensch-Maschine Systemen. ZMMS-Spektrum (1995) S. 15–35.
- [13] *Laird, J. E.; Rosenbloom, P.; Newell, A.*: SOAR: An architecture for general intelligence. In: Artificial Intelligence 33 (1987) S. 1–64.
- [14] *Marrenbach, J.*: Werkzeug-basierte Evaluierung der Benutzungsfreundlichkeit interaktiver Endgeräte mit normativen Benutzermodellen. Dissertation. Aachen: Shaker Verlag 2001.
- [15] *Martin, J.; Odell, J. J.*: Objektorientierte Modellierung mit UML. München: Prentice Hall 1999.
- [16] *Nirschl, G.*: Werkzeug zur Gestaltung und Bewertung von Mensch-Maschine-Dialogen im Kraftfahrzeug. In: VDI-Gesellschaft Fahrzeugtechnik – Elektronik im Kraftfahrzeug. Düsseldorf: VDI-Verlag GmbH (1990) S. 19–35.
- [17] *Wandmacher, J.*: Ein Werkzeug für GOMS-Analysen zur Simulation und Bewertung von Prototypen beim Entwurf. In: G. Szwillus: Prototypen für Benutzungsschnittstellen – Grundlagen, Techniken, Erfahrungen. Paderborn (1997) S. 35–42.
- [18] <http://www.esim.com>
- [19] <http://www.altia.com>
- [20] <http://www.virtualprototypes.ca>



Dipl.-Inform. Nico Hamacher studierte Informatik an der RWTH Aachen und arbeitet seit 2000 als wissenschaftlicher Mitarbeiter am Lehrstuhl für Technische Informatik im Bereich Mensch-Maschine-Kommunikation.

Adresse: Lehrstuhl für Technische Informatik, RWTH Aachen, Ahornstr. 55, D-52074 Aachen
E-Mail: hamacher@techinfo.rwth-aachen.de



Univ.-Prof. Dr.-Ing. Karl-Friedrich Kraiss ist Inhaber des Lehrstuhls für Technische Informatik in der Fakultät für Elektrotechnik der RWTH Aachen. Die Hauptarbeitsgebiete des Lehrstuhls sind die Mensch-Maschine-Kommunikation sowie erfahrungsgestützte und lernfähige Systeme.

Adresse: Lehrstuhl für Technische Informatik, RWTH Aachen, Ahornstr. 55, D-52074 Aachen
E-Mail: kraiss@techinfo.rwth-aachen.de



Dr.-Ing. Jörg Marrenbach studierte Maschinenbau mit der Fachrichtung Luft- und Raumfahrt-technik an der RWTH Aachen. Von 1996 bis 2001 arbeitete er am Lehrstuhl für Technische Informatik als wissenschaftlicher Mitarbeiter und promovierte mit dem Thema „Werkzeug-basierte Evaluierung interaktiver Endgeräte“. Seit 2001 ist er bei der AUDI AG im Bereich Infotainment/MMI beschäftigt.

Adresse: AUDI AG, I/EE-54, D-85045 Ingolstadt
E-Mail: joerg.marrenbach@audi.de