

# Generierung normativer Benutzermodelle aus SDL-Spezifikationen<sup>1</sup>

Nico Hamacher, Jörg Zieren, Jörg Marrenbach, Karl-Friedrich Kraiss

{hamacher, zieren, marrenbach, kraiss}@techinfo.rwth-aachen.de

Lehrstuhl für Technische Informatik - RWTH Aachen  
Ahornstr. 55, D-52074 Aachen  
Tel.: +49-241-8026105  
Fax: +49-241-8888308

## Einleitung

Begriffe wie *Ergonomie* und *Benutzungsfreundlichkeit*<sup>2</sup> spielen bei der Entwicklung interaktiver Geräte 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. Das liegt vor allem daran, dass Kunden die Produkte, mit denen sie nicht zufrieden sind, in immer größerem Maße unberücksichtigt lassen [Jones & Sasser 1995].

Die übliche empirische Bewertung der Gebrauchsfähigkeit erfordert einen Prototypen und liefert dadurch die Ergebnisse meist so spät, dass die laufende Entwicklung keinen Nutzen mehr daraus ziehen kann. Eine formale Evaluierung basierend auf normativen Benutzermodellen kompensiert diese Nachteile, da sie sehr früh bei der Geräteentwicklung eingesetzt werden kann.

Da heutzutage vermehrt Programme zur Erstellung von Gerätespezifikationen bei der Geräteentwicklung zum Einsatz kommen, kann dieses ohnehin vorhandene technische Know-how die Grundlage für formale ergonomische Analysen bilden. Dieser Beitrag beschreibt im Einzelnen die Evaluierung mit Hilfe normativer Benutzermodelle, die Spezifizierung mit SDL sowie die Vorgehensweise bei der Umwandlung von SDL-Spezifikationen in normative Benutzermodelle. Die vorgestellte Vorgehensweise ist in einem Werkzeug zur formalen Bewertung der Gebrauchsfähigkeit realisiert, und kommt mit einer Beispielspezifikation zum Einsatz.

## Entwicklung und Bewertung interaktiver Endgeräte

### *Entwicklung des Gerätes*

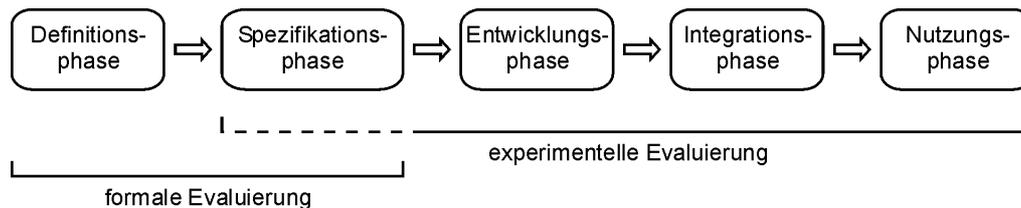
Der Werdegang eines Gerätes gliedert sich in verschiedene voneinander abgrenzbare Lebensphasen (Abbildung 1). Zu Beginn wird in der Definitionsphase ein allgemeines System-

---

<sup>1</sup> 4. Berliner Werkstatt Mensch-Maschine-Systeme "Bedienen und Verstehen", 10.-12. Oktober 2001, Berlin

<sup>2</sup> 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 *Benutzungsfreundlichkeit* gesprochen.

konzept 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.



**Abbildung 1: Die Lebensphasen eines interaktiven Gerätes.**

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

### **Bewertung der Benutzungsfreundlichkeit**

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

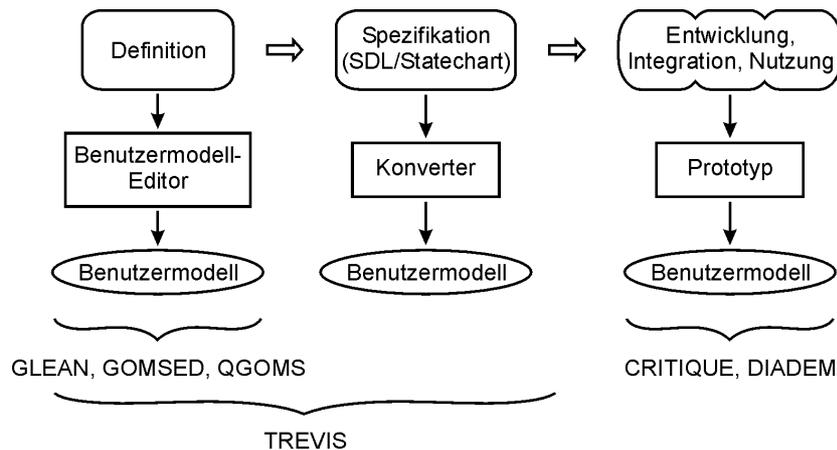
Diese Kriterien legen einen allgemeinen Bewertungsrahmen zur Evaluierung der Gebrauchsfähigkeit von interaktiven Geräten fest. Davon ausgehend können Maße zur Bewertung der Gebrauchsfähigkeit ermittelt werden, wie beispielsweise die Anzahl der zu erlernenden Funktionen, die Lernzeit, die Ausführungszeit,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

Zufriedenheit ist nur durch direktes Befragen von Benutzern im Rahmen empirischer Versuche möglich. Dazu muss ein lauffähiger Prototyp des Systems vorliegen, der erst in späten Lebensphasen zur Verfügung steht. Die Effektivität und Effizienz sind jedoch auch durch formal analytische Evaluierungsmethoden messbar. Diese beschreiben die Interaktionen eines Benutzers mit einem Gerät in einem Benutzermodell und sind vor allem in den frühen Lebensphasen eines Systems anwendbar.

Einen ingenieurmäßigen Ansatz zur Modellierung und Analyse von Benutzermodellen beschreibt die GOMS-Theorie. GOMS steht für die Komponenten des Modells, nämlich Ziele (Goals), Operatoren (Operators), Methoden (Methods) und Auswahlregeln (Selection Rules). *Ziele* sind die Ziele, 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. 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. [Card, Moran & Newell, 1983].

Benutzermodelle lassen sich bereits sehr früh im Entwicklungsprozess generieren. Abbildung 2 gibt einen Überblick über die Möglichkeiten der Erstellung formaler Benutzermodelle.



**Abbildung 2: Möglichkeiten der Erzeugung normativer Benutzermodelle.**

In der Definitionsphase ist die manuelle Erstellung mit Hilfe eines Editors möglich. Die Werkzeuge *QGOMS* von Beard [Beard 1997], *GOMSED* von Wandmacher [Wandmacher 1997] oder *GLEAN* von Kieras [Kieras 1999] ermöglichen diese manuelle Erstellung. Der Zeitaufwand für die Erstellung von Hand ist jedoch beträchtlich, so dass 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 im nächsten Kapitel genauer erläutert wird, basiert auf SDL- oder Statechart-Spezifikationen und ist in dem Werkzeug TREVIS 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 Nirschl [Nirschl 1997] und *CRITIQUE* von Hudson [Hudson et. al. 1999] umgesetzt. Eine solche formale Bewertung empirisch erhobener Daten büßt jedoch den Vorteil, die Anwendbarkeit in frühen Phasen, ein.

## Spezifizierung der Gerätefunktionalität mit SDL

Um die technische Funktionalität eines Gerätes eindeutig und präzise spezifizieren zu können, wurde SDL (*Specification and Description Language*) entwickelt. SDL erlaubt die Festlegung aller Funktionsparameter eines Gerätes in elektronischer Form, wobei die Notation so gewählt wurde, dass sie nicht nur von Computern, sondern vor allem von Menschen gelesen werden kann. Zudem ist SDL nicht darauf angelegt, ein lauffähiges System zu erhalten, was auf Grund des Verzichts auf Implementierungsdetails auch nicht möglich ist.

Die Beschreibung der Systemfunktionalität basiert in SDL auf erweiterten Endlichen Automaten, wobei zwei Darstellungsformen existieren: SDL-GR (*Graphical Representation*) verwendet Symbole und Diagramme während SDL-PR (*Phrase Representation*) ausschließlich Schlüsselwörter und textuelle Beschreibungen enthält.

Abbildung 3 gibt einen Überblick über einige wichtige Symbole der grafischen Darstellungsform SDL-GR.

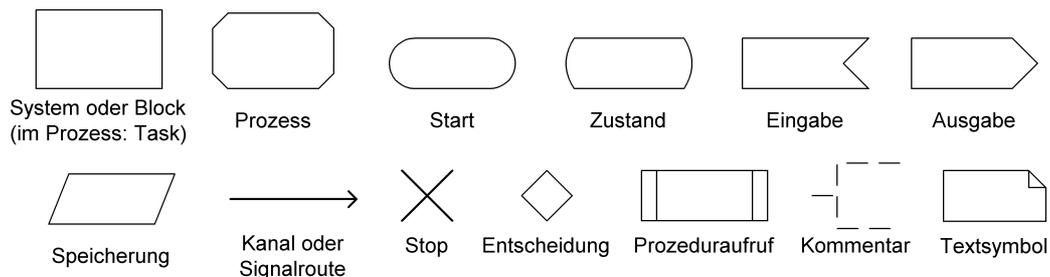


Abbildung 3: Symbole von SDL(-GR).

Ein System wird in SDL aus *Blöcken* und *Prozessen* aufgebaut, wobei Blöcke die hierarchische Struktur und Prozesse das dynamische Verhalten des Systems beschreiben.

Ein Prozess stellt einen erweiterten Endlichen Automaten dar und beginnt seine Ausführung mit einem Startzustand. Die Zustandsübergänge werden durch Ein- und Ausgabesignale sowie durch Task- oder Prozeduraufrufe bewirkt. Dabei liegen die Eingangssignale (wirken von der Umwelt auf das System ein) in einer internen Warteschlange an und werden der Reihe nach abgearbeitet. Ausgangssignale (vom System an die Umwelt) werden asynchron abgegeben, d.h. der Prozess wartet nicht explizit auf eine Bestätigung, sondern arbeitet nach dem Versenden des Signals direkt weiter. Ein Task kann Berechnungsanweisungen oder Variablendeklarationen enthalten, während ein Prozeduraufruf den Sprung in einen Unterprozess ausführt. Kanäle bzw. Signalrouten ordnen die Beschreibungselemente einer vorgegeben Reihenfolge zu. Variableninhalte und -daten können gespeichert und z.B. in Entscheidungen ausgewertet werden.

## Konvertierung der SDL-Spezifikationen in Benutzermodelle

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 [Zieren 2000]. Diese Extraktion formaler Benutzermodelle aus funktionalen Spezifikationen verspricht eine deutliche Zeitersparnis gegenüber der manuellen Vorgehensweise. Abbildung 4 zeigt die Schritte bei der Umwandlung einer in der Beschreibungssprache SDL vorliegenden Spezifikation in ein normatives Benutzermodell (im TREVIS-Format). Beispielhaft ist die Konvertierung anhand der Aufgabe „Aktiviere den Infrarot(IR)-Betrieb eines Handys“ dargestellt.

Der erste grundlegende Schritt besteht in der Konvertierung der SDL-Spezifikation (1) in einen Endlichen Automaten (EA). Dazu wird die SDL-Spezifikation zunächst aus einer Textdatei eingelesen und geparkt. Hierzu kommt eine frei verfügbarer SDL-2000-Grammatikdefinition zum Einsatz [Schmitt 2000]. Diese Grammatik besteht aus einer maschinenlesbaren Definition der SDL-Syntax, aus der mit Hilfe des Parser-Generators ANTLR

[ANTLR 2001] ein Parser generiert wird. Dieser erzeugt aus der SDL-Spezifikation einen Ableitungsbaum (AST, Abstract Syntax Tree), der in einer hierarchischen Struktur die zugrunde liegende Sprachsyntax der Spezifikation beschreibt (2). Aus diesem AST wird eine Liste mit Prozessen erzeugt, aus der der Anwender nun den Prozess auswählt, der die Interaktion des Systems mit dem Benutzer modelliert. Aus diesem Prozess wird automatisch ein EA (3) erzeugt. Auch die Signale, die der Prozess während eines Zustandsübergangs sendet, werden im EA berücksichtigt.

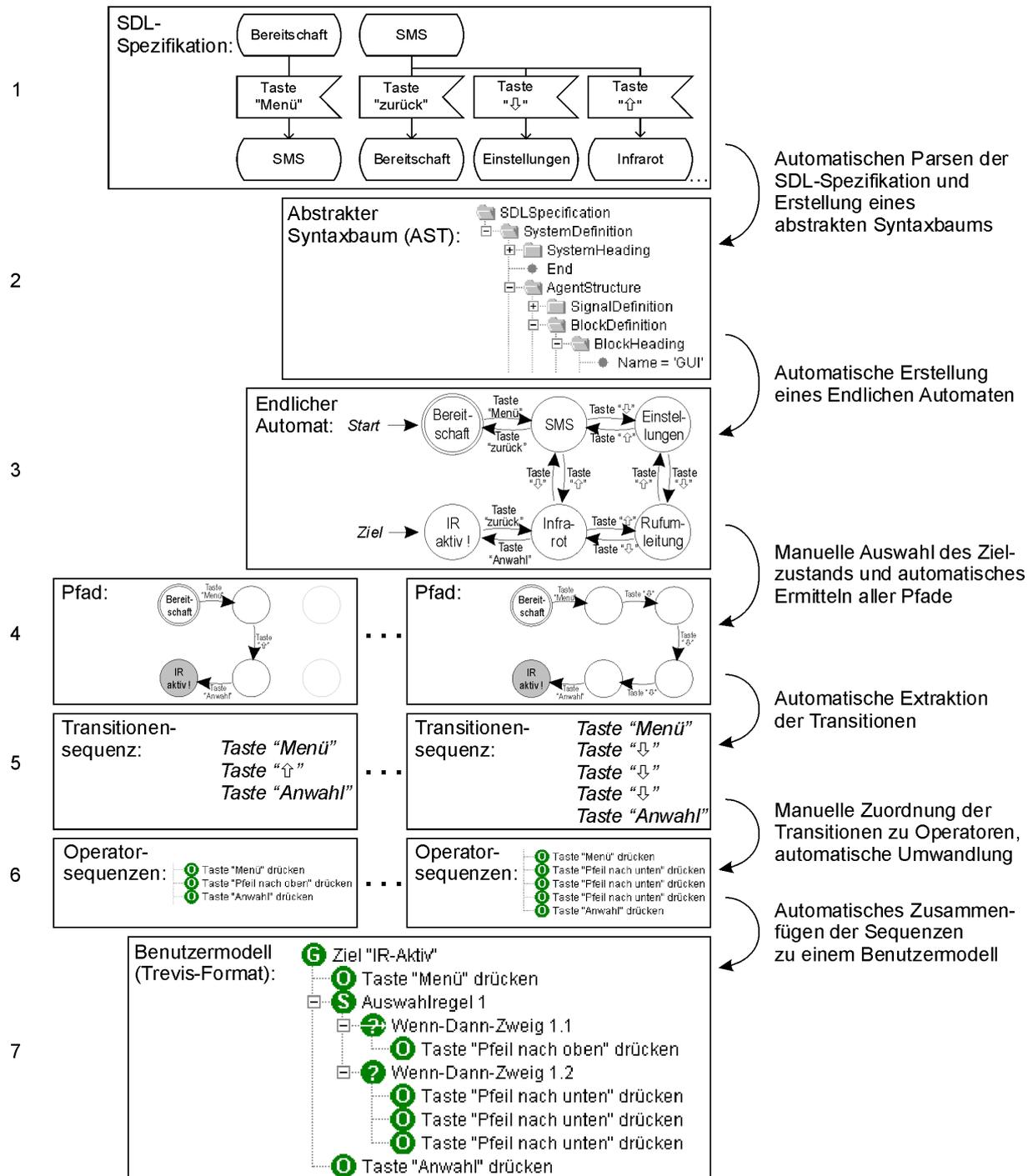


Abbildung 4: Die Schritte bei der Konvertierung einer SDL-Spezifikation in Benutzermodelle am Beispiel der Aufgabe „Aktiviere IR-Betrieb eines Handys“.

Der Startzustand des EA entspricht dem Startzustand des SDL-Prozesses. Der Anwender markiert manuell einen Zustand des Prozesses (bzw. des Automaten) als Zielzustand, in dem die gestellte Aufgabe erfüllt ist. Anschließend werden automatisch alle zyklusfreien Pfade vom Start- zum Zielzustand ermittelt (4). Die Pfade bestehen aus einer alternierenden Sequenz von Zuständen und Zustandsübergängen (Transitionen), wobei letztere jeweils durch ein empfangenes Signal ausgelöst werden. Da die Transitionen den Benutzereingaben und somit einem GOMS-Operator entsprechen, werden die Transitionensequenzen aus den einzelnen Pfaden extrahiert (5). Es folgt die manuelle Zuordnung der vorkommenden Signale zu entsprechenden Operatoren, woraufhin die Transitionensequenzen in Operatorsequenzen umgewandelt werden (6). Auf diese Weise ergibt sich aus jedem Pfad durch den EA eine Operatorsequenz. Unter Verwendung von Auswahlregeln erfolgt eine Zusammenfassung aller Operatorsequenzen zu einem vollständigen Benutzermodell (7).

Falls einige der auf einem Pfad vom Start- in den Zielzustand gesendeten Signale an den Benutzer gerichtet sind und von diesem verarbeitet werden (z.B. optische oder akustische Anzeigen), können sie sensorischen oder kognitiven Operatoren zugeordnet werden. Der Konverter fügt an der entsprechenden Stelle diese entsprechen Operatoren in das Benutzermodell ein.

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

Um diese Benutzermodellgenerierung durchführen zu können, sind folgende Bedingungen an die zugrunde liegende SDL-Spezifikation nötig:

- ) Es muss genau ein Prozess existieren, der im Rahmen des betrachteten Ziels allein für die Interaktion mit dem Benutzer relevant ist. Dieser Prozess muss ausschließlich durch Signale (d.h. Eingaben) des Benutzers in den gewünschten Zielzustand überführbar sein; er darf keine Signale, die lediglich von anderen Prozessen gesendet werden können, erwarten.
- ) Der betrachtete Prozess darf keine Zustandsübergänge enthalten, die aufgrund einer bedingten Verzweigung vom Inhalt einer Variablen abhängen. Andernfalls ist die Erstellung des Endlichen Automaten, der das Zustandsübergangsverhalten des Prozesses nachbildet, nicht möglich.

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

## **Integration des Konverters in das Werkzeug TREVIS**

Der in diesem Beitrag vorgestellte SDL-Konverter wurde in das Werkzeug TREVIS (Tool for Rapid Evaluation of Interactive Systems) integriert. Zur Verdeutlichung der Funktionalität des Konverters werden im Folgenden aus einer Spezifikation eines realen interaktiven Systems beispielhaft Benutzermodelle erstellt. Das dabei betrachtete System ist der Prototyp einer elektronischen Bedienungsanleitung (eBA) für ein KFZ [Marrenbach 2000].

## TREVIS (Tool for Rapid Evaluation of Interactive Systems)

Am Lehrstuhl für Technische Informatik der RWTH Aachen wird das Werkzeug TREVIS entwickelt, welches eine Bewertung der Benutzerfreundlichkeit interaktiver Geräte auf Basis der GOMS-Theorie erlaubt [Marrenbach 2001]. Die Benutzermodelle werden in TREVIS in Form eines hierarchischen Baums dargestellt (Abbildung 5). Ziele werden dabei mit einem „G“, Methoden mit einem „M“, Operatoren mit einem „O“ und Auswahlregeln mit einem „S“ markiert. Die einzelnen Wege einer Auswahlregeln sind durch ein „?“ gekennzeichnet. Die Wiederverwendung einzelner Komponenten wird durch eine Bibliothek ermöglicht, wobei Bibliothekskomponenten durch ein kleines „B“ markiert sind.



Abbildung 5: Darstellung eines Benutzermodells in TREVIS.

TREVIS bietet unterschiedliche Analysen, die Maße für die Gebrauchsfähigkeit des zu untersuchenden Gerätes liefern, nämlich die Benutzermodell-Analyse, die Design-Analyse sowie die Fehleranalyse [Hamacher et. al. 2001].

Prognosen u.a. über die zu erwartende Ausführungs- und Lernzeit erzeugt die Benutzermodell-Analyse. Angaben über die Struktur des Modells und Diagramme der benutzten Operatoren und deren Häufigkeit ergänzen die Analyse.

Die Möglichkeit eines schnellen Vergleichs verschiedener Designvarianten bietet die Design-Analyse, die die Benutzermodell-Analysen der verschiedenen Varianten übersichtlich nebeneinander darstellt.

In der Fehleranalyse werden die normativen Interaktionen der Benutzermodelle mit aufgezeichneten Interaktionen realer Benutzer verglichen. Diese aufgezeichneten Interaktionen können aus empirischen Evaluierungen generiert und in TREVIS importiert werden. Ein Vergleich der *perfekten* mit den *realen* Aktionen gibt Aufschluss über Anzahl und Art der Fehler der realen Benutzer, sowie über Aktionsfrequenzen und Fehlerhäufigkeit.

Weitere Informationen über das Werkzeug sowie eine lauffähige Demoversion finden sich unter <http://www.techinfo.rwth-aachen.de/Forschung/MMI/Trevis>.

### Prototyp „Elektronische Bedienungsanleitung (eBA)“

Bei dem Softwareprototyp der eBA erfolgt die Darstellung der Inhalte eines konventionellen Handbuchs auf dem Bordmonitor des KFZ in multimedialer Form. Die Inhalte sind dabei mittels einer hierarchischen Menüstruktur ähnlich einem Inhaltsverzeichnis in Themenbereiche geordnet. Ein Bedienelement erlaubt die Steuerung mit den Bewegungen *links*, *recht*, *hoch*, und *runter*, sowie die Selektion eines Eintrages. Die Darstellung der eBA auf dem Bordmonitor teilt sich in zwei Bereiche. Der linke Bereich enthält die Menüstruktur sowie die Auswahlmöglichkeiten. Der rechte Bereich dient der Darstellung einer Voraussicht auf kommende Menüs bzw. entsprechenden Handbuchdaten in Listenform oder als Animation. Abbildung 6 zeigt das Hauptmenü der eBA, wobei der Auswahlbalken auf dem Menüpunkt

„Betriebsanleitung“ steht und im rechten Bildschirmbereich eine Voraussicht auf das entsprechende Menü dargestellt wird.



Abbildung 6: Das Hauptmenü des Prototypen der elektronischen Bedienungsanleitung.

Für die eBA wurde eine Spezifikation in SDL modelliert, die den gesamten Funktionsumfang enthält. Die Spezifikation enthält die Prozesse „Dialog“, der die Interaktionen mit dem Benutzer beschreibt, und „Anzeige“, der die Art und Inhalte der Anzeigen steuert. Abbildung 7 zeigt als Beispiel der Spezifikation einen Ausschnitt des Prozesses „Dialog“.

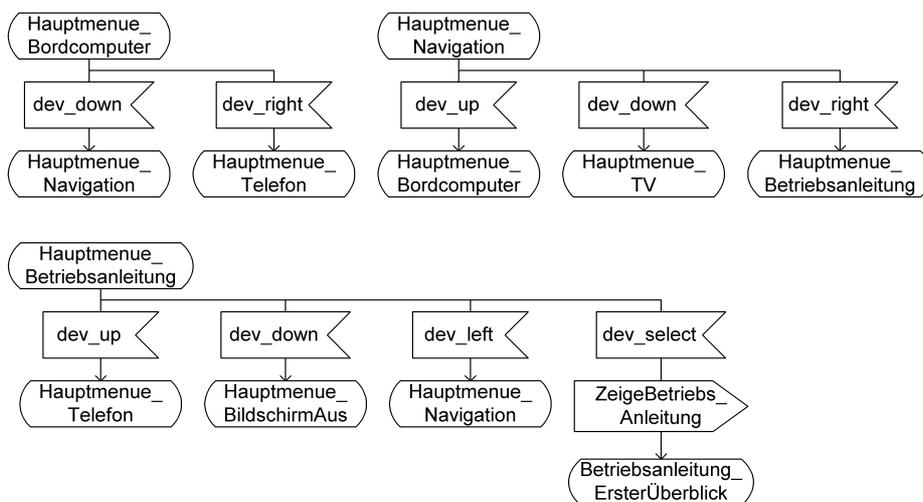


Abbildung 7: Ausschnitt aus der Spezifikation des Prozesses „Dialog“.

### Automatische Generierung der Benutzermodelle

Aus der vorhandenen Spezifikation werden mit Hilfe des SDL-Konverters automatisch Benutzermodelle für die Aufgaben „Bestimmung der Motorleistung, des durchschnittlichen Verbrauchs sowie der Fahrzeuglänge“ erstellt. Nach Angabe der SDL-Datei werden die Prozesse der SDL-Spezifikation aufgelistet (Abbildung 8).

Nach Auswahl der zu betrachtenden Prozesse wird automatisch eine Liste aller vorkommender Signale erzeugt. Diese Signale werden den entsprechenden Operatoren zugeordnet. Abbildung 9 zeigt den Zuordnungdialog für den Zielzustand „TechnischeDatenMotor“.

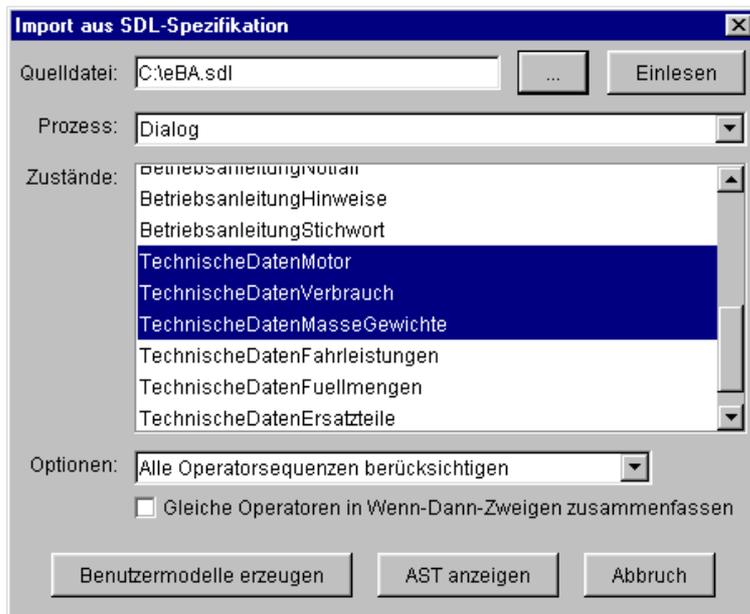


Abbildung 8: Importdialog der Prozesse für das Beispiel „elektronische Bedienungsanleitung“.

Im oberen Bereich stehen links die in der Spezifikation vorkommenden Signale, während rechts die in der Bibliothek von TREVIS enthaltenen Operatoren aufgelistet sind. Für die Zuordnung können diese verwendet, oder neue Operatoren erstellt werden. Die Zuordnung der Signale zu den Operatoren ist in der Tabelle im unteren Bereich dargestellt. Eine Erstellung der Benutzermodelle kann erst vorgenommen werden, wenn allen Signalen ein Operator zugeordnet wird.

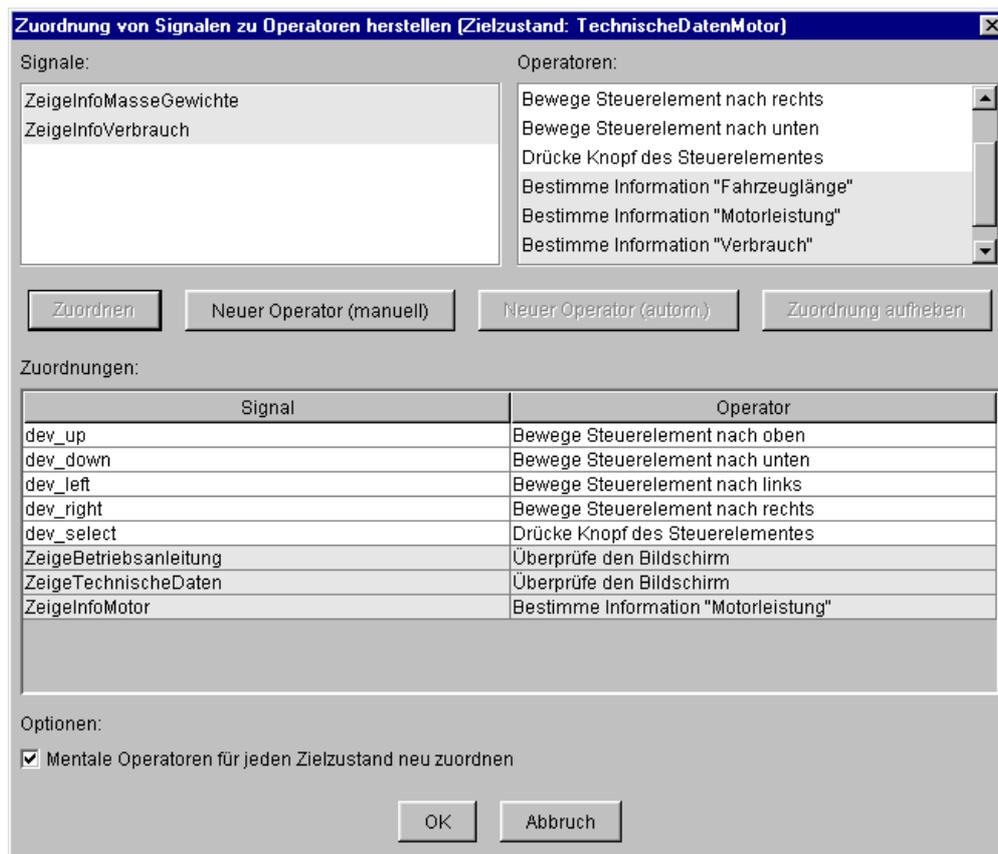


Abbildung 9: Zuordnungsdialg für den Zielzustand „TechnischeDatenMotor“.

Nach der Zuordnung der Signale zu den Operatoren werden die Benutzermodelle automatisch erstellt und in TREVIS angezeigt. Der Konverter erzeugt für jeden ausgewählten Zielzustand ein Benutzermodell, welches alle Möglichkeiten zum Erreichen des Zielzustandes enthält. Diese unterschiedlichen Wege sind jeweils in Auswahlregeln modelliert (Abbildung 10).

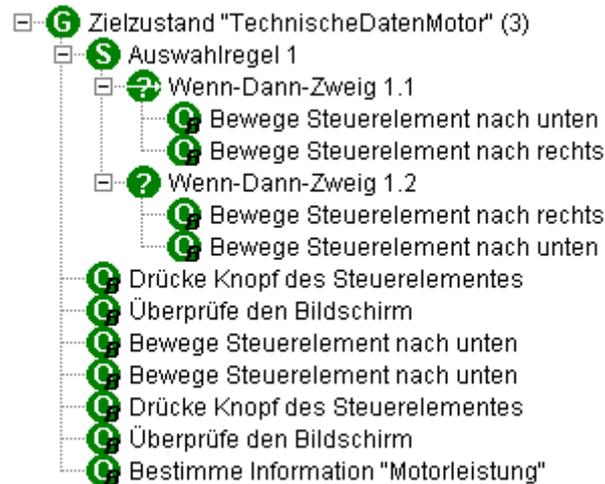


Abbildung 10: Automatisch generiertes Benutzermodell für den Zielzustand „TechnischeDatenMotor“.

## Zusammenfassung

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 SDL-Spezifikationen generiert.

Die Konvertierung erfordert lediglich geringen manuellen Aufwand und beschleunigt dadurch die Erstellung normativer Benutzermodelle. Das vorgestellte Vorgehen bei der Benutzermodellgenerierung ist für alle Spezifikations Sprachen, die auf Endlichen Automaten basieren, anwendbar.

Die korrekt Funktionsweise wird anhand eines Beispiels verdeutlicht.

## Literatur

ANTLR 2001: ANTLR-ANother Tool for Language Recognition. <http://www.antlr.org>

Beard, D., S. Entrikin, P. Conroy, N. Wingert, C. Schou, D. Smith, K. Denelsbeck (1997): Quick GOMS: A visual software engineering tool for simple rapid time-motion modeling. In: *ACM Interactions* 4[3] S. 31-36.

Card, S.K., T.P. Moran, A. Newell (1983): *The Psychology of Human-Computer Interaction*. Hillsdale, NJ USA: Lawrence Erlbaum Associates.

Hamacher N., Marrenbach J., Kraiss K.-F. (2001): *Formal Usability Evaluation of Interactive Systems*. In: Johannsen, G. (Eds.): 8th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human Machine Systems 2001. Preprints, pp. 577-581, September 18-20, Kassel, VDI/VDE-GMA

- Hamacher, N (2000): *Entwicklung und Implementierung eines Werkzeugs zur Bewertung interaktiver Systeme basierend auf normativen Benutzermodellen*. Diplomarbeit. Lehrstuhl für Technische Informatik, RWTH Aachen.
- Hogrefe, D. (1989): *Estelle, LOTOS und SDL*. Berlin: Springer Verlag.
- Hudson, S. , B.E. John, K. Knudsen, M. Byrne (1999): 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 S. 93-102.
- ISO9241 (1999): Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten - Teil 1-17. Deutsches Institut für Normung.
- Jones, T.O., W. Sasser (1995): Why satisfied customers defect ?. In: *Harvard Business Report* 4[1] S. 88-99.
- Kieras, D.E. (1999): *A guide to GOMS model usability evaluation using GOMSL and GLEAN3*. Electrical Engineering and Computer Science Department, University of Michigan.
- Kraiss, K.-F. (1995): Modellierung von Mensch-Maschine Systemen. In: Willumeit H.-P. (Eds.): *Verlässlichkeit von Mensch-Maschine Systemen*. ZMMS-Spektrum S. 15-35.
- Marrenbach J., Kraiss K.-F., Hahn W., Bengler K. (2000): Entwicklung und Evaluierung einer multimedialen Betriebsanleitung für Kfz-Bordsysteme. In: Gärtner, K.-P. (Eds.): *42. Fachausschußsitzung Anthropotechnik, 2000-02, S. 139-159, 24.-25. Oktober, München*
- Marrenbach, J. (2001): *Werkzeug-basierte Evaluierung der Benutzungsfreundlichkeit interaktiver Endgeräte mit normativen Benutzermodellen*. Dissertation. Aachen: Shaker Verlag.
- Nirschl, G. (1990): Werkzeug zur Gestaltung und Bewertung von Mensch-Maschine-Dialogen im Kraftfahrzeug. In: *VDI-Gesellschaft Fahrzeugtechnik - Elektronik im Kraftfahrzeug*. Düsseldorf: VDI-Verlag GmbH S. 19-35.
- Schmitt, M. (2000): <http://www.itm.mu-luebeck.de/english/research/specification/sdl2000/index.html>
- Wandmacher, J. (1997): Ein Werkzeug für GOMS-Analysen zur Simulation und Bewertung von Prototypen beim Entwurf. In: G. Szwillus: *Prototypen für Benutzungsschnittstellen - Grundlagen, Techniken, Erfahrungen*. Paderborn S. 35-42.
- Zieren, J. (2000): *Automatische Generierung normativer Benutzermodelle aus SDL-Spezifikationen*, Diplomarbeit, Lehrstuhl für Technische Informatik, RWTH Aachen.