

# Nutzung von Aufgabenmodellen zur Usability-Evaluation – Konzept und Erfahrungen

**Julian Dabbert**  
Universität Paderborn  
AG Szwillus  
Fürstenallee 11  
33102 Paderborn  
dabbert@upb.de

**Gerd Szwillus**  
Universität Paderborn  
Institut für Informatik  
Fürstenallee 11  
33102 Paderborn  
szwillus@upb.de

## Abstract

Die Usability-Evaluation von Software kann mit unterschiedlichen Methoden durchgeführt werden, abhängig vom geforderten Umfang der Ergebnisse sowie den zu investierenden Mitteln. Für eine kosteneffektive Lösung wird in dieser Arbeit der Einsatz von Aufgabenmodellen vorgeschlagen, deren werkzeugunterstützte Analyse eine Aussage zur Usability einer Benutzeroberfläche gestattet.

Da eine solche Untersuchung bereits in einem frühen Entwurfsstadium der Benutzeroberfläche erfolgen kann, kann durch frühzeitiges Erkennen von Problemen ein hohes Maß an Usability zu vergleichsweise geringen Projektkosten erzielt werden. Andere Vorteile: keine ressourcenaufwendigen Benutzertests, weniger Einfluss von Subjektivität, einfache Adaptierbarkeit an beliebige spezielle Zielgruppen.

## Keywords

Aufgabenmodellierung, Usability-Evaluation, Werkzeugunterstützung, Workfloworientierung, GOMS-Keystroke-Modell

## 1.0 Einleitung

Aufgabenmodelle werden zunehmend als Hilfsmittel beim Entwurf interaktiver Systeme eingesetzt. Sie erlauben die frühe Spezifikation der Arbeitsabläufe einer Applikation aus Sicht des Benutzers und ermöglichen somit ein profundes Vorabverständnis des Entwicklers über die Ziele des späteren Benutzers. Im Folgenden wird ein Ansatz vorgestellt, der es darüber hinaus erlaubt, die Usability der Umsetzung einer interaktiven Applikation anhand eines geeignet erweiterten Aufgabenmodells zu untersuchen und zu bewerten. Dies erlaubt die Aufdeckung von Usability-Schwachstellen, sowie die begründete Abwägung zwischen Entwurfsalternativen ohne eine vorherige Implementierung. Das Verfahren wurde erfolgreich bei der Überarbeitung des User Interface einer komplexen Applikation der dSPACE GmbH eingesetzt und insofern in der industriellen Praxis an einer Anwendung mit typischer Größe erprobt.

## 2.0 Aufgabenmodellierung

Der Begriff Aufgabenmodell (englisch: „task model“) beschreibt die Zerlegung einer Aufgabe in eine hierarchisch strukturierte Menge von disjunkten Unteraufgaben. Diese Unteraufgaben können rekursiv weiter zerlegt werden und schließen mit ihrer vollständigen Bearbeitung auch die Erledigung der ursprünglichen Aufgabe ein.

Eine wesentliche Einflussgröße bei der Aufgabenmodellierung ist der Grad an Konkretheit der Teilaufgaben an den Blattknoten des Aufgabenbaumes. Dieser ist nicht explizit durch die Modellierungstechnik festgelegt (Hackos & Redish 1998), sondern liegt in der Entscheidung des Modellierers. So können Aufgabenmodelle so abstrakt formuliert sein, dass die Aufgabenerledigung durch den Benutzer modelliert wird, ohne Eigenschaften der Bedienschnittstelle einzubeziehen (Biere et al 1999). Das andere Extrem ist die

Formulierung derart konkreter Bedienschnitte im Rahmen der Aufgabenmodellierung, dass eine automatische Darstellung bzw. Generierung einer Bedienschnittstelle ermöglicht wird (Caffiau et al 2007, Mori et al 2004, Reichart et al 2008). Der Einfachheit halber sprechen wir in dieser Arbeit vom abstrakten bzw. konkreten Aufgabenmodell, wobei wir uns dabei der bestehenden Abgrenzungsproblematik durchaus bewusst sind.

Eine zum Abstraktionsgrad orthogonale Dimension ist die Beschreibung der aktuell gegebenen Situation (IST-Modell), im Gegensatz zu einem geplanten zukünftigen Zustand (SOLL-Modell) der Aufgabebearbeitung. Wir gehen hier von einer Situation aus, in der das abstrakte IST-Modell zunächst beschrieben und anschließend um die Modellierung der Aufgabenerledigung mit der existierenden Bedienschnittstelle (konkretes IST-Modell) erweitert wird. Da in dieser Arbeit auch Vergleiche zwischen Varianten konkreter Ausprägungen von Be-

dienoberflächen angestellt werden sollen, die auf dem gleichen abstrakten Aufgabenmodell beruhen, betrachten wir darüber hinaus auch ein oder mehrere konkrete SOLL-Modelle.

### 3.0 Generische Interaktionsschritte

Konkrete Aufgabenmodelle im hier verstandenen Sinne beinhalten sowohl die kognitiven Schritte (Überlegungen, Entscheidungen, Informationsermittlung) bzw. generell die Aktionen des Benutzers für die Aufgabenerledigung bis hinunter zur feingranularen Bedienung des Systems. Somit beschreiben die tiefen Ebenen des Aufgabenbaumes die Interaktionsschritte (etwa Tastendrucke, Scrolling, Drag-and-Drop) des Benutzers an einer konkreten Bedienschnittstelle.

Die Effizienz des eigentlich intendierten, „abstrakteren“ Arbeitsflusses wird durch die Notwendigkeit der Bedienung eingeschränkt, wenn der Anwender mentale oder physische Leistungen aufwenden muss, die nicht Teil der Aufgabe selbst sind, sondern erst durch die Benutzerführung entstehen. Das Ziel der Methode ist es, angelehnt an die Struktur des konkreten Aufgabenmodells diese Aufwände aufzuspüren und zu quantifizieren.

Dazu werden typische generische Interaktionsschritte (Tätigen von Eingaben, Orientierung auf der Oberfläche sowie Interpretieren von Ausgaben) charakterisiert und jeweils an den Stellen des konkreten Aufgabenmodells eingefügt, wo sie durchgeführt werden müssen. Im Anschluss werden dann die entstehenden Aufwände gemessen.

Im Rahmen der konkreten Untersuchung einer Applikation der Firma dSPACE entstand die folgende Liste an generischen Interaktionsoperationen, die in diesem Fall auftraten. Es wird hier kein Anspruch auf Vollständigkeit erhoben: Zusätzliches Domänenwissen kann für

einen spezifischen Anwendungsfall die sinnvolle Integration weiterer Aspekte ermöglichen bzw. auch nötig machen. Jede der betrachteten Interaktionsoperationen wird durch ein Schlüsselwort gekennzeichnet und beschrieben, außerdem wird der mentale und zeitliche Aufwand (verzögernd oder beschleunigend) für den Benutzer kurz charakterisiert.

- **ORIENTATION / MENUORIENTATION:** Das mentale Umschalten auf einen neuen sichtbaren Arbeitsbereich bedingt kognitiven Aufwand beim Benutzer. Eine solche Situation ist gegeben, wenn der neue Arbeitsbereich bislang vollständig verborgen war, etwa durch das Öffnen eines Dialogfensters, das Wechseln des aktiven Tabs oder das Auftauchen eines Kontextmenüs.
- **WORDING:** Jeder Begriff, der auf der Benutzeroberfläche auftaucht, bedingt mehr oder weniger viel Erkennungsaufwand, abhängig davon, ob er allgemein verständlich oder als bekannt vorausgesetzter Fachbegriff ist oder nicht.
- **EXIT:** Die Analyse, wie ein Prozess gesteuert, bzw. abgebrochen werden kann, bedingt, dass der Benutzer sich über die Dialogsteuerung Gedanken machen muss.
- **COUNT** bezeichnet die Situation, dass der Benutzer eine große Zahl von Elementen („Chunks“) verkraften muss, um sich in der Bediensituation zurecht zu finden (Miller 1956). Eine zu große Anzahl von zu erfassenden Objekten im Arbeitsbereich kann zu „visuellem Rauschen“ (Krug 2006) führen, was die Arbeitseffizienz herabsetzt und den Benutzer belastet.
- **SCROLLING:** Um nicht-sichtbare Anteile eines größeren Dialogfensters (oder einer Webseite) sichtbar zu machen, muss der Benutzer aktiv – etwa durch Scrollen – den Arbeitsbereich über den unmittelbar sichtbaren Bereich hinaus erweitern.
- **CONVENTION:** Werden dem Benutzer Interaktionstechniken angeboten, die seinen über die Zeit er-

worbenen Kenntnissen und Annahmen widersprechen (egal ob diese bewusst oder unbewusst erlernt wurden) stört dies den Arbeitsfluss des Benutzers und bewirkt wiederum kognitiven Korrekturaufwand.

- **GUIDE:** Eine Senkung des Bedienungsaufwandes kann eintreten, wenn eine Benutzeroberfläche den Benutzer lenkt und unterstützt, etwa indem es ihm Entscheidungen in seinem Arbeitsfluss abnimmt, oder ihm ausschließlich sinnvolle und zulässige Auswahlmöglichkeiten anbietet.
- **FEEDBACK:** Eine Software kann komplexe Systemzustände annehmen, in die ein Benutzer ohne eine ausdrückliche Information des Programms selbst keinen Einblick bekommt. Da die Eingaben eines Benutzers vom System je nach internem Zustand andere Effekte erzielen können, ist eine Rückmeldung für eine effektive Arbeit unumgänglich – deren Angemessenheit entscheidet über den Interpretationsaufwand des Benutzers.
- **METAPHOR:** Wenn das Programm Metaphern verwendet, die vom erwarteten Zielpublikum nicht oder falsch verstanden werden, entsteht beim Benutzer kognitiver Aufwand, um die Abweichung des Verhaltens vom beschriebenen Objekt zur Metapher zu erkennen.
- **SHORTCUT:** Ein schnelleres, aber auch flüssigeres Abarbeiten der Aufgabe kann das Angebot von „Shortcuts“ bedeuten.
- **ICON:** Werden dem Benutzer in der grafischen Darstellung auch symbolische Darstellungen (etwa Icons) angeboten, müssen diese interpretiert und verstanden werden (ähnlich dem WORDING im textuellen Fall).

Diese Auswahl an generischen Interaktionsoperationen wird im Anschluss verwendet, um in einem konkreten Aufgabenmodell – das ja die Arbeit an einer wohldefinierten Bedienschnittstelle beschreibt – die jeweiligen Stellen zu kennzeichnen, an denen der Benutzer diese Interaktionsoperation ausführen muss bzw. kann, oder mit einer entsprechenden Ausgabe konfrontiert wird. Wir sprechen hier kurz von sogenannten

Keytasks, die in Tabelle 1 noch einmal übersichtlich aufgelistet sind.

#### 4.0 Die Methode

Die Absicht dieser Arbeit ist eine toolunterstützte Analysetechnik zur Evaluation von Usability in Softwareprogrammen. Das entwickelte Verfahren überprüft die Einhaltung von bestimmten Usabilitykriterien in einem modellierten Arbeitsablauf des zu untersuchenden Programms. Dadurch wird erreicht, dass der Prozess der Usability Evaluation nicht von der subjektiven Erfahrung eines spezifischen Testers abhängig ist. Stattdessen können entlang eines ermittelten Workflows mit einer Software die Probleme lokalisiert werden, welche einem Benutzer die Bedienung erschweren können, und diese Problemstellen einem zielgerichteten Redesignprozess zur Korrektur übergeben werden. Eine solche Evaluation kann bereits in einem frühen Planungsstadium der Benutzeroberfläche erfolgen, da keine Implementierung erforderlich ist, sondern eine informelle Fixierung des Bedienkonzepts, etwa durch Papierprototypen, ausreicht.

#### 4.1 Abstraktes Aufgabenmodell

Als erster Schritt ist die Erfassung des allgemeinen Arbeitsflusses nötig, in den typische Benutzungsszenarien (Workflows) eingebettet werden können. Aus diesem Arbeitsfluss wird ein entsprechendes Aufgabenmodell (abstraktes IST-Modell) extrahiert, indem die einzelnen Aufgaben in passende Hierarchieebenen unter geeigneten, zu erzeugenden Oberaufgaben eingefügt werden. Für die Modellierung wurde in diesem Fall das Werkzeug CTTE (Mori et al 2002) verwendet, der Arbeitsfluss ist jedoch auch in anderen Modellierungssprachen mit entsprechenden Änderungen analog durchführbar. Dabei ist zu beachten, dass diese Modellierungs-

phase die allgemeine Aufgabe als solche erfasst und von einer konkreten Ausprägung in einem bestimmten Werkzeug (bzw. einer bestimmten Benutzungsschnittstelle) unabhängig bleibt.

werden, dass der Benutzer eine Personenliste öffnet, indem er auf einen entsprechenden Button klickt, dort mithilfe von Scrollen eine Person sucht und auf diese Eintragung doppelklickt. Insgesamt entsteht das konkrete IST-Modell.

CONVENTION	Die Software verhält sich nicht konform zu gegebenen Konventionen in seinem Feld
COUNT	Die Anzahl der Elemente in Eingabebereich zu groß für intuitive Erfassung
EXIT	Der Dialog hat keinen wählbaren oder eindeutig erkennbaren Ausstiegspfad zum Abbruch
FEEDBACK	Der Benutzer erhält über den Status einer <i>relevanten</i> Aktion keine sichtbare Rückmeldung
GUIDE	Der Benutzer wird nicht durch Eingabeformate geführt, etwa Textfeld statt Selektor
ICON	Ein intuitiv verständliches Icon beschleunigt die Wiedererkennung eines Steuerelements
MENUORIENTATION	Die Orientierung des Benutzers in aufgeblendetem Menü erfordert Zeit
METAPHOR	Eine unpassende Metapher verwirrt den Benutzer
ORIENTATION	Die Orientierung des Benutzers auf einer neuen Seite erfordert Zeit im Arbeitsablauf
SCROLLING	Der Benutzer muss seinen Sichtbereich scrollen, um ein benötigtes Element zu finden, was auch Orientierungsbedarf erzeugt
SHORTCUT-X	Ein angebotener und dokumentierter Shortcut bietet eine Ersparnis von x Eingabeschritten
WORDING	Verwendung eines Begriffes, der für das Zielpublikum unverständlich ist

Tabelle 1: Keytasks im Aufgabenmodell

#### 4.2 Konkretes Aufgabenmodell

Im nächsten Schritt wird das gefundene abstrakte IST-Aufgabenmodell „konkretisiert“, indem die Blattknoten des Modells jeweils um einen entsprechenden Unterbaum erweitert werden, der die Ausführung der Aufgabe des ursprünglichen, abstrakten Blattknotens an der konkret jetzt betrachteten Bedienschnittstelle beschreibt. So kann etwa die abstrakte Aufgabe „Auswahl einer Person“ in einem Unterbaum derart beschrieben

Die Blattaufgaben dieses Modells repräsentieren somit jeweils eine konkrete Benutzeraktion an der betrachteten Bedienoberfläche.

Das Aufstellen eines konkreten Aufgabenmodells anhand einer gegebenen Schnittstelle ist durchaus aufwendig. So hatte das konkrete IST-Modell im Anwendungsfall ca. 400 Knoten. Andererseits ist die Entwicklung dieses Modells in jedem Einzelschritt sehr einfach, sodass eine automatische oder semi-automatische Unterstützung dieses Prozesses möglich und auch angedacht ist.

### 4.3 Anreicherung um Keytasks

Der entscheidende Schritt in dieser Phase besteht in der Anreicherung dieses Modells um generische Interaktionsaufgaben. Das konkrete Aufgabenmodell beinhaltet also neben den konkreten Handlungen des Benutzers auf feingranularer Ebene auch die beim Benutzer auftretenden mentalen Aufwände bzw. Aufgaben in Form von Keytasks. Der reinen, elementaren Interaktion wird dazu der „passende“ Keytask vor- oder hintangestellt. Diese Ergänzung ist durch den Detailgrad und dadurch die pure Größe des konkreten Aufgabenmodells bei einem komplexen Arbeitsfluss natürlich aufwendig, aber im Einzelnen nicht schwierig. Vielmehr gibt es einfache Regeln, nach denen die Keytasks einzufügen sind. Exemplarisch seien hier die folgenden Regeln genannt:

- Wird das aktuelle Fenster oder der Reiter gewechselt, so füge den Keytask ORIENTATION ein
- Wird ein (Kontext-)Menü über die Seite geblendet, so füge den Keytask MENUORIENTATION ein
- Wird ein der Zielgruppe unerwarteter Begriff verwendet, füge den Keytask WORDING ein

Technisch wird das Einfügen der Keytasks nicht in einer separaten Phase, sondern gemeinsam mit dem Aufstellen des konkreten IST-Aufgabenmodells gemäß Abschnitt 4.2 ausgeführt. Dies ist insofern sinnvoll, dass beim Einfügen der konkreten Interaktion die Arbeitsabläufe in der Bedienoberfläche ohnehin im Detail betrachtet werden müssen.

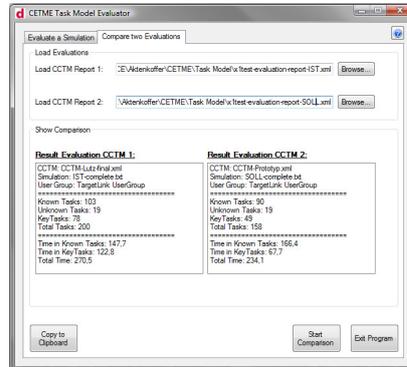


Abbildung 1: Auswertung der Simulation

Insgesamt besteht nach diesem Schritt daher ein konkretes Aufgabenmodell, das die genauen Aktionsfolgen an einer gegebenen (oder geplanten) Bedienoberfläche spezifiziert und zusätzlich den Aufwand des Benutzers beim Arbeiten mit dieser Oberfläche durch die eingestreuten Keytasks beschreibt.

### 4.4 Auswertung

Das so entstandene Modell wird für die Usability-Evaluation eingesetzt, indem typische, wichtige oder häufige Benutzungsszenarien („user stories“) auf dem Modell durchlaufen werden. CTTE bietet hierzu einen sogenannten Simulator (Mori et al 2002) an, der einen schrittweisen Durchlauf durch das Modell erlaubt. Bei dieser Simulation werden die durchlaufenen Schritte einschließlich der Keytasks in eine Protokolldatei geschrieben, die als Eingabe für das Analysewerkzeug CETME (s.

Abbildung 1) dient. Dieses Tool analysiert, wie viel Zeit der (simulierte) Benutzer mit welchen Keytasks verbracht hat. Dazu werden frei konfigurierbare Benutzerprofile verwendet, die je nach Zielgruppe einen unterschiedlichen Zeitbedarf für die Erledigung von bestimmten Aufgaben veranschla-

gen, was analog zum GOMS Keystroke Verfahren (Bonnie & Kieras 1996) eine Abbildung von Arbeit auf Zeit ermöglicht. In der Auswertung sind die für den Benutzer aufwendigen Problembereiche sofort erkennbar.

Auch der Vergleich zweier verschiedener Bedienoberflächen zum gleichen abstrakten Modell, also zur gleichen Aufgabe, ist quantitativ und qualitativ möglich. Dazu werden die verschiedenen Aufgaben, die im Verlauf der Simulation abgearbeitet werden, gesammelt und nach Aufgabentyp sowie gegebenenfalls aus dem Benutzerprofil bekannter Ausführungsdauer klassifiziert. Diese Datenbasis ist der Schlüssel für den Vergleich zweier verschiedener Benutzeroberflächen miteinander, da in direkter Gegenüberstellung die benötigten Aufgabenschritte auf Unterschiede in den jeweiligen Arbeitspfaden hindeuten können. Auf diese Weise kann der Fortschritt eines Überarbeitungsprozesses einer Benutzeroberfläche nachvollzogen beziehungsweise eine fundierte Entscheidung für eines der beiden verglichenen Oberflächenkonzepte getroffen werden.

### 5.0 Praktische Erfahrungen

In einem praktischen Experiment mit einer Applikation der dSPACE GmbH konnte auf diese Weise eine deutliche Verbesserung der Bearbeitungszeiten sowie eine Vereinfachung der Benutzung erzielt werden (wie in

Abbildung 2 schematisch angedeutet). Die Analyse der Aufgabenpfade auf der vorhandenen Oberfläche zeigte auf, dass über 20% der abgeschätzten Zeit auf solche Aufgaben entfallen, die durch eine Veränderung der Benutzerführung reduziert werden können. Dazu zählen insbesondere die modale Veränderung des gesamten Arbeitsbereiches beim

Tabwechsel sowie die Überladung des Platzes durch eine Überzahl von Funktionselementen.

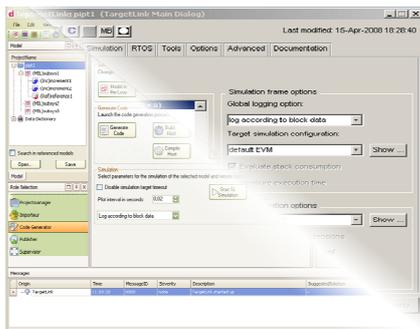


Abbildung 2: Vergleich Nachher/Vorher

In einem Beispielszenario ergab sich durch das Redesign sich ein wesentlicher Unterschied in der gesamten Ausführungsdauer des gewählten Arbeitsablaufes: Von 270,5 Sekunden wurde diese auf 234,1 Sekunden reduziert, eine Einsparung von etwa 14 Prozent. Dies liegt ebenso an einer um 21 % geringeren Anzahl an Klicks, die von 89 auf 76 sank, wie auch an einer deutlichen Verringerung vermeidbaren Keytasks. Die übrigen Tasks unterschieden sich nicht, was darauf zurückzuführen ist, dass beide Szenarien denselben Workflow als Grundlage haben.

Dieses Ergebnis zeigt die Auswirkungen von wesentlichen Veränderungen in der Benutzerführung der Oberfläche der untersuchten Applikation. Diese Änderungen sind im Wesentlichen auf eine Neugruppierung der Funktionen zurückzuführen, wodurch seltener eine Neuorientierung des Benutzers im Arbeitsbereich erforderlich ist. Zudem werden auch weniger Eingabeoperationen im Bereich der Benutzerführung notwendig.

Eine systematische Evaluation der erzielten Verbesserungen steht noch aus. Es sei hier zur Veranschaulichung jedoch der *Group Leader Software Development Target-Link User Interface and*

*Simulation* bei dSPACE zitiert: "Die Anwendung des Verfahrens führte zu einer spürbaren Verbesserung in der Usability an einem Prototyp unserer Benutzeroberfläche".

## 6.0 Schlussbemerkungen

Insgesamt entsteht ein einfach anwendbares und erweiterbares Verfahren, das sich an etablierten Konzepten orientiert, Usability-Betrachtungen stärker als andere Methoden aus dem Subjektiven herauslöst und damit effizient und erfolgreich auf Problembereiche im Design aufmerksam machen kann, um anschließend diese Probleme in einem Neuentwurf gezielt lösen zu können.

Als Ausgangspunkt für weitere Arbeiten bieten sich Erweiterungen auf andere Bereiche an, etwa die Evaluation von Arbeitsflüssen auf Webseiten. Dabei ist eine Kooperation mit dem in (Szwilius 2006, Baule 2008) vorgestellten Verfahren denkbar. Weiterhin kann das Verfahren einfach für den Einsatz in speziellen Anwendungsbereichen erweitert werden, um bereichsspezifische Qualitäten einer Benutzeroberfläche modellieren zu können. So könnten Softwareprodukte im Bereich der assistierenden Technologien für die besonderen Anforderungen von Menschen mit körperlichen oder geistigen Einschränkungen, wie beispielsweise komplexen Mehrfachbehinderungen, evaluiert und optimiert werden.

## 7.0 Literaturverzeichnis

Baule, D. (2008): Konzeption und Implementierung eines Prototypen zur Simulation aufgabenangemessener Abläufe im Web am Beispiel der Modellierungsumgebung WISE, Universität Paderborn, Diplomarbeit.

Biere, M.; Bomsdorf, B.; Szwilius, G. (1999): Building and Simulating Task Models with VTMB. In: CHI'99, Pittsburgh, CHI'99 Extended Abstracts, pp. 1-2.

Bonnie, J.; Kieras, D. E. (1996): The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast, ACM Transactions on Computer-Human Interaction 3,4, 320-351.

Caffiau, S.; Girard, P.; Scapin, D.; Guittet, L. (2007): Generating Interactive Applications from Task Models: A Hard Challenge, Lecture Notes in Computer Science, Springer Berlin / Heidelberg

Hackos, J.; Redish, J. (1998): User and Task Analysis for Interface Design, John Wiley & Sons.

Krug, S. (2006): Don't Make me Think!, mitp-Verlag.

Miller, G.A. (1956): The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information, Psychological Review, 63, S.81-97.

Mori, G.; Paternò, F.; Santoro, C. (2002): CTTE: Support for Developing and Analyzing Task Models for Interactive System Design, IEEE Transactions on Software Engineering, August 2002, pp.797-813.

Mori, G., Paternò, F., Santoro, C. (2004): Design and Development of Multidevice User Interfaces through Multiple Logical Descriptions, IEEE Transactions on Software Engineering, August 2004, pp.507-520.

Reichart, D.; Dittmar, A.; Forbrig, P.; Wurdel, M.: Unterschiedliche Präsentationen von Aufgabenmodellen, Usability Professionals Tagung, Lübeck, September 2008.

Szwilius, G.(2006): Modellierung aufgabenangemessener Abläufe im Web. In: A. M. Heinecke, H. Paul (Hrsg.): Mensch & Computer 2006: Mensch und Computer im StrukturWandel, München: Oldenbourg Verlag.

## User Experience: Was ist das eigentlich?