

# Kognitive Modellierung in der Konzeption und Evaluation von Softwareprodukten

## Dr. Theo Held

SAP User Experience  
Applications, CRM  
SAP AG  
Dietmar-Hopp-Allee 16  
69190 Walldorf  
theo.held@sap.com

## Dr. Martin Schrepp

SAP User Experience  
Applications, CRM  
SAP AG  
Dietmar-Hopp-Allee 16  
69190 Walldorf  
martin.schrepp@sap.com

## Dietrich Mayer-Ullmann

SAP User Experience  
Applications, CRM  
SAP AG  
Dietmar-Hopp-Allee 16  
69190 Walldorf  
dietrich.mayer-ullmann@sap.com

## Daniel Holt

Psychologisches Institut  
Universität Heidelberg  
Hauptstraße 47-51  
69117 Heidelberg  
daniel.holt@psychologie.uni-heidelberg.de

## Abstract

Kognitive Modellierung ist eine sehr hilfreiche Verfahrensweise im Prozess des Produktdesigns. Grundsätzlich geht es darum, möglichst fundierte und präzise Vorhersagen über den zu erwartenden zeitlichen (und auch kognitiven) Aufwand eines geübten Nutzers beim Durchlaufen spezifischer Interaktionsaufgaben zu machen. Derartige Vorhersagen können die Arbeit von Interaktionsdesignern erheblich erleichtern,

wenn es darum geht, Designvarianten miteinander zu vergleichen. Zur Anwendung steht eine Reihe prominenter Modellierungsansätze zur Verfügung. Wir werden hier die Familie der GOMS-Modelle und das ACT-R basierte CogTool vorstellen. Als wichtige Ergänzung stellen wir auch eine Möglichkeit vor, wie die zeitliche Dauer elementarer Operationen effizient und präzise bestimmt werden kann.

## Keywords

Kognitive Modellierung, GOMS, KLM, User Performance

## 1.0 Einführung

Interaktionen mit Softwareprodukten setzen sich üblicherweise aus immer wiederkehrenden elementaren Operationen zusammen. Beispiele dafür sind mehr oder weniger unmittelbar beobachtbare Handlungen, wie z.B. Mausbewegungen, Tastaturanschläge oder auch Änderungen der visuellen Fokussierung. Dazu kommen vielfältige kognitive Operationen wie Verlagerungen der Aufmerksamkeit, Nachdenken oder visuelle Suchen.

Ausgehend von „Low Level“ Modellen der kognitiven Architektur, z.B. ACT-R (Anderson & Lebiere, 1998), entstand eine Reihe vereinfachter und in der Praxis gut verwendbarer Ansätze zur kognitiven Modellierung, die Aussagen über die Zusammenhänge zwischen den elementaren Operationen machen.

Prinzipiell geht es bei allen bekannten Modellierungsansätzen darum, elementare Operationen, die zur Erledigung einer Aufgabe erforderlich sind, zu iden-

tifizieren. Diesen Operationen werden Zeiten zugeordnet und daraus wird die Gesamtzeit für eine Aktion, die sich aus den elementaren Operationen zusammensetzt, ermittelt. Solche Analysen sind im Rahmen der Softwareproduktion sehr hilfreich, wenn es gilt, unterschiedliche konzeptionelle Ansätze zu vergleichen, bevor eine Designentscheidung getroffen wird, bzw.

wenn bereits existierende Produkte mit anderen Produktvarianten oder auch Konkurrenzprodukten verglichen werden sollen. Die vergleichenden Analysen vermitteln Klarheit darüber, welche und wie viele elementare Operationen für das Erledigen typischer Aufgaben erforderlich sind und mit welchen geschätzten zeitlichen Aufwänden (eines geübten Nutzers) insgesamt gerechnet werden muss. Es sind üblicherweise Alternativen zu bevorzugen, die einen geringeren kognitiven Aufwand erfordern und gleichzeitig weniger Zeit benötigen.

In unserem Beitrag werden zwei „High Level“ Ansätze der kognitiven Model-

lierung vorgestellt, die im Rahmen der Produktion von Business-Software eingesetzt werden. Wir diskutieren, wie sich die Ergebnisse einer Modellierung gewinnbringend in den Design- und Entwicklungsprozess integrieren lassen und welcher Beitrag zur Verbesserung der Software-Qualität damit geleistet wird.

Als zwei prominente Vertreter der High Level Ansätze habe wir GOMS (Card, Moran & Newel, 1983) und das ACT-R basierte CogTool (John & Salvucci, 2005) gewählt. Anhand dieser Vertreter werden auch die Limitationen der Anwendbarkeit und generelle Nachteile der Modellierungsansätze diskutiert.

Als Ergänzung der Modellansätze stellen wir zudem ein Software-Tool vor, das die für die Modellierung erforderlichen Zeiten ermittelt. Das „User Performance Measurement Tool“ zeichnet sich aus durch eine umfangreiche Konfigurierbarkeit und eine gute Anpassbarkeit an sehr unterschiedliche Stufen der Granularität der zu untersuchenden Operationen.

## 2.0 GOMS Modelle

GOMS ist eine quantitative Methode zur Bewertung der Effizienz von Benutzerschnittstellen. Eine GOMS Analyse erlaubt vorherzusagen, wie lange ein erfahrener Benutzer für die Bearbeitung einer vorgegebenen Aufgabe in einer vorgegebenen Benutzerschnittstelle benötigt. GOMS ist daher geeignet verschiedene alternative Designvarianten bzgl. ihrer Effizienz zu vergleichen.

Die Abkürzung GOMS steht für „Goals“, „Operators“, „Methods“ und „Selection Rules“. Dies sind die wesentlichen Komponenten eines GOMS Modells:

- Ein Ziel (Goal) beschreibt, was der Benutzer erreichen möchte.
- Operatoren sind grundlegende physische oder kognitive Prozesse, die der Benutzer zur Erreichung des Ziels ausführen muss.
- Methoden repräsentieren erlernte Operator-Sequenzen, die ein Benutzer automatisch ausführt, um ein Ziel oder Teilziel zu erreichen.
- Stehen mehrere Methoden zur Verfügung, so entscheiden Selektionsregeln, welche Methode ausgeführt wird.

Beispiele für typische physische Operatoren in GOMS Modellen sind das Drücken einer Taste, die Bewegung der Hand von der Maus zur Tastatur oder das Positionieren des Mauszeigers. Beispiele kognitiver Operatoren sind der Abruf einer Information aus dem Langzeitgedächtnis, die Entscheidung zwischen zwei Alternativen oder die mentale Vorbereitung für den nächsten Schritt in einer Sequenz.

Verschiedene Personen benötigen natürlich unterschiedliche Zeiten für diese Operationen. Für einen Vergleich zwischen mehreren alternativen Entwürfen ist es aber ausreichend von der konkreten Person zu abstrahieren und auf typische Durchschnittswerte zurückzugreifen. Solche Durchschnittswerte werden in experimentellen Studien ermittelt (z.B.

John & Kieras, 1996; Olson & Olson, 1990; Schrepp & Fischer, 2007). Auch das unten vorgestellte UPM-Tool kann dazu eingesetzt werden.

Beispiele verwendeter Durchschnittszeiten sind:

- Tastendruck beim Tippen einer Zeichenkette (0,2 Sekunden)
- Positionieren des Mauszeigers (1,1 Sekunden)
- Mentale Vorbereitung für den nächsten Schritt einer Sequenz (1,35 Sekunden)

Es sind mehrere Varianten von GOMS vorhanden. Zum Beispiel das „Critical Path Method GOMS“ (CPM-GOMS), die „Natural GOMS Language“ (NGOMSL) oder das „Keystroke Level Model“ (KLM-GOMS). Diese Varianten unterscheiden sich in der Komplexität der Modellierung und damit bzgl. der Genauigkeit der geschätzten Zeiten.

Das Keystroke Level Modell ist die einfachste Form der GOMS Analyse. Diese Variante beschränkt sich bei der Vorhersage der Bearbeitungszeit einer Aufgabe allein auf die dafür notwendigen Operatoren. Alternative Bearbeitungssequenzen, die über Selektionsregeln gesteuert werden, werden bei dieser Variante nicht verwendet. Das führt zu einer erheblichen Vereinfachung des Aufwands für die Modellierung. Deshalb ist diese Variante für die praktische Anwendung in konkreten Designprojekten besonders geeignet.

Bei der Analyse einer Aufgabenstellung mit KLM-GOMS ist es in der Regel sehr einfach, die Sequenz aller notwendigen physischen Operatoren zu bestimmen. Wesentlich schwieriger ist die korrekte Platzierung kognitiver Operatoren. Typischerweise wird dies gelöst, in dem man die Aufgabenstellung in mehrere Teilaufgaben zerlegt. Vor jeder Teilaufgabe wird dann ein kognitiver Operator (z.B. für mentale Vorbereitung) platziert. Dies erfordert

ein gewisses Verständnis menschlicher Problemlöseprozesse im analysierten Aufgabengebiet und ist daher der schwierigste Teil einer Analyse mit KLM-GOMS.

Eine wesentliche Einschränkung des KLM-GOMS ist, dass es dieser Ansatz leider nicht erlaubt, parallele Prozesse menschlicher Informationsverarbeitung abzubilden. Bei Aufgaben, in denen solche parallelen Denkprozesse eine zentrale Rolle spielen, werden die aus der KLM-GOMS Analyse resultierenden Zeiten daher die realen Zeiten zur Aufgabebearbeitung überschätzen.

Aufgrund der relativ einfachen Anwendbarkeit von KLM-GOMS eignet sich diese Methode in folgenden Bereichen:

- Vergleich alternativer Designs bezüglich ihrer Effizienz für erfahrene Benutzer.
- Abschätzung, ob Aufgaben in einer Benutzeroberfläche in der vorgegebenen Zeit erledigt werden können.
- Evaluierung von Fragen im Bereich der Barrierefreiheit von Entwürfen. So kann schon mit Entwürfen geprüft werden, ob die spezifizierete Tastaturbedienung im Vergleich zur Mausbedienung ausreichend effizient ist (Schrepp & Hardt, 2007).

## 3.0 CogTool

CogTool wurde an der Carnegie-Mellon-University entwickelt (John et al., 2004; John & Salvucci, 2005). Im Vergleich mit den Varianten von GOMS basiert CogTool auf der „kognitiven Engine“ ACT-R, die von Anderson und Lebiere (1998) vorgestellt wurde. Das bedeutet, dass in die Zeiten, die CogTool liefert, wesentlich mehr Annahmen und Erkenntnisse bezüglich erforderlicher kognitiver Operationen einfließen, als es bei den klassischen GOMS Modellen der Fall ist. Beispiele sind die Operationen „Nachdenken“, „Betrachten“, „bereite nächste Aktion vor“ etc.

Man kann also annehmen, dass das CogTool akkuratere und realitätsnähere zeitliche Schätzungen liefert als GOMS. Nähere Informationen zu CogTool und die Software zum kostenlosen Download finden sich unter [www.cs.cmu.edu/~bej/cogtool/](http://www.cs.cmu.edu/~bej/cogtool/). Wir werden hier nur auf die wesentlichen Schritte zur Anwendung von CogTool eingehen. Die Arbeit mit CogTool beinhaltet folgende wesentliche Aufgaben:

- Erzeugen eines Design-Storyboards mit dessen Hilfe Aufgaben ausgeführt werden können.
- Definition einer Menge von Aufgaben.
- Erzeugen eines Skripts.
- Modifikation des Designs und Wiederholung der weiteren Schritte.

Ein Design-Storyboard besteht aus einer Menge unterschiedlicher Screens oder Screenzustände des zu untersuchenden Systems (z.B. Eingabeformulare, Dashboards, Ergebnistabellen etc.).

Für jeden Screen (oder „Frame“ in der CogTool Terminologie) werden dann „Interaktions-Widgets“ erzeugt. Diese Widgets sind mit den Screen-Elementen assoziiert, die ein Nutzer beim Durchlaufen einer bestimmten Interaktionssequenz (d.h. einer Aufgabe) „anfassen“ würde (z.B. Eingabefelder, Menüs, Check-Boxes etc.).

Einzelne Frames werden dann mit Hilfe von „Übergängen“ (Transitions) miteinander verbunden. Ein Übergang führt von einem Widget eines Frames zu einem anderen Frame. Abbildung 1 zeigt ein Beispiel für ein Design-Storyboard in dem bereits Widgets definiert sind. Die Pfeile stellen die Übergänge dar.

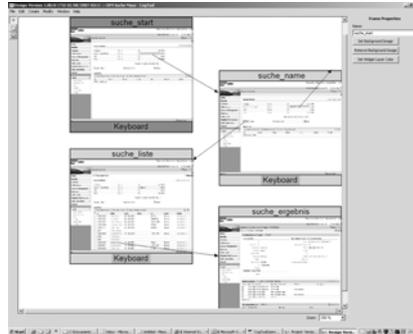


Abb 1: Beispiel für ein CogTool Design-Storyboard mit Widgets und Übergängen.

Nun kommen die Aufgaben ins Spiel. Designs werden natürlich bereits auf der Basis möglicher typischer Interaktionssequenzen zusammengestellt. Häufig lassen sich auf der Grundlage eines gegebenen Designs unterschiedliche Aufgaben durchlaufen. Aus den einzelnen Schritten einer Aufgabe (z.B. „Öffne ein Menü“, „Speichere ein Objekt“, „Navigiere zur Suche“ etc.) kann nun ein Skript aufgezeichnet werden.

Im Vergleich zu einer Aufgabe enthält ein Skript auch Schritte, die automatisch von CogTool hinzugefügt werden (z.B. „Nachdenken“, „visuelle Orientierung“ etc.). In dieser Eigenschaft liegt die besondere Stärke der Funktionsweise von CogTool. De facto wird das Storyboard von CogTool in ein kognitives Modell basierend auf ACT-R übersetzt. Der Nutzer des Tools muss die impliziten, nicht unmittelbar beobachtbaren kognitiven, motorischen oder wahrnehmungsbezogenen Operatoren weder kennen, noch explizit eingeben – sie werden vom Tool geliefert.

Unsere Anwendung von CogTool als Hilfsmittel zum Design von Business-Applikationen hat gezeigt, dass die vorhergesagten Zeiten durchgehend sehr nahe an den Zeiten liegen, die geübte Nutzer benötigen.

Im Rahmen unserer Design-Arbeit wurde und wird CogTool z.B. in den folgenden Kontexten verwendet:

- Vergleich unterschiedlicher Releases eines Softwareproduktes: Wird das Anlegen eines Objekts in Release n+1 wirklich schneller gehen als in Release n?
- Vergleich unterschiedlicher Versionen eines Softwareproduktes: Wird das Anlegen einer Order in der Call-Center Variante des Produkts schneller gehen als in der (funktionsreicheren) Back-Office Version
- Klärung grundsätzlicher Designfragen vor der Implementierung: Wird Designvariante A zu einer höheren Nutzer-Performance führen als Variante B?

Ein sehr wichtiges Feature von CogTool ist das „ACT-R Visualisierungsfenster“. Hier können die gemäß ACT-R angenommenen elementaren Operationen inspiziert werden. Dadurch bekommt der Anwender einen guten Überblick darüber, welcher Art diese Operationen primär sind (z.B. motorische Aktion, visuelle Orientierung, Nachdenken etc.). Abbildung 2 zeigt beispielhaft den Inhalt des Visualisierungsfensters. Einzelne horizontale Balken stehen für die zeitliche Ausdehnung einzelner Arten von Operationen. Zu beachten ist, dass CogTool auch den parallelen Ablauf von Operationen berücksichtigt.



Abb 2: ACT-R Visualisierungsfenster

Ein weiterer interessanter Aspekt ist die Kombination von CogTool Nutzermodellen mit dem reinen Keystroke Level Model-Ansatz, der die Stärken der jeweiligen Systeme nutzt. Dazu werden Interaktionen auf dem Keystroke-Level detailliert mit CogTool modelliert und die so gewonnenen Benutzungszeiten als Makrooperatoren in ein höherstufiges System übertragen, mit dem längere Interaktionssequenzen im Baukasten-system zusammengestellt werden können. Dies verbindet den hohen Auflösungsgrad und einfache Umsetzung der CogTool-Modellierung mit einem pragmatischen Vorgehen zur Modellierung komplexer Interaktionsszenarien.

#### 4.0 Messung elementarer Operationszeiten mit dem User Performance Measurement Tool

Je nach gewähltem Modellierungsansatz und der Auflösung des Modells können die betrachteten Interaktionen aus einem oder mehreren elementaren Operatoren bestehen, denen anschließend jeweils eine geschätzte Dauer zugeordnet wird.

Zu diesem Zweck wurde das Software-Tool UPM (User Performance Measurement) entwickelt, um das Erfassen der Modellparameter zu erleichtern und zu beschleunigen. Hauptfunktion des Tools ist die möglichst exakte, informationshaltige und vollständige Aufzeichnung aller Systemereignisse während der Interaktion eines Nutzers mit einem Softwareprodukt.

Zudem kann das Tool aus aufgezeichneten Kombinationen von Ereignissen bzw. Interaktionspfaden bereits die gesuchten Zeiten für (in Bezug auf einen bestimmten Modellierungsansatz) relevante Operationen bereitstellen.

Das UPM Tool wurde zur Untersuchung von Web-Anwendungen entwickelt und kann durch eine Kalibrierungsfunktion an Änderungen des Programmcodes

angepasst werden. Zudem kann mit der Kalibrierung der „Auflösungsgrad“ für bestimmte Operator-Sequenzen erhöht werden, indem variiert wird, welche Elemente einer Sequenz als eigenständige Ereignisse erfasst werden und welche Elemente mit anderen kombiniert werden.

Durch die Vergabe sogenannter „Klassenattribute“ im Programmcode der zu untersuchenden Bedienoberfläche kann die Bedienung sehr spezifischer oder sogar nur einmalig eingesetzter Bedienelemente ermittelt und so die Interaktion sehr detailliert nachvollzogen werden (z.B. Tabellenzeilen 2,5,10 einer Tabelle wurden mit gedrückter „Strg“-Taste und linker Maustaste selektiert).

Andererseits liefert das Tool auch Daten zu kombinierten Interaktionsschritten, wie z.B. Eingabe-Medienwechsel zwischen Tastatur und Maus.

Neben der Ermittlung von Modellparametern ist das Tool natürlich auch dazu geeignet, modellierte Interaktionsverhalten direkt an Prototypen oder laufenden Systemen nachzutesten, um Modell mit Realität vergleichen zu können. Implementierungs- oder auch Modellierungsfehler (z.B. falsche zugrunde gelegte Annahmen) sind dann schnell auffindbar.

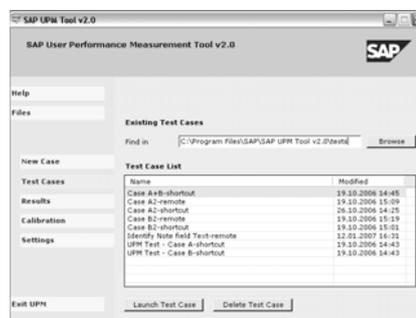


Abb 3: User Performance Measurement Tool (Labor-Version)

Für unterschiedliche Einsatzzwecke und -bedingungen existieren unter-

schiedliche Versionen des Tools. Neben einer „Labor-Version“ mit umfangreichen Funktionen (siehe Abbildung 3) existiert eine kompakte „Online-Version“, die für den Einsatz auf den Rechnern von Endnutzern und das Sammeln von Massendaten konzipiert wurde.

#### 5.0 Diskussion

Kognitive Modellierung wurde von uns im Zusammenspiel mit dem UPM Tool erfolgreich innerhalb des Designprozesses für Business-Anwendungen eingesetzt. Es ist wichtig zu erwähnen, dass kognitive Modellierung wesentliche Beiträge zur Identifikation von Designproblemen leisten und die Entscheidung für das „optimale“ Design maßgeblich unterstützen kann; jedoch sollte es möglichst in Verbindung mit anderen methodischen Ansätzen verwendet werden. Unserer Ansicht nach kann kognitive Modellierung formative Usability Tests nicht ersetzen. Die oft ungeahnten Probleme, die von Endbenutzern „aufgedeckt“ werden, können üblicherweise nicht per Modellierung identifiziert werden.

Wenn es um Designentscheidungen geht, ist der benötigte zeitliche Aufwand für das Bearbeiten von Aufgaben ein wesentlicher Faktor. Andererseits ist bekannt, dass neben der Zeit auch andere Einflüsse auf die „User-Experience“ berücksichtigt werden müssen. Einer dieser Faktoren sind subjektive Präferenzen, die eher mit der Attraktivität einer Nutzungsschnittstelle als mit der zeitlichen Effizienz zu tun haben. Für die Identifikation solcher Präferenzen sind z.B. psychometrische Skalierungsverfahren (siehe Held et al., 2007) oder Fragebögen geeignet (siehe z.B. Laugwitz et al., 2006).

Auch wenn CogTool (und auch GOMS Editoren) den Nutzer beim Aufbau des Modells weitgehend unterstützt, sind für die Anwendung kognitiver Modellie-

rungsverfahren gewisse Grundkenntnisse aus den Bereichen der Modellierung und der kognitiven Psychologie von Vorteil. Wir empfehlen deshalb, zumindest bei den ersten Modellierungsprojekten Experten zu Rate zu ziehen.

## 6.0 Literaturverzeichnis

- Anderson, J.R.; Lebiere, C. (1998): *The Atomic Components of Thought*. Mahwah: Lawrence Erlbaum Associates.
- Card S.; Moran T.P.; Newell A. (1983): *The Psychology of Human Computer Interaction*. Mahwah: Lawrence Erlbaum Associates.
- Held, T.; Schrepp, M.; Fischer, P. (2007): Skalierung von Designalternativen. In: T. Gross (Hrsg.): *Mensch & Computer 2007*, 9–18. München: Oldenbourg Verlag.
- John, B.E.; Kieras, D.E. (1996): The GOMS family of user interface analysis techniques: Comparison and Contrast. *ACM Transactions on Computer-Human Interaction*, 3, 4, S. 320-351.
- John, B.; Prevas, K.; Salvucci, D.; Koedinger, K. (2004): Predictive Human Performance Modeling Made Easy. *Proceedings of CHI, 2004* (Vienna, Austria, April 24-29, 2004) ACM, New York.
- John, B. E.; Salvucci, D. D. (2005): Multi-Purpose Prototypes for Assessing User Interfaces in Pervasive Computing Systems. *IEEE Pervasive Computing* 4 (4), 27-34.
- Laugwitz, B.; Schrepp, M.; Held, T. (2006): Konstruktion eines Fragebogens zur Messung der User Experience von Softwareprodukten. In: A.M. Heinecke & H. Paul (Hrsg.): *Mensch & Computer 2006*, 125–134. München: Oldenbourg Verlag.
- Olson, J.R.; Olson, G.M. (1990): The growth of cognitive modelling in human-computer interactions since GOMS. *Human-Computer Interaction*, 5, S. 221-265.
- Schrepp, M.; Fischer, P. (2007): GOMS models to evaluate the efficiency of keyboard navigation in web units. *Eminds – International Journal of Human Computer Interaction* Vol. 1, No. 2, S. 33-46.
- Schrepp, M.; Hardt, A. (2007): GOMS models to evaluate the quality of an user interface for disabled users. In: Eizmendi, G., Azkoita, J.M. & Craddock, G.M. (Eds.), *Challenges for Assistive Technology*, S. 646-651.