

Anwendung von GOMS-Analysen und CogTool in der Design-Praxis

Martin Schrepp, Theo Held

SAP AG Walldorf

Zusammenfassung

Kognitive Modellierung ist eine kostengünstige Methode betriebswirtschaftliche Anwendungen hinsichtlich ihrer Effizienz zu evaluieren und das Design solcher Anwendungen weiter zu optimieren. Wir untersuchen in diesem Beitrag mit KLMGOMS und CogTool zwei etablierte Methoden der kognitiven Modellierung in Bezug auf ihre Fähigkeit die Bearbeitungszeiten betriebswirtschaftlicher Arbeitsabläufe korrekt vorherzusagen. Die Ergebnisse zeigen, dass die mit Hilfe von CogTool generierten Vorhersagen genauer sind, als die aus einer GOMS-Analyse resultierenden Vorhersagen. Allerdings erfordert die Modellierung mit CogTool schon deutlich detaillierter ausgearbeitete Entwürfe als dies für eine GOMS-Analyse notwendig ist. Aus diesen Ergebnissen können Empfehlungen für den Einsatz und die Einschränkungen beider Methoden abgeleitet werden.

1 Einleitung

Effizienz ist ein wesentliches Qualitätsmerkmal interaktiver Produkte. Speziell bei betriebswirtschaftlich genutzten Anwendungen hat die Effizienz der Benutzerschnittstelle einen starken Einfluss auf die erfolgreiche Einführung eines Produkts in einem Unternehmen. Solche Anwendungen werden in der Regel zur Erledigung betriebswirtschaftlicher Standardprozesse (z.B. Erstellen eines Angebots oder Erfassen eines neuen Auftrags) eingesetzt, die ein einzelner Benutzer während eines typischen Arbeitstages sehr häufig durchführt. Selbst kleinere Zeitverluste, die durch eine ineffiziente Gestaltung der Benutzerschnittstelle entstehen, können dadurch zu ernstzunehmenden Produktivitätsverlusten führen.

Designer betriebswirtschaftlicher Anwendungen haben daher in der Regel ein starkes Interesse daran, die Effizienz einer Benutzerschnittstelle zu messen. Als Effizienzmaß wird typischerweise die Zeit verwendet, die ein erfahrener Benutzer zur Erledigung von typischen Standardaufgaben mit der Benutzerschnittstelle benötigt.

Eine solche zeitliche Schätzung kann zum einen dafür verwendet werden, die Eignung der Benutzerschnittstelle in Bezug auf vorgegebene maximale Bearbeitungszeiten in einem Kundenszenario zu überprüfen. Zum Beispiel gibt es in Call Centern oft klare Vorgaben, wie

lange die Erfassung eines telefonisch eingegangenen Auftrags dauern darf. Vor Einführung einer neuen Call Center Software in einem Unternehmen ist es also notwendig zu prüfen, ob diese Vorgaben erreicht werden können. Zum anderen ist eine solche Quantifizierung der Effizienz notwendig, um schon in der Design-Phase einer Anwendung alternative Entwürfe einer Benutzerschnittstelle vergleichen zu können bzw. einen bestehenden Entwurf hinsichtlich seiner Effizienz weiter zu optimieren.

Wie kommt man nun zu einer solchen quantitativen Beurteilung der Effizienz einer Benutzerschnittstelle? Eine Möglichkeit sind natürlich Benutzertests. Allerdings benötigt man für ausreichend genaue Zeitmessungen bereits ein lauffähiges System. Für die Evaluation während der Design-Phase ist diese Methode daher nicht geeignet. Für die Beurteilung der Effizienz ist man in der Regel nicht an den Zeiten interessiert, die ungeübte Benutzer für die Aufgabenbearbeitung benötigen. Man muss daher sicherstellen, dass die Teilnehmer sich schon hinreichend mit dem System vertraut gemacht haben, bevor man die Zeiten misst. Dies erfordert natürlich eine längerfristige Interaktion der Teilnehmer mit der Anwendung und treibt daher den Aufwand und die Kosten eines solchen Tests in die Höhe.

Kognitive Modellierung erlaubt es, den zeitlichen Aufwand vorgegebener Arbeitsabläufe in einer Benutzerschnittstelle zu schätzen ohne Testpersonen einzubeziehen. Diese Methode kann auch schon ausgehend von groben Entwürfen in der frühen Design-Phase eines Projekts verwendet werden. Es gibt verschiedene Methoden der kognitiven Modellierung, die sich bzgl. ihrer Komplexität stark unterscheiden. Gemeinsam ist diesen Methoden, dass ihnen ein theoretisches Modell menschlicher Informationsverarbeitung zugrunde liegt. Dieses Modell wird verwendet, um die bei der Abarbeitung einer Aufgabe erforderlichen kognitiven und physischen Schritte zu beschreiben und daraus dann eine Zeitdauer für die Bearbeitung einer Aufgabe abzuleiten.

Wir untersuchen in diesem Beitrag mit dem *Keystroke Level Model GOMS* (Card et al. 1983) und *CogTool* (John et al. 2004) zwei etablierte Methoden der kognitiven Modellierung. Das *Keystroke Level Model GOMS* (kurz KLMGOMS) ist die sowohl bzgl. der theoretischen Fundierung als auch bzgl. der konkreten Durchführung einfachste Methode der kognitiven Modellierung. Dagegen ist die dem CogTool zugrundeliegende ACT-R Theorie (Anderson & Lebiere 1998) der im Moment am besten theoretisch ausgearbeitete, aber auch komplexeste kognitive Modellierungsansatz. Wir werden diese beiden Methoden anhand konkreter betriebswirtschaftlicher Szenarien in Bezug auf ihre Fähigkeit vergleichen, genaue Vorhersagen der benötigten Bearbeitungszeiten zu generieren. Weiterhin werden wir Empfehlungen ableiten, in welchen Situationen diese Methoden sinnvoll eingesetzt werden sollten.

2 Kognitive Modellierung mit GOMS

Eine GOMS-Analyse (Card et al. 1983) erlaubt vorherzusagen, wie lange ein erfahrener Benutzer für die Bearbeitung einer vorgegebenen Aufgabe in einer Benutzerschnittstelle benötigt.

Das Akronym GOMS steht für *Goals, Operators, Methods* und *Selection Rules*:

- Ein Ziel (Goal) beschreibt, was der Benutzer erreichen möchte.
- Operatoren sind grundlegende physische (z.B. Drücken einer Taste oder Positionieren des Mauszeigers) oder kognitive Prozesse (z.B. Abruf einer Information aus dem Gedächtnis oder mentale Vorbereitung für den nächsten Schritt in einer Handlungssequenz), die der Benutzer zur Erreichung des Ziels ausführen muss.
- Methoden repräsentieren erlernte Operator-Sequenzen, die ein Benutzer automatisch ausführt, um ein Ziel oder Teil-Ziel zu erreichen.
- Verfügt der Benutzer zu einem Zeitpunkt der Aufgabenbearbeitung über mehrere alternative Methoden zur Erreichung eines Ziels, so entscheiden Selektionsregeln, welche dieser Methoden verwendet wird.

Es wurden mehrere Varianten der GOMS-Analyse veröffentlicht, die sich bzgl. der Standardisierung und Komplexität der Modellierung unterscheiden (siehe z.B. John & Kieras 1996). Wir werden uns im Folgenden auf die einfachste Form der GOMS-Analyse beschränken, dass sogenannte *Keystroke Level Model* (KLMGOMS). Diese Variante verwendet zur Vorhersage der Bearbeitungszeit einer Aufgabe allein die dafür notwendigen Operatoren. Alternative über Selektionsregeln gesteuerte Bearbeitungssequenzen werden hier nicht verwendet. Das führt zu einer erheblichen Vereinfachung des Modellierungsaufwands. Deshalb ist diese Variante für die praktische Anwendung in konkreten Design-Projekten besonders gut geeignet⁶².

Unterschiedliche Personen benötigen natürlich unterschiedliche Zeiten für die grundlegenden physischen oder kognitiven Operationen. Die GOMS Analyse abstrahiert von den Zeiten konkreter Personen durch die Verwendung typischer Durchschnittswerte (z.B. Tastendruck beim Tippen einer Zeichenkette 0,23 Sekunden, Positionieren des Mauszeigers 0,44 Sekunden, Mentale Vorbereitung 1,2 Sekunden, etc.). Diese Durchschnittswerte wurden in experimentellen Studien ermittelt (z.B. John & Kieras 1996; Olson & Olson 1990; Schrepp & Fischer 2007). In KLMGOMS wird zwischen verschiedenen physischen Operatoren unterschieden. Für alle kognitiven Operationen wird im allgemeinen nur ein einziger Operator (typischerweise mit M für *Mental Operation* bezeichnet) verwendet.

Bei der Analyse einer Aufgabenstellung mit KLMGOMS geht man wie folgt vor:

- Man ermittelt die Sequenz aller physischen Operatoren, die zur Erledigung der Aufgabe notwendig sind.
- Man fasst Teilsequenzen physischer Operatoren zu Gruppen zusammen. Eine solche Gruppe repräsentiert dabei eine gelernte Folge von Operatoren, die ohne weiteren kognitiven Aufwand in einer Sequenz ausgeführt wird (z.B. Maus auf ein Ziel positionieren und dieses anklicken).

⁶² Eine Einschränkung des KLMGOMS ist, dass es dieser Ansatz nicht erlaubt, parallele Prozesse menschlicher Informationsverarbeitung abzubilden. Bei Aufgaben, in denen solche parallelen Denkprozesse eine wichtige Rolle spielen, werden die aus der KLMGOMS Analyse resultierenden Zeiten daher die realen Zeiten zur Aufgabenbearbeitung überschätzen. In solchen Anwendungsszenarien kann die Schätzung mit dem sogenannten Critical Path Method GOMS (John & Kieras 1996) durchgeführt werden.

- Man platziert einen kognitiven Operator vor jede Gruppe.

Der Umgang mit kognitiven Operatoren ist der schwierigste Teil einer GOMS Analyse. Um hier die Freiheitsgrade bei der Modellierung einzuschränken, gibt es eine Reihe von Heuristiken dafür, wann ein kognitiver Operator einzufügen ist (siehe z.B. Kieras 2001).

Für eine GOMS Analyse reicht es schon aus, eine hinreichend genaue Vorstellung einer Benutzerschnittstelle zu besitzen. Diese Vorstellung muss nur konkret genug sein, um die in der Benutzerschnittstelle durchzuführenden Operationen abzuleiten. Eine Implementierung oder eine gut ausgearbeitete Skizze der Benutzerschnittstelle sind hier für die Modellierung nicht notwendig. Für die konkrete Durchführung einer GOMS-Analyse kann auf frei verfügbare Programme zurückgegriffen werden, z.B. GOMSED (Wandmacher 1997; 2002).

3 Kognitive Modellierung mit ACT/R (CogTool)

CogTool ist ein an der Carnegie-Mellon-University entwickeltes Programm (John et al. 2004; John & Salvucci 2005) zur Unterstützung der kognitiven Modellierung (das Tool ist frei verfügbar unter www.cs.cmu.edu/~bej/cogtool/). CogTool basiert auf der von Anderson & Lebiere (1998) entwickelten kognitiven Architektur ACT-R, die entwickelt wurde, um kognitive, perzeptuelle und motorische Prozesse zu modellieren, die Nutzer einer Software durchlaufen. Ein wesentlicher Vorteil von CogTool besteht darin, dass die teilweise sehr komplexen in ACT-R modellierten Zusammenhänge in das Tool „eingebaut“ sind und nicht explizit vom Anwender des Tools angegeben oder berücksichtigt werden müssen. Operationen wie „Betrachten eines Bildschirmbereichs“ oder „Vorbereitung einer motorischen (bzw. visuellen) Aktion“ werden automatisch verwendet während der Anwender angibt, welche (offensichtlichen) Handlungen ein Nutzer bei der Verwendung einer Software durchführt.⁶³ Ein CogTool-Anwender hat lediglich die folgenden 4 Schritte zu durchlaufen:

- Erzeugen eines Design-Storyboards.
- Definition einer Menge von Aufgaben.
- Erzeugen eines Skripts.
- Modifikation des Designs und Wiederholung der weiteren Schritte.

Ein *Design-Storyboard* besteht aus einer Menge unterschiedlicher Zustände (die als *Frames* bezeichnet werden) der Benutzerschnittstelle des modellierten Systems (z.B. Eingabeformulare oder Tabellen). Innerhalb jedes Frames werden interaktive Bereiche durch sogenannte *Widgets* identifiziert. Diese Widgets repräsentieren Elemente, mit denen der Benutzer während der Bearbeitung einer Aufgabe interagiert (z.B. Eingabefelder, Links, Buttons oder Menüs). Einzelne Frames werden dann mit Hilfe von Übergängen (*Transitions*) miteinander

⁶³ Die genannten Operatoren sind in KLMGOMS nicht vorhanden. Dort gibt es nur den pauschalen „Think“ Operator. Das bedeutet, dass mit CogTool/ACT-R wegen des erweiterten Repertoires an Operatoren auch potenziell differenzierte Modellierungen erzeugt werden können.

verbunden, um den Interaktionsfluss beim Bearbeiten einer Aufgabe zu beschreiben. Ein Übergang führt stets 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.

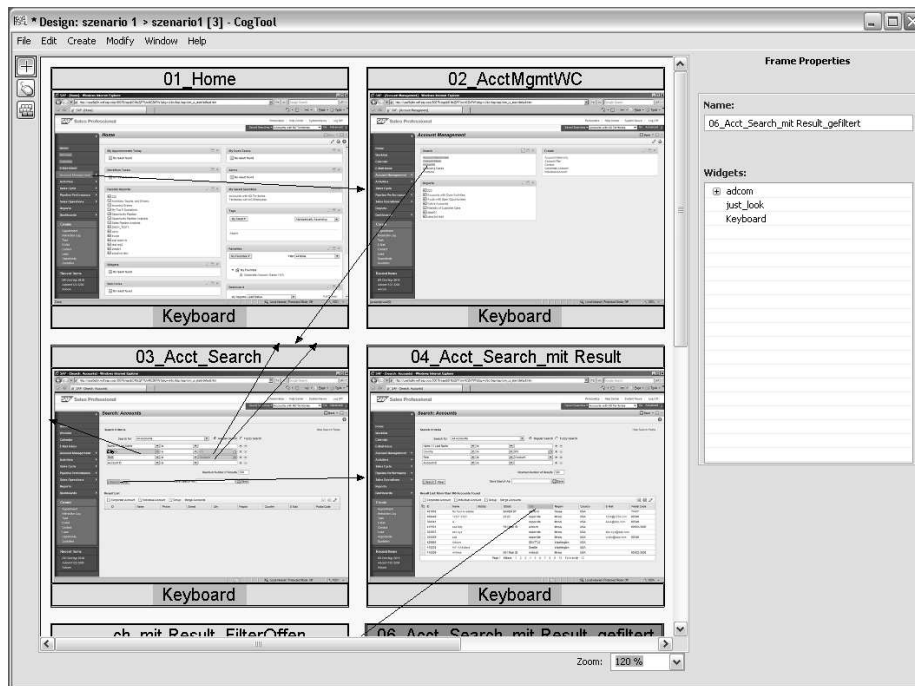


Abbildung 1: Beispiel für ein CogTool Design-Storyboard mit Widgets und Übergängen.

Design-Storyboards werden bereits auf der Basis möglicher typischer Interaktionssequenzen zusammengestellt. Aus den einzelnen Schritten einer Aufgabenbearbeitung (z.B. Öffne ein Menü, Speichere ein Objekt, Navigiere zur Suche) kann nun ein Skript aufgezeichnet werden. Hierzu wird das Design-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 (z.B. visuelle Orientierung, kognitive Operation) weder kennen, noch explizit eingeben – sie werden von CogTool automatisch eingefügt. Dies ist eine der wesentlichen Stärken von CogTool, da hier der Aufwand für die Modellierung massiv verringert wird.

Das Script wird nun verwendet, um die für die Aufgabenbearbeitung notwendigen Zeiten vorherzusagen. Hierfür werden ähnlich wie in GOMS bekannte Zeiten für bestimmte Operationen benutzt, wobei allerdings auch parallele Prozesse der Informationsverarbeitung berücksichtigt werden.

Ein wichtiges Hilfsmittel von CogTool ist das ACT-R Visualisierungsfenster, das die gemäß ACT-R angenommenen elementaren Operationen anzeigt. Dadurch bekommt der Anwender einen guten Überblick darüber, welcher Art diese Operationen primär sind (z.B. motorische

Aktion, visuelle Orientierung, Nachdenken). Abbildung 2 zeigt ein Beispiel für den Inhalt des Visualisierungsfensters. Einzelne horizontale Balken stehen für die zeitliche Ausdehnung einzelner Arten von Operationen.

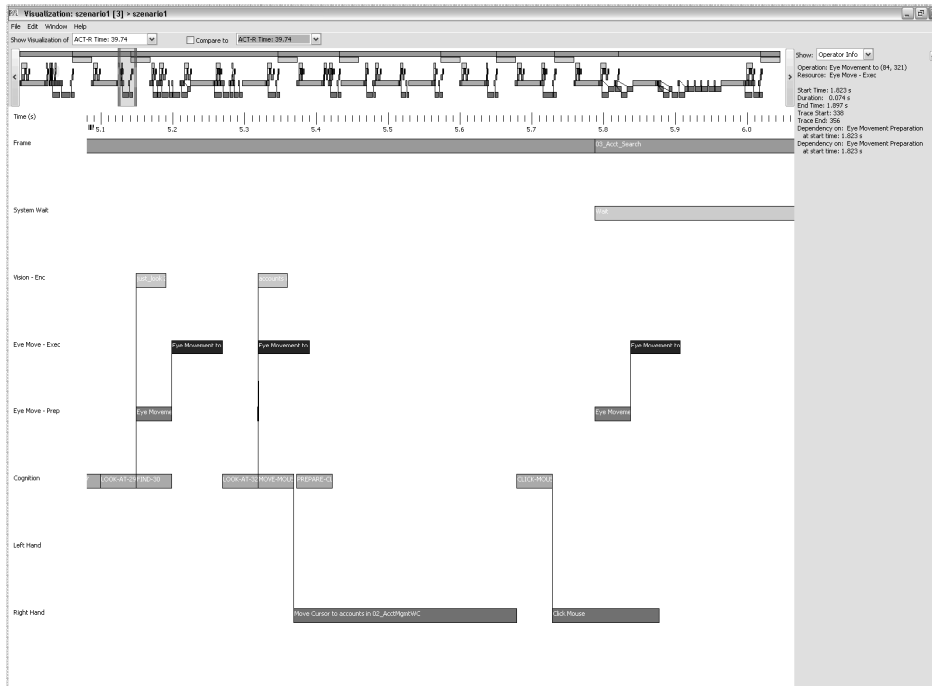


Abbildung 2: ACT-R Visualisierungsfenster

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 Baukastensystem zusammengestellt werden können. Dies verbindet den hohen Auflösungsgrad der CogTool-Modellierung mit einem pragmatischen Vorgehen zur Modellierung komplexer Interaktionsszenarien.

4 Vergleich der Methoden

Eine kognitive Modellierung betriebswirtschaftlicher Anwendungen kann zur Beantwortung folgender Fragestellungen verwendet werden:

- Ist es möglich vorgegebene Anwendungsszenarien innerhalb bestimmter zeitlicher Vorgaben mit der Anwendung zu erledigen?

- Welcher Entwurf der Benutzerschnittstelle erlaubt eine schnellere Abarbeitung der typischen Arbeitsaufgaben?

Für die zweite Fragestellung ist es in der Regel auch sehr interessant das Verhältnis der Zeiten zu betrachten, um abschätzen zu können, wie stark sich Modifikationen an bestehenden Entwürfen auf die Bearbeitungszeiten auswirken, d.h. ob sich solche Modifikationen überhaupt lohnen.

CogTool liegt eine detailliertere Modellierung der kognitiven Architektur zugrunde als GOMS. Zum Beispiel berücksichtigt CogTool bei der Zeitschätzung einer Mausbewegung die Distanz zwischen Start und Endpunkt, während in einer GOMS Analyse hierfür stets ein konstanter Zeitfaktor angesetzt wird⁶⁴. Daher sollten die mit CogTool geschätzten Bearbeitungszeiten näher an der Realität liegen, als die Ergebnisse einer GOMS Analyse. Allerdings ist natürlich bei längeren Szenarien auch zu erwarten, dass sich die Unterschiede evtl. ausmitteln (die in der GOMS-Analyse verwendeten Zeiten für Operatoren sind ja gerade als mittlere Zeiten über ein realistisches Spektrum an verschiedenen Szenarien definiert).

Für den Vergleich der beiden Methoden wurden 3 typische Bearbeitungsszenarien eines Web-basierten CRM Systems mit KLMGOMS und CogTool analysiert:

- Szenario 1: Kundendaten suchen und ändern,
- Szenario 2: Daten zu einem neuen Kunden erfassen,
- Szenario 3: Auftrag mit mehreren Positionen erfassen.

Die Szenarien unterscheiden sich stark in Bezug auf die einzugebenden Daten. Bei Szenario 1 steht die Navigation im Vordergrund (Navigation über 8 Screens bzw. Screen-Zustände), während nur wenige Daten (15 Zeichen in 3 Feldern) eingegeben werden. Szenario 2 beschreibt massive Dateneingabe in ein Formular (Navigation über 3 Screens, Eingabe von 107 Zeichen in 17 Feldern). Szenario 3 beschreibt Dateneingabe in einer Tabelle (2 Screens, Eingabe von 33 Zeichen in 9 Tabellenzellen).

Für die Modellierung mit KLMGOMS wurden ausschließlich die in Wandmacher (2002) beschriebenen Operatoren und Operatorzeiten verwendet. Eine Ausnahme bildet die Modellierung von Szenario 2. Hier musste berücksichtigt werden, dass die in das Formular einzutragenden Daten in der Regel vom Benutzer aus einem anderen Medium abgelesen werden (Ausdruck oder auch evtl. eine Visitenkarte). Für eine realistische Zeitschätzung musste hier also auch das Lesen der Daten und der Wechsel der Orientierung vom Bildschirm zum Ausdruck und zurück berücksichtigt werden. Ein solches Nachlesen wurde als Sequenz aus einem mentalen Operator, einem Aufmerksamkeitswechsel (vom Screen zum Ausdruck), einer Leseoperation und einem weiteren Aufmerksamkeitswechsel (vom Ausdruck zum Screen) modelliert. Die für den Aufmerksamkeitswechsel notwendige Zeit wurde auf 0,36 Sek. gesetzt (siehe Holleis et al. 2007).

⁶⁴ Dies gilt nur für die von uns untersuchte einfachste Form der GOMS Analyse. Die Distanz einer Mausbewegung kann auch in GOMS berücksichtigt werden, z.B. in dem die dafür notwendigen Zeiten über Fitt's Law berechnet werden.

In einer kleinen empirischen Untersuchung wurden diese 3 Szenarien in einem Testsystem von 5 Personen durchgeführt. Die Teilnehmer waren mit der Anwendung sehr gut vertraut. Jeder Teilnehmer spielte jedes Szenario dreimal durch. Für den folgenden Vergleich mit den durch die Modellierung vorhergesagten Zeiten wurden die Mittelwerte der jeweils kürzesten Bearbeitungszeit der 5 Teilnehmer verwendet.

| Szenario | Zeiten | GOMS | Abw. GOMS | CogTool | Abw. CogTool |
|----------|------------------|--------|-----------|---------|--------------|
| 1 | 41,28 (4,39) | 50,24 | 21,7% | 39,74 | 3,73% |
| 2 | 98,16 (18,91) | 102,44 | 0,04% | 96,44 | 1,75% |
| 3 | 35,04 (3,81) | 39,85 | 13,73% | 36,98 | 5,54% |

Tabelle 1: Vergleich der mittleren Bearbeitungszeiten der Teilnehmer mit den Vorhersagen. Werte in Klammern beschreiben die Standardabweichungen der Zeiten.

Abweichungen zwischen den aus der Modellierung resultierenden Zeiten und realen Zeiten von 5-20% werden in der Regel als gute Passung zwischen Modell und Realität akzeptiert (siehe z.B. Teo & John 2006). Die Vorhersagen mit CogTool sind alle sehr nahe an den realen Zeiten. Die Vorhersagen mit KLMGOMS sind für die Szenarien 2 und 3 ebenfalls relativ gut. Für das Szenario 1 gilt dies nicht. D.h. bei den eher eingabeorientierten Szenarien liefert KLMGOMS gute Voraussagen, bei dem eher durch Navigation mit der Maus gekennzeichneten Szenario ist die Abweichung erheblich.

Wodurch kommt dieser Effekt zustande? Wie schon erwähnt, berücksichtigt eine KLMGOMS-Analyse nicht die realen Entfernungen der Mausbewegungen, sondern setzt für jedes Positionieren des Mauszeigers einen konstanten Wert an. Dies ist natürlich vorwiegend bei Szenarien nachteilig, in denen vorwiegend mit der Maus navigiert wird. Bei der Dateneingabe in Formularen hat dieser Fehler in der Schätzung weniger Einfluss.

Eines der Hauptanwendungsszenarien der kognitiven Modellierung ist der Vergleich alternativer Entwürfe einer Benutzerschnittstelle. Für diese Art der Anwendung sind genau genommen nicht die absoluten Zeitschätzungen relevant, sondern lediglich die Relation der Zeiten. D.h. betrachten wir zwei Szenarien S_1 und S_2 . Nehmen wir an, dass $K(S_1)$ und $K(S_2)$ die aufgrund der kognitiven Modellierung geschätzten Bearbeitungszeiten für diese Aufgaben sind und $T(S_1)$ bzw. $T(S_2)$ die real beobachteten Bearbeitungszeiten. Beim Vergleich zweier solcher Szenarien ist es insbesondere interessant, ob die Modellierung eine Aussage erlaubt, um wie viel besser eine der beiden Alternativen ist. Es sollte also gelten $K(S_1) / K(S_2) = T(S_1) / T(S_2)$, d.h. es genügt, wenn die aus der Modellierung gewonnenen Zeitschätzungen das Verhältnis der realen Zeiten widerspiegeln. Die folgende Tabelle 2 zeigt für alle untersuchten Szenarien die Verhältnisse der Zeiten.

| | S1/S2 | S1/S3 | S2/S1 | S2/S3 | S3/S1 | S3/S2 |
|----------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Zeiten | 0,42 | 1,18 | 2,38 | 2,80 | 0,85 | 0,36 |
| GOMS | 0,49 | 1,14 | 2,04 | 2,33 | 0,87 | 0,43 |
| CogTool | 0,41 | 1,07 | 2,43 | 2,61 | 0,93 | 0,38 |

Tabelle 2: Vergleich der Verhältnisse der realen Zeiten und geschätzten Zeiten.

Beide Methoden beschreiben also die Verhältnisse der realen Zeiten recht genau. Wenn es also nur um die Entscheidung zwischen alternativen Entwürfen geht, liefert auch eine Analyse mit KLMGOMS schon gute Ergebnisse.

5 Zusammenfassung

Kognitive Modellierung ist eine kostengünstige Methode, um die Effizienz einer Benutzerschnittstelle quantitativ zu erfassen. Einer der Hauptvorteile der kognitiven Modellierung ist es, dass als Grundlage der Schätzung der notwendigen Bearbeitungszeiten schon eine genauere Vorstellung oder eine grob ausgearbeitete Skizze der Benutzerschnittstelle ausreicht. Damit kann diese Methode schon in der frühen Design-Phase eines Projekts wertvolle Erkenntnisse zur Effizienz eines Entwurfs liefern.

Wir haben in dieser Arbeit zwei etablierte Methoden der kognitiven Modellierung bzgl. ihrer Eignung im Bereich betriebswirtschaftlicher Anwendungen untersucht. Hierfür wurde für eine Menge betriebswirtschaftlicher Standardabläufe jeweils eine Modellierung mit KLMGOMS und CogTool durchgeführt. Ein Vergleich der aus der Modellierung geschätzten Bearbeitungszeiten mit real gemessenen Bearbeitungszeiten von Experten zeigte, dass die über CogTool ermittelten Schätzungen erwartungsgemäß näher an den real beobachteten Zeiten liegen, als die über eine KLMGOMS-Analyse ermittelten Zeitschätzungen. Falls also eine genaue Schätzung der realen Bearbeitungszeiten notwendig ist, sollte CogTool für die Modellierung verwendet werden. Allerdings ist hier anzumerken, dass unsere Daten auf einer sehr kleinen Stichprobe beruhen. Hier sollten weitere Studien mit größeren Personenzahlen und weiteren Anwendungsszenarien folgen.

Andererseits ist eine Modellierung mit KLMGOMS weniger aufwändig und im Design-Prozess schon früh einsetzbar, ohne dass eine detaillierte Skizze der Benutzerschnittstelle erforderlich ist. Man ist in dieser Phase vorwiegend an der Frage interessiert, welche einer Menge von Design-Alternativen effizienter ist und wie groß die zu erwartenden Unterschiede sind. Genaue Zeitschätzungen sind daher hier in der Regel nicht notwendig.

Wie ein Vergleich der relativen Bearbeitungszeiten zeigt, geben auch die aus einer GOMS-Analyse ermittelten Schätzungen schon ein recht gutes Bild der relativen Zeitunterschiede. Falls also in einer sehr frühen Design-Phase noch keine genaueren Skizzen der Benutzer-

schnittstelle vorliegen, können auch schon mit einer KLMGOMS-Analyse sehr fundierte Design-Entscheidungen getroffen werden.

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.
- Holleis, P., Otto, F., Hußmann, H. & Schmidt, A. (2007). Keystroke-Level Model for Advanced Mobile Phone Interaction. In *Proceedings of ACM CHI 2007 Conference on Human Factors in Computing Systems*. New York: ACM Press. S. 1505-1514.
- 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. In Dykstra-Erickson, E. & Tscheligi, M. (Hrsg.), *Proceedings of CHI Conference on Human Factors in Computing Systems*. New York: ACM Press. S. 455 – 462.
- John, B. E. & Salvucci, D. D. (2005) Multi-Purpose Prototypes for Assessing User Interfaces in Pervasive Computing Systems. *IEEE Pervasive Computing* 4(4), S. 27-34.
- Kieras, D. (2001). Using the Keystroke-Level Model to Estimate Execution Times, Online handout, <ftp://www.eecs.umich.edu/people/kieras/GOMS/KLM.pdf>.
- 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* 1(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. (Hrsg.), *Challenges for Assistive Technology*. Amsterdam: IOS Press. S. 646-651.
- Teo, L. & John, B.E. (2006). Comparisons of Keystroke-Level Model Predictions to Observed Data. In *Proceedings of ACM CHI 2007 Conference on Human Factors in Computing Systems*. New York: ACM Press. S. 1424 – 1426.
- Wandmacher, J. (2002). GOMS-Analysen mit GOMSED. Online verfügbar unter: <http://www1.tu-darmstadt.de/fb/fb3/psy/kogpsy/indexgoms.htm>.
- Wandmacher, J. (1997). Ein Werkzeug für GOMS-Analysen zur Simulation und Bewertung von Prototypen beim Entwurf. In Szwillus G. (Hrsg.), *Prototypen für Benutzungsschnittstellen*. Paderborn: Universität Paderborn. Erschienen als Notizen zu Interaktiven Systemen 19, S. 35–42.

Kontaktinformationen: Dr. Martin Schrepp: martin.schrepp@sap.com – Dr. Theo Held: theo.held@sap.com