

Wie effizient ist mein User Interface?

Bearbeitungszeiten einfach und genau mit kognitiver Modellierung schätzen

Martin Schrepp

SAP SE
Dietmar-Hopp-Allee 16
68766 Walldorf
martin.schrepp@sap.com

Theo Held

SAP SE
Dietmar-Hopp-Allee 16
68766 Walldorf
theo.held@sap.com

Abstract

Kognitive Modellierung ist ein nützliches Hilfsmittel zur Abschätzung der Effizienz einer Benutzungsschnittstelle. Darüber hinaus helfen kognitive Modelle dabei, potenzielle Benutzbarkeitsprobleme in einem sehr frühen Stadium des Designprozesses zu identifizieren und zu beseitigen. Kognitive Modellierung kann sowohl zur Evaluation existierender Produkte, aber auch schon mit grob skizzierten ersten Design-Prototypen durchgeführt werden. Der Beitrag vermittelt das notwendige Wissen zur Anwendung kognitiver Modellierung (speziell GOMS und CogTool) in der Design-Praxis. Die Anwendungsmöglichkeiten, aber auch die Grenzen der Methoden, werden durch Beispiele konkreter Projekte untermauert.

Keywords

Kognitive Modellierung, GOMS, CogTool, Effizienz, Bearbeitungszeiten

Einleitung

Bei vielen Anwendungen ist die Effizienz der Benutzungsschnittstelle (d.h. die Zeit, die ein geübter Benutzer für typische Standardaufgaben benötigt) eine extrem wichtige Produkteigenschaft. Dies gilt insbesondere für betriebswirtschaftliche Anwendungen. Da betriebswirtschaftliche Standardprozesse (z.B. Erstellen eines Angebots oder Erfassen eines neuen Auftrags) von einem Benutzer während eines typischen Arbeitstages oft sehr häufig durchgeführt werden müssen, können selbst kleinere Zeitverluste durch eine ineffiziente Benutzungsschnittstelle erhebliche Produktivitätsverluste verursachen.

Daher ist es wichtig, die Effizienz einer Benutzungsschnittstelle abschätzen zu können. Eine solche Schätzung kann dafür verwendet werden, die Eignung der Benutzungsschnittstelle 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. Daneben ist eine solche Zeitschätzung notwendig, um schon in der Design-Phase alternative Entwürfe vergleichen zu können bzw. einen bestehenden Entwurf hinsichtlich seiner Effizienz weiter zu verbessern.

Eine Möglichkeit Bearbeitungszeiten zu schätzen, 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 auch 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 verursacht daher hohe Zeitaufwände und Kosten.

Kognitive Modellierung erlaubt es, den zeitlichen Aufwand vorgegebener Arbeitsabläufe zu schätzen, ohne Testpersonen einzubeziehen. Diese Methode kann auch schon ausgehend von groben Entwürfen in der frühen Design-Phase eines Projekts verwendet werden (Held & Schrepp 2011). Wir stellen mit GOMS und CogTool zwei einfache Möglichkeiten der Kognitiven Modellierung vor, die man mit wenig Übung in eigenen Designprojekten einsetzen kann.

Einsatzgebiete kognitiver Modellierung

Das Spektrum der möglichen Anwendungsbereiche kognitiver Modellierung reicht von der Validierung rudimentärer Entwürfe von Benutzungsschnittstellen bis hin zum objektiven Vergleich der Effizienz implementierter Systeme. Die folgenden Beispiele vermitteln einen Eindruck zu möglichen Einsatzgebieten.

Vergleichende Untersuchung der Effizienz von Softwareprodukten

In einem Einführungsprojekt für eine neue Call-Center Software wünschte der Kunde zu erfahren, wie schnell bestimmte Standardprozesse mit der neuen Software im Vergleich zur bestehenden Software abgearbeitet werden können. Für jeden der untersuchten Prozesse existieren explizite Zeitvorgaben, die unter keinen Umständen überschritten werden dürfen. Um den Vergleich der Zeiten so objektiv und nachvollziehbar wie möglich zu gestalten, wurde entschieden, anstelle der Messung realer Nutzerzeiten eine vergleichende Modellierung mit CogTool durchzuführen. Diese Vorgehensweise hat den Vorteil, dass Faktoren wie unterschiedliche Systemantwortzeiten oder auch variierende Bearbeitungsgeschwindigkeiten von Nutzern die ermittelten Zeiten nicht beeinflussen.

Die Ergebnisse der vergleichenden Untersuchung ergaben, dass die Bearbeitungszeiten des neuen Systems bei allen Standardprozessen deutlich geringer sind und die zeitlichen Verluste ausschließlich auf verzögerte Antwortzeiten zurückzuführen sind, die durch entsprechende Anpassung der Hardware optimiert werden können. Zudem konnte mit Hilfe der CogTool

Modellierungen eindrucksvoll dargestellt werden, welche Nutzbarkeitsprobleme im Altsystem bestanden, die im Neusystem weitgehend beseitigt waren.

Prüfen, ob absolute Zeitvorgaben haltbar sind

Für eine betriebswirtschaftliche Anwendung war eines der Design-Ziele, dass einfache Belege innerhalb einer Minute eingegeben werden können. Dies sollte natürlich vor der Implementierung schon direkt am Design getestet werden. Dazu wurden mehrere Szenarien ausgearbeitet (und die dazu einzugebenden Daten festgelegt) und dann anhand der ersten Mockups durchgespielt. Durch eine Modellierung mit CogTool konnte ermittelt werden, dass das vorgegebene Ziel für die Effizienz mit dem vorgeschlagenen Design erreichbar war.

Effizienz von Keyboard Interaktionen messen

Für viele Nutzer ist eine effiziente Bedienung der Benutzungsschnittstelle über Tastatur sehr wichtig. Das betrifft zum Beispiel behinderte Nutzer, die nicht mit einer Maus arbeiten können (aber auch Experten, die eine Anwendung im Laufe eines Arbeitstages sehr häufig benutzen und nicht ständig zwischen Tastatur und Maus wechseln wollen). Eine zentrale Frage bei der Gestaltung der notwendigen Keyboard Funktionen ist natürlich, wie effizient die reine Keyboard-Bedienung im Vergleich zur Nutzerführung mit Maus und Keyboard ist. Über eine kognitive Modellierung können die beiden Interaktionsmechanismen leicht verglichen werden, um diese Frage zu beantworten (siehe Schrepp & Hardt 2007 für eine entsprechende Anwendung).

GOMS Analyse

Die Grundidee einer GOMS-Analyse (Card et al. 1983) ist es, die Benutzerinteraktion bei der Bearbeitung einer Aufgabe in elementare Operatoren zu zerlegen. Die Bearbeitungszeit der Aufgabe wird dann aus den bekannten Zeiten dieser elementaren Operatoren geschätzt.

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

- Ein Ziel (Goal) beschreibt, was der Benutzer erreichen möchte. Zum Beispiel Geld an einem Geldautomaten abheben.
- Operatoren sind grundlegende physische (z.B. Drücken einer Taste oder eine Touch-Interaktion mit dem Bildschirm des Automaten) 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 gibt eine Reihe von Varianten der GOMS-Analyse. Diese unterscheiden sich massiv bzgl. der Komplexität der Modellierung (siehe z.B. John & Kieras 1996) und im notwendigen Aufwand zur Durchführung der Analyse.

Für die praktische Anwendung ist vor allem das sogenannte Keystroke Level Model (KLMGOMS) geeignet. Hier wird die Bearbeitungszeit einer Aufgabe allein aus den dafür notwendigen elementaren Operatoren berechnet. Alternative über Selektionsregeln gesteuerte Bearbeitungssequenzen werden hier ebenso wie parallele Prozesse der Informationsverarbeitung ignoriert (falls man solche Prozesse berücksichtigen will, kann man z.B. CPMGOMS verwenden, siehe z.B. John & Kieras 1996). Diese Vereinfachungen führen natürlich zu weniger genauen Zeitschätzungen, als dies mit anderen GOMS Varianten möglich ist. Vergleiche der geschätzten Zeiten mit realen Bearbeitungszeiten zeigen aber, dass die Schätzungen für den praktischen Bedarf immer noch ausreichend genau sind (z.B. Teo & John 2006; Schrepp & Held 2010).

Die KLMGOMS Analyse basiert auf der Zerlegung einer Aufgabenbearbeitung in elementare Operatoren. 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 nur ein einziger Operator 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).
- Man platziert einen kognitiven Operator vor jede Gruppe.

Der Umgang mit den kognitiven Operatoren ist der schwierigste Teil einer GOMS Analyse. Hier muss man unter Umständen Annahmen über die bei der Lösung der Aufgabe ablaufenden Denkprozesse machen. Daher gibt es hier in der praktischen Anwendung in der Regel einige Freiheitsgrade bei der Modellierung. Um diese einzuschränken, listet Kieras (2001) eine Reihe von Heuristiken für das Einfügen kognitiver Operatoren auf. Danach sollte z.B. ein kognitiver Operator eingefügt werden, wenn:

- ein neuer Teilschritt eingeleitet wird,
- eine Information aus dem Gedächtnis abgerufen werden muss,

- der Nutzer eine Information auf dem Bildschirm suchen muss,
- der Nutzer verifiziert, ob seine Eingaben korrekt sind.

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 detaillierte Skizze der Benutzerschnittstelle sind hier nicht notwendig.

Für die konkrete Durchführung einer GOMS-Analyse kann auf frei verfügbare Programme zurückgegriffen werden, z.B. GOMSED (Wandmacher 1997; 2002). Es ist aber auch leicht möglich eine solche Analyse schon mit Papier und Bleistift durchzuführen oder sich selbst ein kleines Tool zu basteln, z.B. eine Excel-Tabelle, in die man nur die Operatoren eingibt und die die Zeiten automatisch ergänzt und aufsummiert.

Modellierung mit CogTool

CogTool ist eine Software, die die kognitive Modellierung von computerbasierten Aufgaben ermöglicht (John et al. 2004, John & Salvucci 2005). Die Software wurde an der Carnegie Mellon Universität entwickelt und ist verfügbar unter cogtool.com. CogTool basiert auf der kognitiven Architektur ACT-R (Anderson & Lebiere 1998; für weitere theoretische Grundlagen siehe auch Card et al. 1983). ACT-R stellt den theoretischen Rahmen zur Beschreibung kognitiver, perzeptueller und motorischer Prozesse dar, die für menschliches Problemlöseverhalten relevant sind.

Die Entwicklung eines kognitiven Modells mit CogTool beinhaltet die folgenden Schritte: (1) Erzeugen eines „Design-Storyboards“, (2) Definition von Aufgaben, (3) Erzeugen eines Aufgaben-Skripts (d.h. eines konkreten Handlungsablaufs) und (4) die iterative Modifikation von Designs und Wiederholung der Schritte wie oben beschrieben.

Ein „Design-Storyboard“ zeigt die relevanten Zustände, die die Benutzungsschnittstelle innerhalb einer Nutzungssequenz annehmen kann (die sog. „Frames“). Die für eine Nutzungssequenz relevanten Elemente der einzelnen Frames (z.B. Eingabefelder, Menüs, Drucktasten) werden in CogTool mit „Widgets“ markiert. Die einzelnen Frames werden durch Übergänge („Transitions“) verbunden, die den Ablauf der Interaktionen innerhalb einer Nutzungssequenz definieren. Jeder Übergang verbindet ein Widget eines Frames mit einem „Zielframe“ (d.h. es wird definiert, wie sich die Benutzungsschnittstelle durch Betätigung einer Funktion ändert). Abbildung 1 zeigt ein Beispiel für ein Design Storyboard mit Widgets und Transitions.

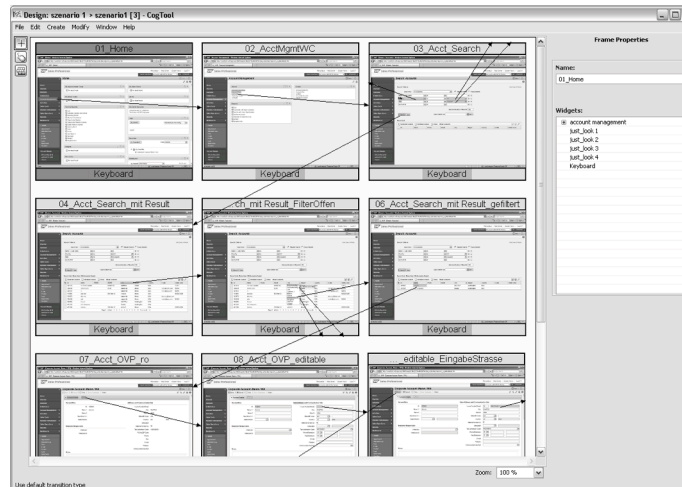


Abbildung 1: Beispiel für ein CogTool Design Storyboard mit Widgets und Transitions

Aus einem so festgelegten Ablauf wird ein „Script“ abgeleitet, mit dem ein CogTool Design in ein auf ACT-R basierendes kognitives Modell übersetzt wird. Dabei muss der Nutzer von CogTool die zugrunde liegenden kognitiven, motorischen oder visuellen Prozesse weder kennen noch explizit einfügen – diese Aufgabe wird von CogTool automatisch erledigt. Im Gegensatz zu Modellierungen mit KLMGOMS werden in CogTool parallele Prozesse der Informationsverarbeitung berücksichtigt. Um solche Prozesse zu visualisieren, bietet CogTool ein „ACT-R Visualisierungsfenster“. Hier werden kognitive, motorische und visuelle Prozesse im zeitlichen Ablauf dargestellt. Abbildung 2 zeigt ein Beispiel.

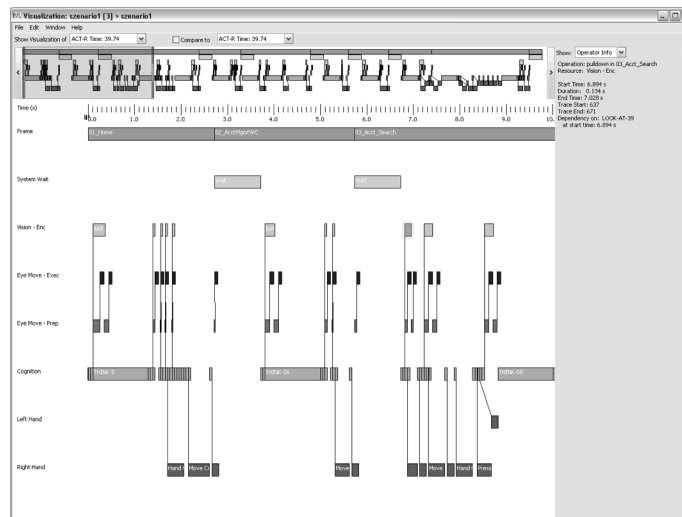


Abbildung 2: ACT-R Visualisierungsfenster

Wie genau sind die geschätzten Zeiten?

Eine zentrale Frage ist natürlich, wie genau die mit GOMS oder CogTool geschätzten Zeiten die wirklichen Bearbeitungszeiten abbilden. In einer Studie (Schrepp & Held, 2010) wurden drei längere Szenarien in einem betriebswirtschaftlichen System mit GOMS und CogTool modelliert und mit gemessenen Zeiten von realen Personen verglichen (die diese Szenarien in einem Experiment mehrmals durchspielten). Bei Szenario 1 stand die Navigation im Vordergrund, Szenario 2 beschrieb die massive Dateneingabe mit der Tastatur in ein Formular und Szenario 3 die Dateneingabe in eine Tabelle.

Die Ergebnisse zeigten, dass die mit CogTool modellierten Zeiten sehr nahe an den Mittelwerten der Teilnehmer lagen. Die Abweichungen zwischen geschätzter Bearbeitungszeit und dem Mittelwert der gemessenen Zeiten aller Teilnehmer für die drei Szenarien betragen nur 3,7% (Szenario 1), 1,75% (Szenario 2) und 5,54% (Szenario 3). Die Abweichungen für GOMS waren mit 21,7% (Szenario 1), 0,04% (Szenario 2) und 13,73% (Szenario 3) im Schnitt deutlich größer, bewegten sich aber immer noch in einem für praktische Anwendungen brauchbaren Rahmen.

Diese Ergebnisse zeigen auch, dass CogTool vor allem bei Szenarien, bei denen vor allem navigiert wird, einen deutlichen Vorteil besitzt. GOMS setzt für alle Mausbewegungen einen konstanten Zeitwert an, während CogTool die real zurückgelegten Mauswege berücksichtigt. Für Szenarien, bei denen die Dateneingabe mit der Tastatur dominiert, spielt dies keine Rolle, hier sind die mit GOMS erzielten Ergebnisse gleichwertig.

Diskussion

GOMS und CogTool wurden entwickelt, um die Interaktionen eines Nutzers mit klassischen Eingabemechanismen (Tastatur und Maus) zu beschreiben. Da heute Touch-Interfaces zunehmend dominieren, stellt sich natürlich die Frage, wie zeitgemäß diese Methoden sind. Es gibt allerdings mittlerweile Erweiterungen von GOMS, die es erlauben auch die Interaktion mit Touch-Interfaces zu modellieren (siehe z.B. Holleis et al. 2007). Hier werden einige klassische GOMS Operatoren durch äquivalente Touch-Operatoren und deren Zeiten ersetzt (z.B. das Bewegen der Maus durch eine Positionierung des Fingers oder das Drücken einer Taste auf dem Keyboard durch ein Berühren der Taste auf der virtuellen Tastatur). Einige Operatoren, z.B. der mentale Operator, werden unverändert weiter verwendet. Es werden aber auch neue, für Mobile-Interaktionen typische Operationen, eingeführt, z.B. der Wechsel der Aufmerksamkeit vom Display eines Smartphones auf ein Objekt der realen Welt und zurück.

Eine weitere wesentliche Einschränkung kognitiver Modellierung ist, dass typische Nutzerfehler nur sehr eingeschränkt modelliert werden können. Falls eine Nutzungsschnittstelle auf Kosten etwas komplexerer Interaktionen die Wahrscheinlichkeit von Eingabefehlern minimiert und damit die Bearbeitung beschleunigt, ist das mit kognitiver Modellierung nur schwer zu erfassen.

Literatur

- Anderson, J.R. & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah: Lawrence Erlbaum Associates.
- Card, S., Moran T.P. & Newel A. (1983). *The Psychology of Human Computer Interaction*. Mahwah: Lawrence Erlbaum Associates.
- Held, T. & Schrepp, M. (2011). Effective cognitive modeling for business software – Where and how can cognitive models support usability professionals in their everyday work? Usability Professionals Association International Conference 2011. Online: <http://upa.omnibooksonline.com/papers/012.pdf>.
- 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.E., Prevas, K., Salvucci, D. & Koedinger, K. (2004). Predictive Human Performance Modeling Made Easy. *Proceedings of CHI, 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), 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.
- Schrepp, M. & Held, T. (2010). Anwendung von GOMS-Analysen und CogTool in der Design-Praxis. In: J. Ziegler & A. Schmidt (Eds.): *Mensch & Computer 2010*. Oldenbourg Verlag, S. 351-360.
- 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. (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.
- Wandmacher, J. (2002). GOMS-Analysen mit GOMSED. Online verfügbar unter: <http://www1.tudarmstadt.de/fb/fb3/psy/kogpsy/indexgoms.htm>.