

# Rapid Cognitive Modeling, Prototyping und Evaluation mit CogTool

**Tobias Ehni**

Freiberuflicher Usability Consultant  
Jonny-Schehr-Straße 10  
99085 Erfurt  
tobias.ehni@googlemail.com

## Abstract

Das hier skizzierte Tutorial stellt die Open-Source Software CogTool vor. Es ist ein Werkzeug für Usability Engineers, Konzepter oder Designer, um Entwürfe in einer frühen Entwicklungsphase zu überprüfen, gegeneinander zu testen (benchmarking) oder zu evaluieren. Auf der Grundlage von Mock-Ups berechnet das Programm die Zeit, die ein routinierter Benutzer braucht, um eine Aufgabe am System auszuführen. Grundlage der Berechnung ist das formal-analytische Verfahren KLM-GOMS. Mit CogTool muss dieses Verfahren nicht mehr von Hand durchgeführt werden und kann schneller erlernt werden. Die Software eignet sich für eine Vielzahl von Anwendungsbereichen wie Websites, mobile Endgeräte und Software.

Im Tutorial soll nach einer Einführung der konkrete Einsatz von CogTool erprobt und anschließend reflektiert werden.

## Keywords:

/// cognitive modeling  
/// CogTool  
/// GOMS  
/// formal-analytische Methode

## 1. Einleitung

Neben klassischen Usability-Methoden, wie z. B. Usability Testing, gibt es weitere Methoden, um Interfaces oder Prototypen zu bewerten oder zu vergleichen. Unter der Bezeichnung cognitive modeling bzw. GOMS existiert eine Reihe von Verfahren, mit denen das Verhalten von Nutzern prognostiziert werden kann (Card et al. 1980, 1983). Gemeinsam ist ihnen die aufgabenanalytische Herangehensweise ohne direkten Kontakt zu Nutzern (Sarodnick & Brau 2011). Sie sind in der Literatur dokumentiert (Preece 1994, Dix 2006, Cox & Peebles 2008, Raskin 2009), scheinen sich in der Praxis aber aufgrund tatsächlicher oder wahrgenommener Hürden noch nicht auf breiter Front durchgesetzt zu haben (Kaindl 2001, John et al. 2004).

Die in diesem Tutorial vorgestellte Software CogTool<sup>1</sup> ist ein Versuch, den Lernaufwand im Umgang mit GOMS bzw. kognitiven Modellen zu verringern und somit ihren Einsatz in der Praxis zu erleichtern. CogTool wurde als Open-Source-Software für Mac und PC an der Carnegie Mellon University unter der Leitung von Prof.

Bonnie E. John erstellt und wird laufend weiterentwickelt.<sup>2</sup>

Bei der Entwicklung wurde darauf geachtet, an der Arbeitsweise von Konzeptern und Interface-Designern mit Prototypen, Wireframes oder Mock-ups anzuknüpfen bzw. diese direkt zu unterstützen.

Die Bedienung von CogTool basiert auf bekannten Interface-Konzepten (WYSIWIG, drag & drop) und kann schnell erlernt werden. Nach der Erstellung eines Prototypen oder dem Import von Screenshots wird der Interaktionsablauf in CogTool modelliert und demonstriert. Danach berechnet das Programm die Zeit zur Aufgabenerledigung automatisch. Diese kann als Maß für Effizienz des Interfaces herangezogen werden (vgl. DIN EN ISO 9241-11). So sind Aussagen über Designalternativen bereits in frühen Entwicklungsphasen möglich, auch wenn kein Zugang zu geeigneten Testpersonen besteht. Auch die Bewertung von Interfaces oder das Benchmarking werden auf diese Weise unterstützt.

Zielgruppe des Tutorials sind Konzepter, Interaktionsdesigner, Usability Engineers, User Experience Consultants (intern und extern). Vertreter aus angrenzenden

Berufsgruppen sind ebenso willkommen wie Universitätsangehörige mit Lehr- oder Forschungstätigkeit im Gebiet Human-Computer Interaction oder verwandter Fachrichtungen.

Die Teilnehmer erhalten an einem Demoprojekt Einblick in die Arbeit mit CogTool und werden im Rahmen des Tutorials in die Lage versetzt, selbst Verhaltensprognosen zu erstellen.

## 2. Anwendung von CogTool

Bei der Bearbeitung von Aufgaben mit interaktiven Systemen vollziehen Benutzer Arbeitsschritte, um ein bestimmtes Ziel zu erreichen (vgl. DIN EN ISO 9241-11). CogTool ermöglicht eine Prognose der Bearbeitungszeit eines routinierten Nutzers, um eine Aufgabe am System zu erledigen (skilled task execution time als Maß für Effizienz). Um zu dieser Prognose zu gelangen, werden bei der Erstellung eines kognitiven Modells mit CogTool drei Schritte durchlaufen:

1. Anlegen eines Projekts und Erstellung eines Prototyps entweder direkt in CogTool oder durch den Import von

- HTML-Dateien, Screenshots, Scans oder Bildern (.png, .gif, .jpg)
- Erstellen des Interaktionsablaufs und Demonstrieren der Aufgaben. Die am System zu erledigenden Aufgaben werden als bekannt vorausgesetzt.
  - Automatische Durchführung der Analyse und Berechnung der Verhaltensprognose.

### 2.1. Interaktionsabläufe erstellen und demonstrieren

In CogTool werden die Zustände eines zu bewertenden Interface oder Prototypen durch frames dargestellt. Ein frame enthält z. B. einen Screenshot einer bestehenden Anwendung und eines oder mehrere widgets. Widgets sind interaktive Elemente wie Buttons, Menüs oder Eingabefelder, an denen Benutzeraktionen definiert werden, z. B. anklicken oder Texteingabe. Zustandsübergänge werden durch transitions modelliert. Eine Transition erfolgt beispielsweise, wenn durch Interaktion mit einem Link eine neue Seite aufgerufen wird. Die Gesamtheit eines Interaktionsverlaufs wird in einem design festgehalten, das mehrere Frames enthält, die durch Transitions verbunden sind (s. Abbildung 1). Ist eine Aufgabe in einem Design durch Frames, Widgets und Transitions beschrieben, muss sie noch demonstriert werden. Dazu werden die spezifizierten Interaktionsschritte einmal ausgeführt. Parallel dazu erstellt CogTool automatisch ein script, das die Interaktionsschritte auflistet. Nach diesem Arbeitsschritt wird die Zeit der Ausführung der Aufgabe automatisch berechnet.

Diese Berechnung erfolgt auf Basis von KLM-GOMS (Card et al. 1980, 1983) und wird intern in ACT-R durchgeführt, einer in der Programmiersprache Lisp implementierten kognitiven Theorie menschlicher Wahrnehmung, Informationsverarbeitung und Bewegung (vgl. Anderson and Lebiere 1998). [Abb. 1]

### 2.2. Ergebnisse berechnen und analysieren

Nach Erstellung des Skripts und Durchführung der Berechnung werden deren

Ergebnisse im Projektfenster angezeigt. Hier ist ein direkter Vergleich von Bearbeitungszeiten verschiedener Entwürfe möglich. Bei Bedarf lassen sich Aufgaben in Gruppen zusammenfassen. Die Berechnung von Summe, Mittelwert, Minimum und Maximum der Aufgabenbearbeitungszeiten innerhalb einer Gruppe erfolgt ebenfalls automatisch. Eine Visualisierung der Interaktion auf Basis des kognitiven Modells ist ebenfalls verfügbar, hier können Aufgaben detailliert analysiert und miteinander verglichen werden. [Abb. 2]

### 2.3. Nutzen, Stärken und Schwächen

CogTool vereinfacht die Erstellung kognitiver Modelle auf Basis von KLM-GOMS und macht dieses Analysewerkzeug Praktikern zugänglich (John 2010). Es besteht eine Vielzahl bereits erprobter und möglicher Anwendungsgebiete (s. Abschnitt 2.4).

Zu den Stärken des Tool zählt die Einsatzmöglichkeit in frühen Projektphasen, besonders wenn kein oder nur eingeschränkter Zugang zu echten Nutzern besteht oder viele Designalternativen verglichen werden sollen. In der Praxis scheinen sich die quantitative Metrik der Aufgabenbearbeitungszeit und die Visualisierung als positive Eigenschaften zu zeigen (LeVan & LeVan 2008). Hervorzuheben ist der bereits implementierte HTML-Export, der für anschließendes Testing mit echten Nutzern dienen kann.

Zu den Schwächen zählen die begrenzte Aussagekraft (Prognosen gelten nur für routinierte Nutzer) und die Tatsache, dass Nutzungsprobleme oder Ermüdung nicht berücksichtigt werden. Auch müssen die durch CogTool zu bewertenden Aufgaben bekannt sein, so dass als Vorarbeit eine Aufgaben- oder Nutzungskontextanalyse notwendig ist. Rückschlüsse auf die Passung zwischen System und Einsatzkontext sind

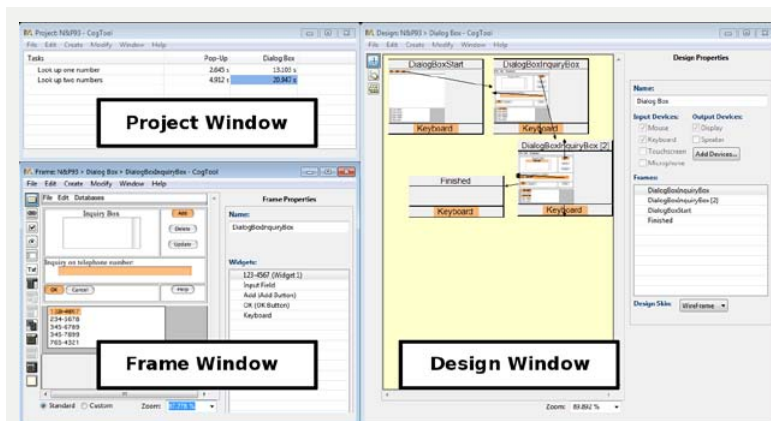


Abb. 1. Modellierung in CogTool

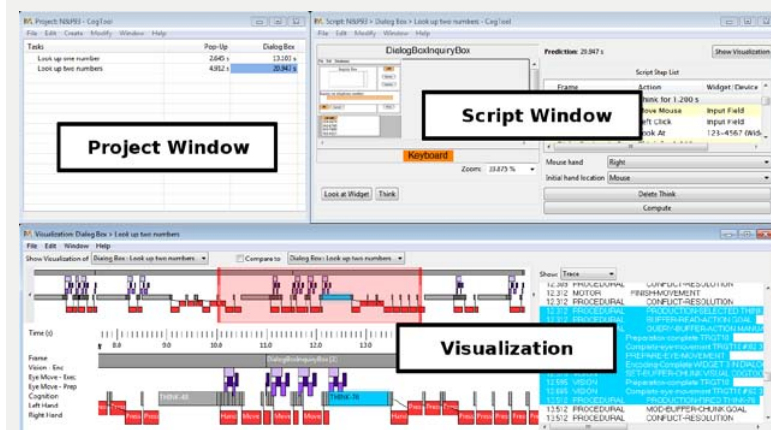


Abb. 2. Skript, Ergebnisse und Visualisierung in CogTool



allein aus der Software heraus auch wegen des zu Grunde liegenden Analyseverfahrens (KLM-GOMS) nicht möglich (vgl. Sarodnick & Brau 2011, S. 131). CogTool berechnet nur die Zeit, die ein routinierter Nutzer entlang einer festgelegten Interaktionsfolge benötigt. Diese Zeit ist ohne Vergleichswerte wenig aussagekräftig, d. h. es besteht die Notwendigkeit, Iterationen oder Benchmarks vorzunehmen. Den Einbezug echter Nutzer kann CogTool nicht ersetzen, wohl aber gibt es sinnvolle Einsatzmöglichkeiten.

## 2.4. Einsatzmöglichkeiten von CogTool

CogTool wurde bereits für verschiedene Produkte und Anwendungskontexte eingesetzt, beispielsweise Websites (John 2009), Datenbanken (John 2004), Handheld devices (John & Salvucci 2005), Expertensysteme wie z. B. Flugzeuginstrumente (John et al. 2009), Optimierung des Verhaltens von Browsertabs (Knight et al. 2007) oder interaktive Geräte in Polizeifahrzeugen (Callander & Zorman 2007). Im UCD-Prozess unterstützt CogTool das iterative Vorgehen in den Phasen Konzeption und Evaluation (vgl. DIN EN ISO 9241-210). Im Rahmen einer Marktanalyse lässt sich durch Benchmarking ein erster Eindruck der Usability von Angeboten der Mitbewerber gewinnen. Ebenfalls vorstellbar ist der Einsatz von CogTool als Werkzeug zur Kommunikation und Entscheidungsfindung im Entwicklungsteam. Auch die Definition von usability-bezogenen Erfolgskriterien oder Kennzahlen lässt sich durch CogTool unterstützen.

CogTool wird derzeit weiterentwickelt, um das Verhalten ungeübter Nutzer vorherzusagen (Teo & John 2008) und um den direkten Import aus verschiedenen Prototyping Tools wie balsamiq<sup>3</sup>, iRise<sup>4</sup> oder dijit<sup>5</sup> zu ermöglichen (Harris et al. 2010).

## 3. Diskussion

Nach der Vorstellung und Anwendung von CogTool sollen Einsatzmöglichkeiten, Praxisrelevanz, Stärken, Schwächen und Entwicklungsmöglichkeiten reflektiert werden.

## Literatur

1. Callander, M. & Zorman, L. (2007). Usability on patrol. In CHI '07 Extended Abstracts on Human Factors in Computing Systems (San Jose, CA, USA, April 28 - May 03, 2007), New York: ACM, 1709-1714
2. Cox, A. L. & Peebles, D. (2008). Cognitive Modelling in HCI Research. In: Cairns, P. A. and Cox, A. L., (Hrsg.): Research Methods for Human-Computer Interaction, Cambridge: Cambridge Univ. Press, 70-87
3. DIN EN ISO 9241-11 (1998). Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten. Teil 11: Anforderungen an die Gebrauchstauglichkeit – Leitsätze, Berlin: Beuth Verlag
4. Dix, A. J. (2006). Human-computer interaction. 3. Aufl., Harlow: Pearson Prentice-Hall
5. Harris, B. N., John, B. E., & Brezin, J. (2010). Human performance modeling for all: importing UI prototypes into cogtool. In Proceedings of the 28th of the international Conference Extended Abstracts on Human Factors in Computing Systems (Atlanta, Georgia, USA, April 10 - 15, 2010). CHI EA '10. New York: ACM, 3481-3486
6. John, B. E. (1994). Toward a deeper comparison of methods: A reaction to Nielsen & Phillips and new data. In Proceedings Companion of CHI, 1994 (Boston, MA, April 24-28, 1994), New York: ACM, 285-286
7. John, B. E. (1995). Why GOMS? interactions, 2, 4, 80-89
8. John, B. E. (2009). Collaborative Shopping. <http://cogtool.hcii.cs.cmu.edu/use-today/examples/collaborative-shopping>
9. John, B. E. (2010). Reducing the Variability between Novice Modelers: Results of a Tool for Human Performance Modeling Produced through Human-Centered Design. Proceedings of the 19th Annual Conference on Behavior Representation in Modeling and Simulation (BRIMS). Charleston, SC, March 22-25, 2010
10. John, B., Blackmon, M. H., Polson, P. G., Fennell, K. & Leonghwee, T. (2009). Rapid Theory Prototyping: An Example of an Aviation Task, Human Factors and Ergonomics Society Annual Meeting Proceedings, 53, 794-798
11. John, B. E., Prevas, K., Salvucci, D. D. & Koedinger, K. (2004). Predictive Human Performance Modeling Made Easy. In Proceedings of the SIGCHI conference on Human factors in computing systems (Vienna, Austria, April 24-29, 2004). New York: ACM, 455-462
12. 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
13. Kaindl, H. (2001): Methods and modeling: fiction or useful reality? CHI '01 extended abstracts on Human factors in computing systems, CHI EA '01, New York: ACM
14. Knight, A., Pyrzak, G., & Green, C. (2007). When two methods are better than one: Combining user study with cognitive modeling. In M.B. Rosson & D.J. Gilmore (Hrsg.): Extended Abstracts of the 2007 Conference on Human Factors in Computing Systems CHI 2007, San Jose: ACM, 1783-1788
15. LeVan, S. & LeVan, C. (2008): Predict Expert Task Time With CogTool, The Perfect Tuna, <http://www.perfecttuna.com/2008/07/predict-expert-task-time-with-cogtool.html>.
16. Raskin, Jef (2009): The human interface. New directions for designing interactive systems. 11. Aufl., Boston: Addison-Wesley.
17. Sarodnick, F. & Brau, H. (2011). Methoden der Usability Evaluation. Wissenschaftliche Grundlagen und praktische Anwendung, 2. Aufl., Bern: Huber
18. Snyder, C. (2003): Paper prototyping. The fast and easy way to design and refine user interfaces. Amsterdam: Kaufmann
19. Kurzvita
20. Tobias Ehni ist Diplom-Medienwissenschaftler und arbeitet nach einem Volontariat bei der SirValUse GmbH als freiberuflicher Usability Consultant. Derzeit ist er wissenschaftlicher Assistent an der TU Ilmenau und Consultant für die Open Source Software GIMP. Er ist Mitglied der GC UPA und interessiert sich für Usability Evaluation und User Research.

<sup>1</sup> <http://cogtool.hcii.cs.cmu.edu/>

<sup>2</sup> Version 1.1.5 datiert vom März 2011

<sup>3</sup> <http://balsamiq.com/>

<sup>4</sup> <http://www.irise.com/>

<sup>5</sup> <http://dojotoolkit.org/reference-guide/dijit/index.html>