

# Ein Software-Tool zur flexiblen Ermittlung von Nutzungszeiten und Nutzungssequenzen

Theo Held, Dietrich Mayer-Ullmann

SAP AG Walldorf

## **Zusammenfassung**

Das „User Performance Measurement Tool“ (UPM) ist ein flexibles und umfangreich konfigurierbares Programm zur Protokollierung von Systemereignissen und den damit zusammenhängenden zeitlichen Aufwänden für spezifische elementare Nutzeraktionen. Ein wesentliches Anwendungsgebiet von UPM ist das Erzeugen von Input-Daten für verschiedene Arten kognitiver Modellierungsansätze. Außerdem eignet es sich als Hilfsmittel zur Erzeugung und Analyse von datenbasierten Nutzungssequenzen, die theoretischen Annahmen zur Nutzung von Software gegenüber gestellt werden können.

## 1 Einleitung

Für die Vorhersage und generelle Modellierung von Nutzerverhalten wurden in den vergangenen Jahrzehnten zahlreiche Ansätze zur kognitiven Modellierung entwickelt. Ausgehend von elaborierten und sehr hoch auflösenden Modellen der kognitiven Architektur wie z.B. ACT-R (Anderson & Lebiere 1998) wurden zahlreiche „pragmatischere“ und einfach zu handhabende Ansätze wie die verschiedenen Ausprägungen des GOMS Modells vorgestellt (Gray, John & Atwood 1993; John & Kieras 1996). Eine wichtige Gemeinsamkeit der Ansätze zur kognitiven Modellierung ist, dass jeder der im Modell auftretenden elementaren Operationen eine Schätzung der für die Durchführung der Operation erforderlichen Zeit zugeordnet werden muss. Beispiele sind die Zeiten, die benötigt werden, um vom Bedienungsmedium Tastatur auf das Medium Maus überzugehen oder um die Maus von einem Bedienelement zu einem anderen zu bewegen.

Für eine Reihe von Standardoperationen sind Schätzungen der Zeiten vorhanden, die bei der Anwendung eines Modelltyps verwendet werden können. Dennoch ist es unvermeidlich, für spezielle Operationen die Zeiten empirisch zu ermitteln, bzw. zu überprüfen, ob bereits verfügbare Zeitschätzungen für bestimmte Operationen einer konkreten zu modellierenden Nut-

zungssequenz zutreffend sind. Je nach Modellierungsansatz und dessen Granularität bilden beobachtbare Ereignisse oder deren Kombinationen die Grundlage für die Definition von elementaren Operationen und deren Dauer. Um den Anforderungen der Modellierungsansätze entsprechen zu können, ist es erforderlich, möglichst viele Ereignisse während einer Interaktionssequenz identifizieren und möglichst präzise aufzeichnen zu können. Weiterhin ist es wünschenswert, dass vordefinierte Kombinationen dieser Ereignisse möglichst automatisch zu Modellparametern zusammengefasst werden.

In dieser Arbeit stellen wir ein flexibles Software-Tool vor, das diesen Anforderungen entspricht. Dieses „User Performance Measurement Tool“ (UPM) bietet sowohl Unterstützung bei der Ermittlung der oben genannten Modellparameter sowie bei der Identifikation typischer Interaktionssequenzen. Das Tool sowie seine unterschiedlichen Anwendungsmöglichkeiten werden im Folgenden dargestellt.

## 2 Das User Performance Measurement Tool (UPM)

Die wesentliche Funktion des Tools besteht darin, eine möglichst hohe Bandbreite an Systemereignissen, die bei der Interaktion eines Nutzers mit einer Applikation auftreten zu protokollieren. Die Anwendung ist derzeit auf Web-Applikationen, die im Internet Explorer 6 laufen, beschränkt. Alle registrierten Ereignisse werden in einer XML-Datei aufgezeichnet, aus der dann über dedizierte Filter-Funktionen die oben erwähnten Modellparameter zur kognitiven Modellierung oder unterschiedlich detaillierte Interaktionssequenzen extrahiert werden können. Ein in der Skriptsprache Python geschriebener Filter ermittelt zum Beispiel Werte, die als direkte Eingabe für den kognitiven Modellierungsansatz des an der Carnegie Mellon University entwickelten CogTool (John & Salvucci 2005) dienen.

Neben seiner Flexibilität und internen Konfigurierbarkeit ist die *Kalibrierungsfunktion* ein weiterer wichtiger Aspekt von UPM. Diese Funktion ermöglicht die exakte Differenzierung von Systemereignissen durch das interaktive Erzeugen von Benennungen für die Ereignisse. Zum Beispiel ist es möglich, das Ereignis des Klicks auf einen Link noch mit dem Namen des Links zu versehen, um das Ereignis in allen folgenden Durchgängen eindeutig identifizieren zu können. Die wichtigste differenzierende Eigenschaft des Tools im Vergleich zu verwandten Produkten wie TechSmith Morae (siehe <http://www.techsmith.com/morae.asp>) oder reinen Event-Logging-Programmen, ist die Möglichkeit, zu konfigurieren, welche Events in welcher Detailstufe protokolliert und zusammengefasst werden. Diese Konfiguration kann im Programmcode sowie in nach geschalteten Ausgabefiltern erfolgen. Tabelle 1 zeigt Beispiele von Systemereignissen, die eine aktuell verwendete Version von UPM protokolliert.

Event	Description
Idle Time	duration of time which was not used for operating with mouse and/or keyboard
Loading Time	(duration of the browser loading time; indicated by the browser status bar animation)
Drag Actions	(number of left mouse clicks and mouse movings in parallel)
Pressed Key Strokes	keep pressed key strokes
Input Device Changes	changing the input devices mouse and keyboard or using both in parallel
Keyboard Usage	sequence using the keyboard only
Mouse Usage	sequence of using the mouse only
Keyboard and Mouse Usage	sequence of using the keyboard and a mouse down click in parallel
Modifier and Mouse Usage	modifier key stroke and mouse down clicks in parallel
Modifier and Arrow Keys	modifier and Arrow key stroke event in parallel
Mouse Focus	setting focus action manually with mouse down click event
Keyboard Focus	setting focus action manually with key down click event

*Tabelle 1: Beispiele für Systemereignisse, die das UP Tool aufzeichnet bzw. auswertet.*

Eine weitere Eigenschaft von UPM ist die Möglichkeit, das Programm mit sehr geringem Aufwand auf beliebigen räumlich entfernten Zielrechnern zur Verfügung zu stellen und mit Hilfe einer Browser-integrierten Version Massendaten zu sammeln.

### 3 Diskussion und Ausblick

Das Tool wird derzeit innerhalb des Softwareentwurfs- und Evaluationsprozesses eingesetzt. Da sich das UPM Tool sehr gut dazu eignet, relativ schnell Massendaten für die Details von Interaktionsabläufen zu sammeln, liegt es nahe, aus diesen Massendaten typische Interaktionsmuster zu extrahieren (z.B. mit pfadanalytischen Methoden) und diese beobachteten „gemittelten“ Interaktionsabläufe mit den Abläufen zu vergleichen, die als Hypothesen dem Design der verwendeten Software zugrunde liegen. Abbildung 1 zeigt eine schematische Darstellung zweier beobachteter Nutzungssequenzen. Weiterhin ist geplant, (Bildschirm-)Koordinaten als weitere Referenz für die Kalibrierung einzubinden.

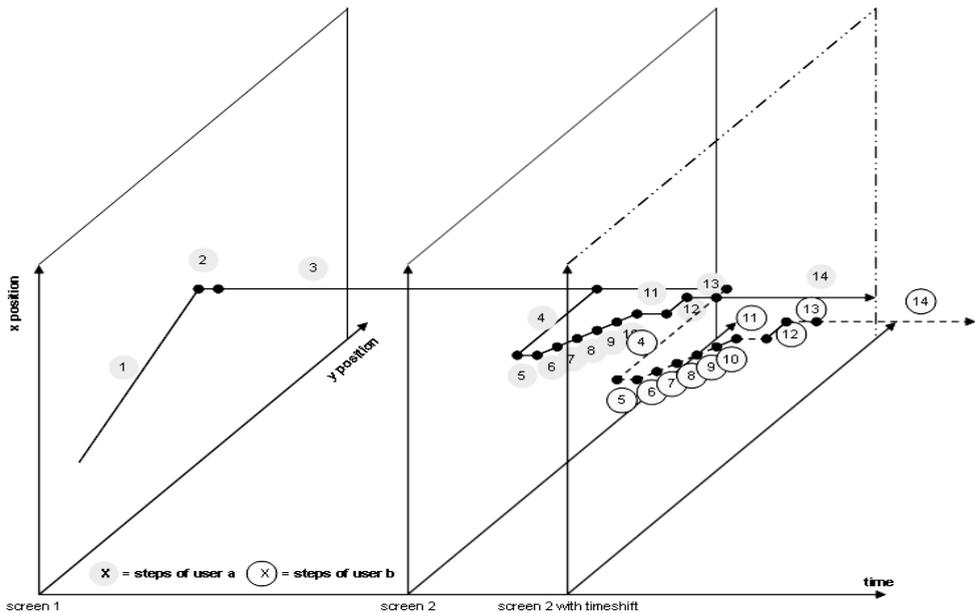


Abbildung 1: Zwei schematische Nutzungssequenzen beim Übergang von einem Bildschirminhalt (screen 1) zum nächsten (screen 2). Der Übergang erfolgt zeitversetzt

## Literaturverzeichnis

- Anderson, J. R. & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah: Lawrence Erlbaum Associates.
- Gray, W. D., John, B. E. & Atwood, M. E. (1993). Project Ernestine: A validation of GOMS for prediction and explanation of real-world task performance. *Human-Computer Interaction*, 8, 237-309.
- 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, 320-351.
- 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.

## Kontaktinformationen

Dr. Theo Held, Dietrich Mayer-Ullmann: SAP AG, CRMMGMT, Dietmar-Hopp-Allee 16, 69190 Waldorf, {theo.held | dietrich.mayer-ullmann}@sap.com