

Gastherausgeber:

Sandro Leuchter
Martin C. Kindsmüller
Dirk Schulze-Kissing
Leon Urbas

*MoDyS Research Group
Zentrum Mensch-Maschine-Systeme
Technische Universität Berlin*

MMI-Interaktiv

**Modellierung und Simulation in
Mensch-Maschine-Systemen**

ISSN: 1439-7854

<http://useworld.net/mmij/musimms>

MMI-Interaktiv ist ein Online-Journal im Bereich Mensch-Maschine
Interaktion und Human Factors. Es wird getragen von

Institut für Arbeitswissenschaft, RWTH Aachen
Zentrum Mensch-Maschine-Systeme, TU Berlin
Institut für Ergonomie und Designforschung, Uni Duisburg-Essen
Zentrum Mensch-Maschine-Interaktion, TU Kaiserslautern
Institut für Arbeitspsychologie, ETH Zürich

Alle Beiträge sind durch einen Review-Prozess gegangen und
wurden von mindestens zwei Experten begutachtet.

Inhalt

Editorial zum Themenheft Modellierung und Simulation in Mensch-Maschine-Systemen <i>S. Leuchter, M.C. Kindsmüller, D. Schulze-Kissing & L. Urbas</i>	1
Modellierung und Simulation menschlichen Verhaltens als Methode der Mensch-Maschine-System-Forschung <i>M.C. Kindsmüller, S. Leuchter, D. Schulze-Kissing & L. Urbas</i>	4
Choosing and getting started with a cognitive architecture to test and use human-machine interfaces <i>F.E. Ritter</i>	17
Extending ACT-R for modeling dynamics and timing for operating human-machine systems <i>S. Leuchter & L. Urbas</i>	38
Opportunities and challenges of modeling user behavior in complex real world tasks <i>W. Schoppek & D.A. Boehm-Davis</i>	47
Antizipative Modellierung des Benutzerverhaltens mit Hilfe von Aktionsvorhersage-Algorithmen <i>A. Künzer, F. Ohmann & L. Schmidt</i>	61
Ansätze zur kognitiven Simulation eines Autofahrers <i>D. Krajzewicz & P. Wagner</i>	84
Akteursbasierte Simulation in Virtuellen Welten <i>T. Asselmeyer-Maluga, P. Frank, A. Hoheisel, H. Rosé & B. Walter</i>	98
Serviceteil: Modellierung und Simulation für Mensch-Maschine-Systeme	106

Editorial zum Themenheft

Modellierung und Simulation in Mensch-Maschine-Systemen

SANDRO LEUCHTER, MARTIN C. KINDSMÜLLER,
DIRK SCHULZE-KISSING & LEON URBAS

MoDyS Research Group, Zentrum Mensch-Maschine-Systeme, TU Berlin

1. Vorgeschichte: Vernetzen der Modellierungsgruppen

Im Herbst 2000 hat die VolkswagenStiftung die Nachwuchsforschergruppe MoDyS am Zentrum Mensch-Maschine-Systeme eingerichtet. Das Forschungsziel ist die Entwicklung und Bewertung von Methoden zur Modellierung und Simulation in dynamischen Mensch-Maschine-Systemen. Obwohl dies die Modellbildung sowohl der technischen Komponenten als auch der Bedienung und der Voraussetzungen dafür einschließt, liegt der Schwerpunkt der Arbeiten in der Gruppe auf der Entwicklung neuer Methoden für die Modellierung der kognitiven Vorgänge beim Bedienen. Es ist das Ziel, die Schnittstellen der Mensch-Maschine-Interaktion in Hinblick auf Effektivität und Effizienz zu optimieren.

Da dieser Bereich in der Forschungs- und Ausbildungslandschaft im deutschsprachigen Raum noch nicht zum Mainstream bzw. zum Stand der Technik gehört, ist es sowohl notwendig als auch nützlich, sich mit anderen Gruppen zu vernetzen. Die vorliegende Schwerpunktausgabe zum Thema Modellierung und Simulation in Mensch-Maschine-Systemen ist ein Ergebnis der gemeinsamen Diskussion und dokumentiert den Stand der Forschung bei der Anwendung dieser Methode in der Mensch-Maschine-Systemtechnik.

2. Beiträge: Aufspannen des Modellierungsraumes

Die Beiträge beziehen sich auf Simulationen und Modellierungstechniken, die einen Bezug zum *human factor* in Mensch-Maschine-Systemen haben. Bei der Planung der Themen spielten mehrere Dimensionen eine Rolle: Herkunftsdomäne und Anwendungszweck der Simulationen und Modellierungstechniken sowie Modellierungsansatz bzw. -methode. Es war ein wichtiges Ziel die Auswahl der Beiträge so zu gestalten, dass ein möglichst großes Gebiet in dem durch diese Dimensionen aufgespannten Raum abgedeckt wird.

Die Beiträge kommen aus erfreulich unterschiedlichen Domänen, in denen eine Betrachtung vom Mensch-Maschine-System-Standpunkt geschieht: Im Bereich Verkehr gibt es Beiträge über die Flugführung (Schoppek & Boehm-Davis) und die Kraftfahrzeugführung (Krajzewicz & Wagner). Der Maschinenbau wird in der Anwendung autonomer Produktionszellen aufgegriffen (Künzer et al.). Human-Computer-Interaction (Ritter) und chemische Prozesstechnik (Leuchter & Urbas) werden zwar nicht detailliert modelliert, sind aber Ursprung des Interesses, das zu hier präsentierten Modellierungsfragestellungen führte.

Der Anwendungszweck von Modellen und Simulationen in der Mensch-Maschine-Systemtechnik kann neben dem reinen Streben nach Erkenntnisgewinn (Asselmeyer-Maluga et al.) in der Prädiktion von Bedienungsverhalten für die Aufgaben- und Schnittstellengestaltung (Schoppek & Boehm-Davis, Krajzewicz & Wagner), in der Unterstützung der Interaktion durch adaptive Interfaces (Künzer et al.), in der Analyse von Aufgaben und Bedienvorgängen zur Vorbereitung von Trainingsmaßnahmen und in der modellgestützten Fehlerdiagnose und -klassifizierung liegen.

Die Beiträge entstammen etwa zur Hälfte aus dem methodischen Bereich der Modellierung mit kognitiven Architekturen (Ritter, Schoppek & Boehm-Davis, Leuchter & Urbas), aber auch ad hoc Modelle sind vertreten (Krajzewicz & Wagner). Außerdem werden KI-orientierte Methoden mit statistischen Benutzermodellen (Künzer et al.) und der agentenorientierten Simulation in virtuellen Welten (Asselmeyer-Maluga et al.) vorgestellt. Unabhängig von den Modellierungsmethoden wird auch die Auswahl und das Erlernen von Modellierungswerkzeugen (Ritter) sowie die Kommunizierbarkeit von Modellen (Kindsmüller et al.) thematisiert.

Die Anordnung der Artikel in der Schwerpunktausgabe erfolgt schließlich nach dem Darstellungsbereich: Die ersten drei Beiträge liefern einen Überblick über allgemeine Fragestellungen (Ritter, Kindsmüller et al., Leuchter & Urbas), die zur Modellierung relevant sind, die folgenden drei Beiträge stellen konkrete Modelle vor (Schoppek & Boehm-Davis, Künzer et al., Krajzewicz & Wagner) und der letzte Beitrag beschreibt eine Modellintegrationsumgebung (Asselmeyer-Maluga et al.).

3. Service: Landkarte der Infrastruktur der *scientific community*

Die Schwerpunktausgabe Modellierung und Simulation in Mensch-Maschine-Systemen enthält am Ende auch einen Service-Teil, in dem Neueinsteiger in das Feld der Modellierung und Simulation in Mensch-Maschine-Systemen einen Überblick über die „Infrastruktur“ in diesem Forschungsbereich erhalten können.

Der Service-Teil soll als „roter Faden“ dienen, an dem das Gebiet erkundet und erschlossen werden kann. Neueinsteiger können sich dem Gebiet auf ganz unterschiedlicher Ebene mit unterschiedlichen Zielstellungen nähern. So sind für die möglichen Zielgruppen Studierende, Praktiker und Forscher unterschiedliche Infrastrukturelemente wichtig. In unserer Auswahl und Kommentierung haben wir versucht, möglichst viele dieser Bedürfnisse zu adressieren. Mit dem Serviceteil wollen wir auch dokumentieren, dass es hier ein aktives und zukunftsweisendes Forschungsgebiet gibt. Wie im Studien- und Forschungsführer Künstliche Intelligenz (Bibel et al. 1987) demonstrieren wir, welche Anstrengungen gegenwärtig unternommen werden,

um Modellierung und Simulation in Mensch-Maschine-Systemen voranzubringen und zu institutionalisieren.

Gerade in neuen Forschungsgebieten ist nicht nur der Erkenntniszuwachs sehr dynamisch, auch die „Infrastruktur“, mit der sich die *scientific community* vernetzt, ist noch nicht erstarrt und deshalb im Fluss. Folglich muss die als Service-Teil vorliegende Momentaufnahme der Landschaft mit den Änderungen Schritt halten und aktualisiert werden. Wir werden dazu das Mensch-Maschine-Interaktions-Portal <http://useworld.net>, in dem MMI-Interaktiv gehostet wird, nutzen und in der Kategorie „Methoden/Modellierung & Simulation“ den Stand der *community* weiter dokumentieren, um Neueinsteigern mit Hilfe unseres roten Fadens einen Weg zu weisen. Alle Modelliererinnen und Modellierer im Mensch-Maschine-Systeme-Bereich sind herzlich eingeladen, ihre eigenen Fäden dort anzuknüpfen.

Die Gastherausgeber möchten sich an dieser Stelle bei den Autorinnen und Autoren der einzelnen Beiträge und den Reviewern für die gute Zusammenarbeit und die interessanten Beiträge bedanken. Wir wünschen den Leserinnen und Lesern dieses Schwerpunktheftes von MMI-Interaktiv, dass sie aus den Artikeln Nutzen ziehen können. Ergreifen Sie die Möglichkeit die Beiträge in useworld.net zu kommentieren und zu diskutieren.

Berlin im Juni 2004

Sandro Leuchter, Martin C. Kindsmüller, Dirk Schulze-Kissing & Leon Urbas

4. Literatur

Bibel, W., Eisinger, N., Schneeberger, J. & Siekmann, J. (1987). *Studien- und Forschungsführer Künstliche Intelligenz*. Berlin: Springer-Verlag.

Modellierung und Simulation menschlichen Verhaltens als Methode der Mensch- Maschine-System-Forschung¹

MARTIN CHRISTOF KINDSMÜLLER, SANDRO LEUCHTER, DIRK SCHULZE-KISSING &
LEON URBAS

MoDyS Research Group, Zentrum Mensch-Maschine-Systeme, TU Berlin

1. Eine Standortbestimmung

Modelle und Simulationen gehören, sowohl bei der Entwicklung und Optimierung technischer Prozesse, als auch bei den technischen Anteilen der Produktentwicklung, seit langem zum Methodeninventar der Ingenieure. Die zunehmende Komplexität industrieller Prozesse sowie die damit verbundenen steigenden Anforderungen bei der Bedienung der Artefakte mit denen diese Prozesse gesteuert werden erfordern jedoch eine zunehmende Orientierung an den Möglichkeiten und Grenzen des menschlichen kognitiven Systems, in anderen Worten, die adäquate Berücksichtigung des *human factor*. Der damit einhergehende Paradigmenwechsel hin zur systemtechnischen Betrachtungsweise des Gesamtsystems aus Mensch und Maschine (vergleiche Mensch-Maschine-Systemtechnik; Timpe, Jürgensohn & Kolrep 2002) gilt in vergleichbarer Weise auch für Fahrzeuge, Luftfahrzeuge oder Konsumprodukte wie Mobiltelefone oder „Persönliche Digitale Assistenten“ (PDA). Dennoch wird Modellierung und Simulation für den Bereich des menschlichen Verhaltens innerhalb eines Mensch-Maschine-Systems, beispielsweise zur Verbesserung der Mensch-Maschine-Interaktion noch vergleichsweise wenig eingesetzt.

Ziel des Artikels ist die Identifizierung potentieller Barrieren eines solchen Einsatzes und das Aufzeigen von Maßnahmen zu deren Überwindung. Gegenstand der Betrachtung ist sowohl die aktuelle betriebliche Praxis und die derzeitigen Randbedingungen des Einsatzes von Modellierungs- und Simulationsmethoden, als auch die Weiterentwicklung dieser Methoden an Universitäten und außeruniversitären Forschungseinrichtungen. Des Weiteren werden Anforderungen abgeleitet und der Ent-

¹ Dieser Artikel basiert im Kern auf Ergebnissen des Arbeitskreises „Transparenz gestalten – Einsatz von Modellierung und Simulation in Mensch-Maschine-Systemen“ der 4. Berliner Werkstatt Mensch-Maschine-Systeme.

wicklungsbedarf in Bezug auf Theorien, Modellierungsmethoden und Simulationswerkzeugen definiert.

2. Modellierung, Simulation, Modell

Unter Modellierung (synonym zu Modellbildung) soll im Folgenden die Erstellung eines vollständigen formalen und ablauffähigen Modells eines technischen und/oder kognitiven Prozesses verstanden werden, während Simulation (oder auch Simulationsexperiment) das „Laufenlassen“ eines solchen Modells bezeichnet.

Ein Modell eines Systems wird aufgrund eines konkreten Bedarfs, der Problemstellung (s. Abbildung 1), entwickelt. Über ein konzeptionelles Modell (Norman 1983), das möglicherweise auch nur implizit, d.h. im Kopf des Modellierers, bei der Formulierung eines konkreten formalen Modells vorhanden ist, wird ein mathematisches Modell erzeugt. Der Modellzweck liegt zumeist nicht nur in der Beschreibung des Systems, beispielsweise zur Dokumentation oder um sicher darüber kommunizieren zu können. Stattdessen sollen Simulationsexperimente einen Einblick in das Verhalten unter bestimmten gegebenen Bedingungen ermöglichen. Dazu muss eine ablauffähige Version dieses Modells z.B. in Form eines Computerprogramms entwickelt werden, mit dem in Simulationsläufen Ergebnisse auf der Basis wirklichkeitsanaloger Prozesse generiert werden können. Aus Abbildung 1 wird weiterhin deutlich, dass es sich bei der Modellierung um einen rückgekoppelten Prozess handelt, bei dem die Überprüfung und Interpretation der Ergebnisse eines Arbeitsschrittes mögliche Nachbearbeitung erfordern. Insbesondere durch die Auswirkungen der Simulationsergebnisse auf die Problemstellung ist die Modellierung in der Regel ein zyklischer Prozess.

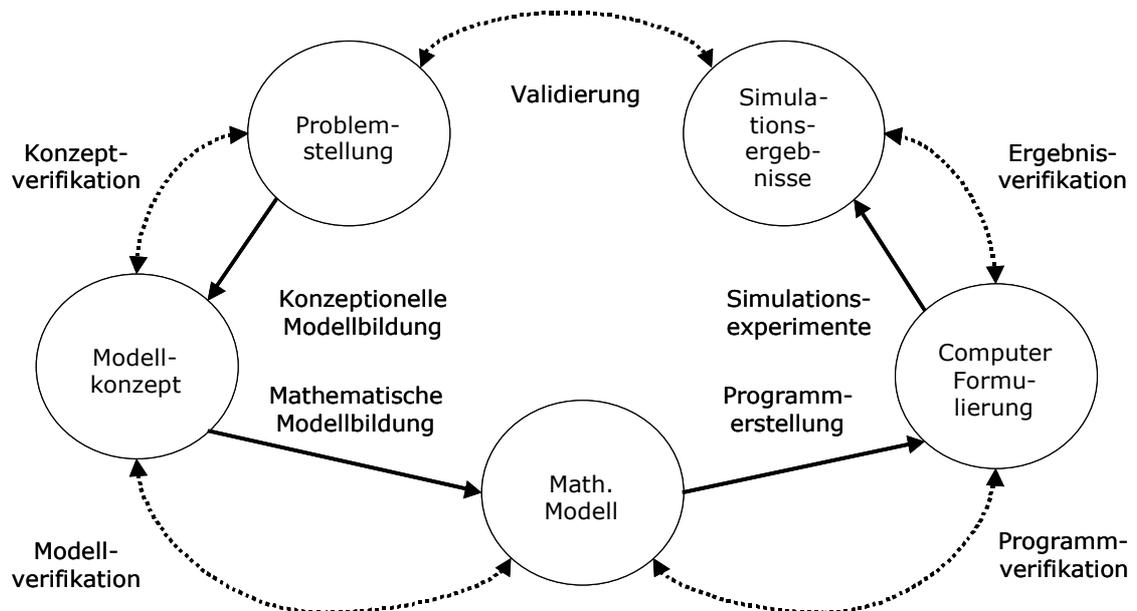


Abbildung 1: Modellbildung und Simulation (nach Bub & Lugner 1990).

Doch wie kann dieser Prozess genutzt werden, um beispielsweise die Systemtransparenz in einem Mensch-Maschine-Systemen zu erhöhen? Bereits in sehr frühen Pha-

sen des Entwurfs technischer Systeme wird – gewollt oder ungewollt – festgelegt, wie transparent sich das System darstellen wird. Selbst wenn dem Entwickler die Wirkungen von Veränderungen am technischen (Teil-)System auf dessen zukünftiges Verhalten bekannt sind, so ist dieses Wissen noch nicht ausreichend, um das zukünftige Verhalten des gesamten Mensch-Maschine-Systems zu prognostizieren. Durch Veränderungen des technischen Teilsystems werden die an die Bediener gerichteten Anforderungen, die zum Funktionieren des Gesamtsystems erforderlich sind, ebenfalls verändert. In frühen Phasen des Entwurfs wird ohne Hinzuziehen expliziter Modelle menschlichen Verhaltens das Gesamtverhalten des zukünftigen Systems daher möglicherweise falsch eingeschätzt. Dies kann zu falschen Entscheidungen bei der grundsätzlichen Systemauslegung führen, deren Effekte dann im weiteren Entwicklungsprozess möglicherweise noch kompensiert, aber nicht mehr beseitigt werden können.

Für die Simulation solcher Menschmodelle gibt es sehr unterschiedliche Problemstellungen, welche die Modellierung beeinflussen und zu einer Vielzahl von sich ergänzenden Ansätzen führen. Es können deskriptive Modelle der Physiologie von Bedienern, der kognitiven Vorgänge von Benutzern oder auch normative Modelle von Arbeits- oder Bedienabläufen eingesetzt werden, um Fragen der Systemgestaltung zu beantworten. Oft steht die Frage nach Bedienfähigkeiten oder Kapazitätsgrenzen im Vordergrund der Systemgestaltung, um den Möglichkeitsraum denkbarer Systementwürfe einzugrenzen.

3. Erfahrungen beim Einsatz von Menschmodellen

Dieser Abschnitt umfasst eine Sammlung von Erfahrungen und *best practices* beim Einsatz von Menschmodellen in der Mensch-Maschine-System-Forschung im deutschsprachigen Bereich. Sie ist als Ergänzung zu den in diesem *special issue* vertretenen Artikeln zu sehen. Die Sammlung orientiert sich an den Ergebnissen des Arbeitskreises „Transparenz gestalten – Einsatz von Modellierung und Simulation in Mensch-Maschine-Systemen“ der 4. Berliner Werkstatt Mensch-Maschine-Systeme (Kindsmüller, Leuchter & Urbas 2002) und wurde um neuere – im Wesentlichen auf der 5. Berliner Werkstatt Mensch-Maschine-Systeme (Steffens, Thüning & Urbas 2004) vorgestellte – Ansätze ergänzt. Ziel ist die Darstellung der Gemeinsamkeiten aber auch der Heterogenität und Vielschichtigkeit der Ansätze, ohne dass ein Anspruch auf Vollständigkeit erhoben würde. Ergänzend zu den

Explizit nicht enthalten sind Ansätze, die sich auf rein physische Mensch-Modelle beziehen. Einen Überblick zum Thema physische Mensch-Modelle liefern die Tagungsbände zur *Digital Human Modeling Conference* (z.B. DHMC 2002).

Produktion: Tätigkeitsabläufe in autonomen Produktionszellen wurden durch Beobachtung und mit Hilfe von Interviews erhoben. Die so entstandene Aufgabenbeschreibungen wurden mit gefärbten Petrinetzen (Jensen 1998) formalisiert und zum Softwareentwurf herangezogen. Um in den frühen Phasen der Modellierung einen Überblick zu gewinnen, wurden UML-Modelle (*unified modeling language*) (Jacobson, Booch & Rumbaugh 1999) verwendet. Formalisierung und Simulation erfolgten rechnergestützt. Papiermodelle wurden in Diskussionen und zur Ableitung von Anforderungen herangezogen. Mit den verwendeten Formalismen und Werkzeugen konnte bei Modellen begrenzter Komplexität über gute Erfahrungen berichtet werden (Reuth, Künzer, Boldt Schmidt, Luczak & Murrenhoff 2002).

Streckenflugkontrolle: Mit dem Ziel, die komplexen mentalen Prozesse von Fluglotsen zu beschreiben, wurde aufgrund von Verhaltensdaten, die mit Experimenten gewonnen wurden, ein Modell in ACT (*adaptive control of thought*) (Anderson & Lebiere 1999) entwickelt. Die initiale Erstellung der Modells erfolgte zunächst aus weitgehend grundlagengetriebenem Erkenntnisinteresse: „die Instrumente der Kognitionsforschung auf die Spitze treiben“, also ohne explizite Anwendungsorientierung. Erst im Nachhinein wurden Simulationen des Modells im Lotsentraining (Leuchter & Jürgensohn 2000) und zur Gestaltung von Unterstützungssystemen angewendet (Niessen & Eyferth 2001).

Avioniksysteme (1): In sehr sicherheitskritisch Bereichen wie Avioniksystemen ist es unbedingt notwendig die Bedienbarkeit von Mensch-Maschine-Schnittstelle zu optimieren. Ein Ansatz der Optimierung besteht darin Bedienungsfehler nachzuvollziehen beziehungsweise vorherzusagen. Dazu wurde ein Modell mit Mittel-Ziel-Regeln in Prolog realisiert (Lüdtke & Möbus 2002) und um eine Komponente, die das psychologische Konstrukt „gelernte Sorglosigkeit“ (Frey & Schulz-Hardt 1996) realisiert, ergänzt (Lüdtke & Möbus 2004). Simulationsexperimente des Modells, das als Kern den fehlerproduzierenden Lernprozess (gelernte Sorglosigkeit) implementiert, können die empirisch gefundenen Fehler (Kontrollstrukturverletzungen) sehr gut vorhersagen. Die Güte des Pilotenmodells ist für den zukünftigen Einsatz als Werkzeug zur Unterstützung beim Design interaktiver Systeme bereits ausreichend um zu untersuchen welche Bedienkonzepte der Gefahr unterliegen durch „gelernte Sorglosigkeit“ unzulässig modifiziert zu werden.

Kleine interaktive Geräte: Zur Abschätzung von Bedien- und Lernzeiten kleiner interaktiver Geräte mit beschränkter Komplexität (wie beispielsweise Mobiltelefone, CD-Player etc...) werden während der Softwareentwicklung Systemspezifikationen semiautomatisch in GOMS-Modelle (Kieras 1999) überführt mit deren Hilfe die Bedienung modelliert wird. Dazu steht das rechnergestützte Werkzeug TREVIS (Marrenbach, Maaßen & Kraiss 2000) zur Verfügung. Im Rahmen der GOMS-Theorie und deren Beschränkungen konnten sehr gute Erfahrungen mit der frühzeitigen Evaluation von Prototypen-Spezifikationen gewonnen werden. Mit den gleichen Methoden wurden elektronische Bedienungsanleitung erfolgreich bewertet (Hamacher, Marrenbach, Zieren & Kraiss 2002).

Chemische Prozessführung: Modelle von Anlagenbedienern in der chemischen Prozessindustrie sollen in Trainingssystemen eingesetzt werden, um Strategien zum Umgang mit nebenläufigen Zielen und Aufgaben sowie zum Umgang mit Zeitkonstanten und anderen Eigenarten, die aus der Dynamik der Aufgabe erwachsen, zu einzuüben. Die Modelle von erfahrenen Bedienern liefern im Sinne eines Unterstützungssystems normative Daten über situationsabhängige Gedächtnisinhalte und kognitive Vorgänge zur Adaption einer Trainingssimulation an die jeweiligen Trainees (Leuchter & Urbas 2002). Ein wesentliches Merkmal bei der Modellierung solcher Vorgänge ist der Umgang mit Zeitdauern. Untersuchungen der MoDyS-Gruppe zeigen, dass die Wahrnehmung und Verarbeitung von Zeitdauern von der kognitiven Beanspruchung bei der Bedienung abhängen (Schulze-Kissing et al. 2003). Entsprechend wird gegenwärtig an einer Erweiterung der kognitiven Architektur ACT-R/PM um eine entsprechende Zeitdauerverarbeitung gearbeitet. Mit dieser für die Modellierung von polytelischen Aufgaben wesentlichen Erweiterung wird sowohl ein Scheduling paralleler Aufgaben mit Zeitdauerbedingungen (z.B. Streckenflugsicherung), als

auch die Modellierung übergeordneter zeitdauerbasierter Aufgaben (insbesondere in der Prozesskontrolle in chemischen Anlagen) möglich.

Effiziente Modellerstellung: Um eine ingenieurmäßige Bedienermodellierung in Mensch-Maschine-Systemen zu gewährleisten, werden gegenwärtig Anstrengungen unternommen, die Wiederverwendbarkeit von Teilmodellen zu erhöhen und bessere Werkzeuge für die Modellerstellung und –interpretation zur Verfügung zu stellen. Die Wiederverwendbarkeit von Teilmodellen wird durch den sogenannten Compilation-Ansatz verbessert: Einfachere Modelle, wie sie z.B. direkt aus einer Aufgabenanalyse in GOMS vorliegen, werden durch die Anwendung von komplexen Transformationsregeln in Modelle mit einer weitergehenden Aussagekraft z.B. in ACT-R/PM transformiert (z.B. simple-act: Salvucci & Lee 2003). Eine andere Möglichkeit, die Wiederverwendbarkeit zu erhöhen ist die Bereitstellung von Teilmodellbibliotheken. Das von der MoDyS-Gruppe entwickelte agimap-System bietet eine Programmieroberfläche für die bibliotheksbasierte Modellierung der Informationsverarbeitung in displaygestützten Schnittstellen (Leuchter & Urbas in Druck). Während es für die Visualisierung von kognitiven Simulationen bereits einige Werkzeuge gibt (z.B. CaDaDis: Tor et al. 2004), ist die Visualisierung von Programmen in diesem Bereich völlig unbekannt. Aktuelle Arbeiten in der MoDyS-Gruppe zielen auf die Visualisierung der Kontrollflüsse in ACT-R-Modellen ab, um den Modellierungsprozess softwaretechnisch zu unterstützen und die Interpretierbarkeit fremder Modelle zu verbessern.

Prozessführung (Mikrowelt CAMS): Ziel der Modellierung mit Hilfe gefärbter Petrinetze (Jensen 1998) war eine Optimierung der Aufgabenallokation zwischen Mensch und Maschine, um so über eine Reduzierung der Belastung die Wahrscheinlichkeit des Auftretens menschliche Fehlleistungen zu senken (Werther 2004). Dazu wurde zunächst ein normatives Ressourcenmodell erstellt. Das Modell wird derzeit so weiterentwickelt, dass quantitative Bewertungsmetriken für mentale Belastung und *situation awareness* (Endsley 1995) so wie Interferenz- und Konkurrenzeffekte bei Mehrfachaufgaben abgeleitet werden können. Der Einsatz gefärbter Petrinetze erlaubt die Modellierung eines Mensch-Maschine-Systems als Ganzes. Die graphischen Darstellungsmöglichkeiten des Ansatzes erleichtern Verständnis und Kommunizierbarkeit des Modells.

Avioniksysteme (2): Zur Onlinediagnose von Flugführungsaufgaben zur Anwendung in Trainingsystemen wurde mit Hilfe von Data-Mining-Methoden ein Klassifikator erstellt, der für einzelne Flugabschnitte die Güte der gewählten Trajektorie beurteilt (Mehl & Köster 2004). Der Klassifikator erlaubt die Identifikation der, für die Wahl einer optimumnahen Trajektorie, ursächlichen Parameterausprägungen auf der Basis von insgesamt 16 Variablen. Ein Vergleich der vorhergesagten Parameterausprägungen einer optimumnahen Trajektorie mit den aktuell vorliegenden Parameterwerten eines Trainees erlauben es noch während einer Trainingsmaßnahme gezielt an den Defiziten des Trainees zu arbeiten

An den referierten Beispielen zeigt sich, wie groß der nutzbringende Einsatzbereich über Domänen und Methoden ist (vergleiche Tab. 1). Sie zeigen auch, welche unterschiedliche Arten von Modellen eingesetzt werden. Je nach Aufgabenstellung und Modellzweck ergeben sich sehr unterschiedliche Sichtweisen. Über die Formalisierung und Simulation von Arbeitsabläufen mehrerer Agenten in Geschäftsprozessen, Petrinetzen oder objektorientierten Modellen über die GOMS-Bedienermodelle bis hin zu kognitiv orientierten Bedienermodellen finden sich verschiedene Ansätze. Die

Modelle haben teils normativen, teils deskriptiven Charakter. Der daraus resultierende Werkzeugbedarf ist uneinheitlich.

Tabelle 1: Möglichkeitsraum der Modellierung in Mensch-Maschine-Systemen: Ziele, Domänen, Methoden und Werkzeuge

Zielsetzung	Domäne	Methode	Werkzeuge	Ergebnisse
Aufgabenbeschreibung	Produktion	gefärbte Petri-netze, UML	verschiedene (papier- & rechnergestützt)	Anforderungen ableiten Softwareentwurf
Modellierung komplexer Prozesse	Streckenflugkontrolle	Modellierung mit kognitiven Architekturen	ATC-R	Training & Gestaltung von Unterstützungssystemen
Rapid Evaluation	kleine interaktive Geräte	GOMS-Modellierung	GOMS, TREVIS	relativ robuste Vorhersage der Lern- & Bedienzeiten
Bedienfehler vorhersagen	Avionik	Mittel-Ziel-Analyse	PROLOG	Erklärung von Bedienfehlern
Trainingsunterstützung	chemische Prozessführung	Modellierung mit kognitiven Architekturen	ACT-R/PM	modellgestütztes Situation Awareness Training
Effiziente Modellerstellung	displaygestützte Schnittstellen	Modellierung mit kognitiven Architekturen	ACT-R/PM	agimap-Werkzeug Programmvisualisierung
Optimierte Aufgabenallokation	Prozessführung (Mikrowelt CAMS)	gefärbte Petri-netze	Design CPM	Leistungsdaten eines kognitiven Modells bei der Abarbeitung verschiedener Handlungsfolgen
Onlinediagnose	Avionik	Data-Mining	EA-MOLE	Klassifikator für die Güte von Flugverlaufsabschnitten

3.1 Wie übertragbar sind die Erkenntnisse zwischen den Domänen?

Noch immer sehr kontrovers diskutiert wird die Frage nach der Übertragbarkeit von Methoden und Ergebnissen zwischen den Domänen. Der schwer zu widersprechenden Aussage: „Es sind ja schließlich die gleichen Menschen die alles bedienen (könnten)“ stehen mahnende Äußerungen (aus leidvollen Erfahrungen) entgegen: „Wir müssen domänenspezifisch anfangen – alles andere wäre verwegen“.

Unbestritten weisen Menschen generische Eigenschaften auf, welche sich domänenübergreifend finden lassen sollten (Ernst & Newell 1969; Newell & Simon 1972). Ebenfalls unbestritten existiert domänenspezifisches Wissen, das nur relativ zur Domäne verwertbar ist (Chase & Simon 1973). Ein Großteil der Widersprüche lässt sich auflösen, indem man bei der Modellierung klar zwischen generischen und domänenspezifischen Anteilen in den Modellen unterscheidet. Eine Übertragung zwischen Modellen verschiedener Domänen ist demnach umso erfolgversprechender, je höher der generische Anteil im Modell ist, beziehungsweise je ähnlicher die modellierten Domänen sind.

Ein weiterer Vorschlag zur Verbesserung der domänenübergreifenden Modellbildung, der vorsieht nicht die gesamte Interaktion eines Menschen mit dem techni-

schen System zu modellieren, sondern sich zunächst auf die domänenübergreifend geltenden Ausschnitte zu beschränken, ist derzeit ebenfalls umstritten. Ergebnis eines solchen Vorgehens wären dann Elementarmodelle spezifischer kognitiver Funktionen. Der Nutzen dieser Elementarmodelle ist jedoch umstritten; können diese doch kaum mehr als Zeiten und Fehlerraten liefern, die man ebenso gut und mit weniger Aufwand aus der bestehenden psychologischen Literatur entnehmen oder mit vergleichsweise preiswerten Experimenten bestimmen könnte.

3.2 Erwartungen an die Modellierungsmethode

Weitgehende Einigkeit besteht hinsichtlich der Erwartung, dass Menschmodelle als Entscheidungshilfe in frühen Phasen der Entwicklung eingesetzt werden könnten und in diesem Bereich in Zukunft eine große Rolle spielen werden. Der Aufwand dafür darf jedoch nicht zu hoch sein und es scheint, vor allem im Gegensatz zu anderen Methoden (z.B. Experiment), vergleichsweise schwierig zu sein den Aufwand im Vorfeld abschätzen zu können. Der Einsatz simulierter Menschmodelle wird nicht nur in diesem Zusammenhang eine zunehmend attraktive Alternative zum „klassischen Experiment“ darstellen. Die Hoffungen auf simulierte Menschmodelle als Ersatz für Experimente nähren sich aus der Tatsache, dass Experimentieren vergleichsweise teuer (Versuchspersonengelder) und zeitaufwändig ist. Zudem lassen sich gerade hochkomplexe Verhaltensweisen oft nur schwer bis überhaupt nicht experimentell untersuchen. Sehr feingranulares Experimentieren führt häufig zu der paradoxen Situation, dass wir „immer mehr über immer weniger wissen“. Modellieren und Validieren der vom Modell gelieferten Verhaltensdaten sind dann die einzig verbleibende Alternative, um Aussagen mit einem größeren Gültigkeitsbereich treffen zu können (Newell 1973; 1990).

Auf der anderen Seite werden experimentelle Methoden mitunter zu früh und zu unspezifisch genutzt. In diesen Fällen könnten mit Gewinn Modellierungsmethoden zum Einsatz kommen, um den Möglichkeitsraum so einzuschränken, dass Experimente darauf aufbauend gezielter eingesetzt werden können. Probleme bei den Modellierungsmethoden werden derzeit in ihren oft beschränkten Gültigkeitsbereichen, so wie in den hohen Einstiegskosten bei gesehen.

Betrachtet man den gemeinsamen Kern hinter den positiven Erwartungen und den Befürchtungen, so wird deutlich, dass adäquat eingesetzte Methoden die positiven Erwartungen stützen und gleichzeitig die Befürchtungen weitgehend ausräumen können. Es gibt mit Sicherheit keinen „*one best way*“ zur Entscheidung zwischen Experiment und Modellierung. Anzustreben ist vielmehr die Beherrschung eines integrativen Methodeninventars in dem Modellieren und Experimentieren als sich ergänzende Methoden die zentralen Rollen spielen. Anwendbarkeit, Aufwand, Aussagekraft, Gültigkeitsbereich und Kosten stellen dann Parameter dar, die *a priori* abgeschätzt werden müssen, um die adäquate Methode zu finden. Das Ergebnis dieser Abschätzung determiniert dann die Wahl zwischen experimentellen Methoden einerseits und den Modellierungsmethoden andererseits oder legt eine Kombination beider Methoden nahe.

4. Maßnahmen zur Etablierung von Modellierungsmethoden in Mensch-Maschine-Systemen

Der Entwicklung allgemeiner Maßnahmen sind Grenzen gesetzt, weil es keine allgemeinen Modelle gibt. So sind beispielsweise Modelle von Lotsen im Bereich der Streckenflugkontrolle (Niessen, Leuchter & Eyferth 1998) sehr weit von Modellen bei der Fahrermodellierung im KFZ (Jürgensohn 1998) entfernt (Jürgensohn, Nielsen & Leuchter 2000). Dies liegt nicht ausschließlich daran, dass in verschiedenen Domänen unterschiedliche Tätigkeiten modelliert werden, sondern vor allem daran, dass sich die Modellierer bei der Modellbildung auf verschiedene Aspekte in den Tätigkeiten konzentrieren. Neben dieser domänen- bzw. tätigkeitsaspektsbedingten Unterschiedlichkeit der Modelle sind auch innerhalb einer Domäne – oder bei weitgehend vergleichbaren Domänen – Klassen von Modellen zu unterscheiden. So bedingen beispielsweise Tätigkeiten, die eine langjährige Expertise erfordern andere, Modellierungsmethoden (weil die Expertise in das Modell integriert werden muss) und damit auch andere Maßnahmen zur Förderung der Modellierung, als Tätigkeiten bei denen dies nicht der Fall ist.

Es lassen sich dennoch eine Reihe eher allgemeiner Maßnahmen zur Etablierung von Modellierungsmethoden in Mensch-Maschine-Systemen ableiten. Zur Reduzierung von Aufwand und Kosten der Modellierung scheint es notwendig langlebigere Modelle zu schaffen, die in der Lage sind, mehrere Produktzyklen zu überdauern. Diese Maßnahme steht partiell im Widerspruch zur Auffassung, dass versucht werden sollte, sehr kleine (*Ad-hoc*-)Modelle zu entwickeln, um mit dem geringst möglichen Aufwand Produkte evaluieren zu können. Eine Entscheidung über die Adäquatheit dieser beiden Aussagen ist eventuell über die Spezifizierung der Anwendungsdomäne möglich. Kurzlebige (*Ad-hoc*-)Modelle sollten eingesetzt werden, wenn die technische Komponente des Mensch-Maschine-Systems raschem Wandel unterworfen ist, der noch dazu starken Einfluss auf die Interaktionsprinzipien ausübt. Als typische Domäne könnten hier kleine mobile Konsumprodukte (Mobiltelefone, PDAs...) genannt werden. Langlebige Modelle können stattdessen erfolgversprechend in Domänen wie Flugführung, Flugsicherung, Leitwarten, etc. zum Einsatz kommen; hier sind die Produktzyklen wesentlich länger, zudem ändern sich die Systeme von Generation zu Generation nur sehr wenig.

Eine weitere Maßnahme, die der Modellierungsmethode entscheidende Impulse liefern kann, ist die Weiterentwicklung bestehender kognitiver Architekturen wie beispielsweise ACT (Anderson & Lebiere 1999) oder Soar (Rosenbloom, Laird & Newell 1993, Newell 1990). In diesen Architekturen wurde bereits eine Fülle verschiedener Modelle aus diversen Bereichen kognitiver Leistung (von der Ausbildung von Quantifizierungsoperatoren im Kindesalter bis zum Erlernen einer Programmiersprache) implementiert und experimentell validiert. Defizite weisen diese – ansonsten erfolgversprechenden – Architekturen derzeit im Bereich dynamischer Mensch-Maschine-Systeme auf. Eine Modifizierung bestehender kognitiver Architekturen, so dass in diesen dynamische Bedienvorgänge abgebildet werden können, ohne „gegen die Architektur“ modellieren zu müssen, würde die Mensch-Maschine-System-Modellierung auf die breite Basis kognitionspsychologischer Befunde stellen, die in den kognitiven Architekturen bereits implementiert sind.

Nicht unerwähnt bleiben sollte die Tatsache, dass wegen der Diversität und Komplexität von Modellierungsmethoden (aber auch Experimentalmethoden) die gesamt

Methodenkompetenz mitunter nicht in einer Person vereinbar ist. Die deshalb notwendige Verteilung der Kompetenzen auf mehrere Spezialisten erfordert die Schaffung eines gemeinsamen Begriffsraums bzw. einer gemeinsamen Sprache. Dies kann über sorgfältige Teambildungsmaßnahmen so gestaltet werden, dass ein funktionierendes Kompetenznetzwerk aus Einzelkompetenzen entsteht.

5. Anforderungen an kognitive Modelle in der Systementwicklung

Der Einsatz von Benutzermodellen in der Systementwicklung stellt pragmatische Anforderungen an die zum Einsatz kommenden Modellierungsmethoden und -formalismen, die über den Grundanspruch einer an die Aufgabenstellung angemessenen Vorhersagegüte deutlich hinausgehen:

Kommunizierbarkeit von Modellen und Simulationsergebnissen: Die Gestaltung von Mensch-Maschine-Systemen ist eine interdisziplinäre Herausforderung für Designer, Ergonomen, Ingenieure und zunehmend auch Informatiker. Die Modellierung wird – wie im Technikbereich auch – sicherlich wenigen Spezialisten vorbehalten bleiben, es gilt aber wie ebendort, dass die Mechanismen, Grundannahmen und Vereinfachungen die das Verhalten des Simulationsmodells determinieren zwischen den Beteiligten kommunizierbar sein müssen.

Modellierungseffizienz: Während in der grundlagenorientierten Benutzermodellierung maximale Validität und langfristiger Erkenntnisgewinn die Zielgrößen darstellen, ist für den industriellen Einsatz die Modellierungseffizienz das ausschlaggebende Kriterium. Die Vorhersagegüte der Simulationsmodelle muss dabei lediglich der in der aktuellen Entwicklungsphase geforderten entsprechen.

Beide Zielgrößen sind nur durch einen konsequent **systemtechnisch geprägten Modellierungsansatz** erreichbar. Eine **hierarchische Dekomposition** in vernetzbare, aber für sich unabhängig behandelbare (und validierbare) Teilsysteme trägt sowohl zur Kommunizierbarkeit, als auch zur Effizienzsteigerung durch Wiederverwendung bei. Um Fragestellungen effizient beantworten zu können, muss der verwendete Modellierungsformalismus eine **variable Abbildungstiefe** zulassen, d.h. bestimmte Teilsysteme (i.e. kognitiven Prozesse) sollten für die Betrachtung nicht zwingend weiter zerlegt werden müssen. Wird die hierarchische Dekomposition/Komposition mit klar definierten Beziehungen zwischen den Teilsystemen und definierten Schnittstellen zu Teilsystemen (Gedächtnis, interne Uhr, Wahrnehmung, etc.) konsequent durchgehalten, ist darüber hinaus eine Effizienzsteigerung durch eine **hybride Modellierung** möglich, d.h. die Teilsysteme würden mit der jeweils effizientesten Methode beschrieben werden können.

6. Ausblick

Wenn beim Thema Modellierung und Simulation in Mensch-Maschine-Systemen zwar in Teilgebieten des Einen Wunsch die Befürchtung des Anderen darstellt, so lässt sich insgesamt doch ein zunehmendes Interesse an Modellierungsmethoden, sowohl im Bereich der universitären Forschung, wie auch in der industriellen Anwendung konstatieren. Die Verwendung von Menschmodellen wird heute in der Re-

gel nicht mehr in Frage gestellt, es geht vielmehr darum, der Modellierungsmethode im Kanon anderer Methoden den richtigen Stellenwert zuzuweisen.

Die Hoffnung, dass mit Hilfe der Bedienermodellierung zukünftig Probleme gelöst werden können, die experimentell nicht effizient lösbar sind, ist Motivation genug, um in die Weiterentwicklung der Modellierungsmethoden zu investieren. In wenigen Jahren sind dann, beispielsweise im Bereich der Navigationssysteme, Modelle des Blickverhaltens zu erwarten, die valide Aussagen über die „Auswirkungen eines um 10% nach oben verschobenen Displays, auf dessen Ablesbarkeit liefern können“. Heute müssen zur Evaluation eines derartigen Eingriffs verschiedene Prototypen erstellt und in langwierigen Experimentalreihen kosten- und zeitintensiv evaluiert werden.

Diese Arbeit wird von der VolkswagenStiftung im Rahmen des Programms „Nachwuchsgruppen an Universitäten“ unterstützt.

7. Literatur

- Anderson, J. R. & Lebiere, C. (1999). *The Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Bub, W. & Lugner, P. (1990). Modellbildung und Modellreduktion - Systematik der konzeptionellen Modellbildung. In F. Breitenecker, I. Troch, & P. Kopack (Hrsg.) *Simulationstechnik. 6. Symposium in Wien, September 1990* (S. 62-66). Braunschweig: Vieweg.
- Chase, W. G., & H. A. Simon. (1973). The mind's eye in chess. In W. G. Chase, (Hrsg.), *Visual Information Processing* (S. 215-281). New York, NY: Academic Press.
- DHMC (2002). *Digitale Mensch-Modellierung. Tagung München 18.-20. Juni 2002 Digital Human Modeling Conference*. Düsseldorf: VDI-Verlag.
- Endsley, M. R. (1995). Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors*, 37 (1), 32-64.
- Ernst, G. & Newell, A. (1969). *GPS: A Case Study in Generality and Problem Solving*. New York: Academic Press.
- Frey, D. & Schulz-Hardt, S. (1996). Eine Theorie der gelernten Sorglosigkeit. In H. Mandl (Hrsg.), *Bericht über den 40. Kongress der Deutschen Gesellschaft für Psychologie* (S. 604-611). Göttingen: Hogrefe.
- Hamacher, N., Marrenbach, J., Zieren, J. & Kraiss, K.-F. (2002). Generierung normativer Benutzermodelle aus SDL-Spezifikationen. In R. Marzi, V. Karavezyris, H.-H. Erbe & K.-P. Timpe (Hrsg.), *Bedienen und Verstehen. 4. Berliner Werkstatt Mensch-Maschine-Systeme. 10.-12. Oktober 2001*. (S. 86-100). Düsseldorf: VDI-Verlag.
- Jacobson, I, Booch, G. & Rumbaugh, J. (1999), *The Unified Software Development Process*. Reading, MA: Addison-Wesley.

- Jensen, K. (1998). An Introduction to the Practical Use of Coloured Petri Nets. In: W. Reisig and G. Rozenberg (Hrsg.): *Lectures on Petri Nets II: Applications, Lecture Notes in Computer Science Band 1492* (S. 237-292). Heidelberg: Springer.
- Jürgensohn, T. (1998). *Hybride Fahrermodelle. ZMMS-Spektrum Band 4*. Sinzheim: Pro Universitate.
- Jürgensohn, T., Niessen, C. & Leuchter, S. (2000). Bedienermodellierung: Beispiele. In K.-P. Timpe, T. Jürgensohn, H. Kolrep (Hrsg.), *Mensch Maschine Systemtechnik – Konzepte, Modellierung, Gestaltung, Evaluation* (S. 149-177). Düsseldorf: Symposion.
- Kieras, D. E. (1999). *A Guide to GOMS Model Usability Evaluation using GOMSL and GLEAN3*. University of Michigan. Artificial Intelligence Laboratory. Online-Dokument. Letzter Zugriff am 4.11.2001 unter ftp://ftp.eecs.umich.edu/people/kieras/GOMS/GOMSL_Guide.pdf
- Kindsmüller, M. C., Leuchter, S. & Urbas, L. (2002). Arbeitskreis: Einsatz von Modellierung und Simulation in Mensch-Maschine-Systemen - Transparenz gestalten. In R. Marzi, V. Karavezyris, H.-H. Erbe & K.-P. Timpe (Hrsg.), *Bedienen und Verstehen. 4. Berliner Werkstatt Mensch-Maschine-Systeme. 10.-12. Oktober 2001* (S. 318-329). Düsseldorf: VDI-Verlag.
- Leuchter, S. & Jürgensohn, T. (2000). A tutoring system for air traffic control on the basis of a cognitive model. In: J.L. Alty (Hrsg.), *Proceedings of the XVIII. European Annual Conference on Human Decision Making and Manual Control* (S. 275-281). Loughborough, UK: Group D Publications.
- Leuchter, S. & Urbas, L. (in Druck). Useware Engineering mit kognitiven Architekturen. *Useware 2004, Darmstadt 22.-23.06.2004*.
- Leuchter, S. & Urbas, L. (2002). Simulation Based Situation Awareness Training for Control of Human-Machine-Systems. In Valery Petrushin, Piet Kommers, Kinshuk, & Ildar Galeev (Eds.), *IEEE International Conference on Advanced Learning Technologies. Media and the Culture of Learning. Kazan, Russia: Sep 9-12, 2002* (S. 34-39). Palmerston North, New Zealand: IEEE Learning Technology Task Force.
- Lüdtke, A. & Möbus, C. (2002). Prognose von Bedienfehlern durch Routinebildung in teilautonomen Systemen. Konzept und empirische Untersuchung. In R. Marzi, V. Karavezyris, H.-H. Erbe & K.-P. Timpe (Hrsg.), *Bedienen und Verstehen. 4. Berliner Werkstatt Mensch-Maschine-Systeme. 10.-12. Oktober 2001* (S. 164-184). Düsseldorf: VDI-Verlag.
- Lüdtke, A. & Möbus, C. (2004). Überprüfung eines lernenden Pilotenmodells durch Rekonstruktion von Handlungsprotokollen. In C. Steffens, M. Thüring, & L. Urbas (Hrsg.), *Entwerfen und Gestalten. 5. Berliner Werkstatt Mensch-Maschine-Systeme. 8.-10. Oktober 2003* (S. 160-180). Düsseldorf: VDI-Verlag.
- Marrenbach, J., Maaßen, D. & Kraiss, K.-F. (2000). Formale Methode zur Evaluierung der Gebrauchstauglichkeit von technischen Systemen. In: K.-P. Timpe, H.-P. Willumeit & H. Kolrep (Hrsg.), *Bewertung von Mensch-Maschine-*

- Systemen. 3. Berliner Werkstatt Mensch-Maschine-Systeme* (S. 283-296). Düsseldorf: VDI.
- Mehl, K. & Köster, F. (2004). Online Diagnose von Führungsaufgaben. In C. Steffens, M. Thüning, & L. Urbas (Hrsg.), *Entwerfen und Gestalten. 5. Berliner Werkstatt Mensch-Maschine-Systeme. 8.-10. Oktober 2003* (S. 145-159). Düsseldorf: VDI-Verlag.
- Newell, A. (1973). You can't play 20 questions with nature and win: Projective comments on the papers in this symposium. In W.E. Chase (Hrsg.), *Visual Information Processing* (S. 283-308). New York: Academic Press.
- Newell A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.
- Newell, A. & Simon, H. (1972). *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall.
- Niessen, C. & Eyferth, K. (2001). A Model of Air Traffic Controller's Situation Awareness. *Safety Science*, 37, 187-202.
- Niessen, C., Leuchter, S. & Eyferth, K. (1998). A Psychological Model of Air Traffic Control and Its Implementation. In F. E. Ritter and R. M. Young (Hrsg.), *Proceedings of the Second European Conference on Cognitive Modelling* (S. 104-111). Nottingham: Nottingham University Press.
- Norman, D. A. (1983). Some Observations on Mental Models. In D. Gentner and A. L. Stevens (Hrsg.) *Mental Models* (S. 7-14). Hillsdale, NJ: Erlbaum..
- Reuth, R., Künzer, A., Boldt, T., Schmidt, L., Luczak, H. & Murrenhoff, H. (2002). Modellbasierte Gestaltung einer multimodalen Benutzungsschnittstelle zur Unterstützung von Greif- und Spannprozessen beim 3D-Laserschweißen. In R. Marzi, V. Karavezyris, H.-H. Erbe & K.-P. Timpe (Hrsg.), *Bedienen und Verstehen. 4. Berliner Werkstatt Mensch-Maschine-Systeme. 10.-12. Oktober 2001* (S. 55-70). Düsseldorf: VDI-Verlag.
- Rosenbloom, P.S., Laird, J.E. & Newell, A. (1993). *The Soar Papers: Readings on Integrated Intelligence, Vol. 1 & Vol. 2*, Cambridge, MA: MIT Press.
- Salvucci, D.D.; Lee, F.J. (2003). Simple Cognitive Modeling in a Complex Cognitive Architecture. In Stacey Ashlund, Kevin Mullet, Austin Henderson, Erik Hollnagel & Ted White (Hrsg.), *Human Factors in Computing Systems: CHI 2003 Conference Proceedings*. New York: ACM Press.
- Scheer, A.-W. (1998). *ARIS – Modellierungsmethoden, Metamodelle, Anwendungen*. Berlin: Springer.
- Schulze-Kissing, D.; van der Meer, E. & Urbas, L. (2003). The Effect of Temporal Distortions on the Error Diagnosis in the Control of a Complex Technical System. In Theresa Bajo and Juan Lupiáñez (Eds.), *XIII Conference of the European Society of Cognitive Psychology ESCOP 2003* (S. 295-296). Granada: Actas.

- Steffens, C., Thüning, M. & Urbas, L. (2004). *Entwerfen und Gestalten. 5. Berliner Werkstatt Mensch-Maschine-Systeme. 8.-10. Oktober 2003*. Düsseldorf: VDI-Verlag.
- Timpe, K.-P., Jürgensohn, T. & Kolrep, H. (2002). *Mensch-Maschine-Systemtechnik: Konzepte, Modellierung, Gestaltung, Evaluation (2. Auflage)*. Düsseldorf: Symposium.
- Tor, K., Ritter, F. E., Haynes, S. R., & Cohen, M. A. (2004). CaDaDis: A tool for displaying the behavior of cognitive models and agents. In *Proceedings of the 13th Conference on Behavior Representation in Modeling and Simulation* (S. 192-200). Orlando, FL: U. of Central Florida.
- Werther, B. (2004). Modellbasierte Bewertung menschlicher Informationsverarbeitung mit höheren Petrinetzen. In C. Steffens, M. Thüning & L. Urbas (Hrsg.), *Entwerfen und Gestalten. 5. Berliner Werkstatt Mensch-Maschine-Systeme. 8.-10. Oktober 2003* (S. 181-196). Düsseldorf: VDI-Verlag.

Choosing and getting started with a cognitive architecture to test and use human-machine interfaces

FRANK E. RITTER

*Applied Cognitive Science Lab, School of Information Sciences and Technology
The Pennsylvania State University, University Park, PA 16802
frank.ritter@psu.edu*

Keywords: cognitive models, cognitive architectures

1. Introduction

This article provides a tutorial review of creating cognitive models with cognitive architectures to help with human-machine interface design. It is becoming increasingly popular and increasingly possible to consider creating cognitive models to assist in design, particularly of users of computer interfaces, and also of human-machine interfaces.

As these models and modeling mature, we will have an approach that can be used to evaluate and test a wide range of interfaces by simulating the human component of the system, and how humans interact with interface and machine components. The requirements for this approach can be sketched (Kieras, 2003; Ritter, Van Rooy, & St. Amant, 2002) and point to early examples (Byrne, Wood, Sukaviriya, Foley, & Kieras, 1994; Kieras, Wood, Abotell, & Hornof, 1995).

Commercially prepared cognitive models are already being used for system analysis, although in more limited ways than electrical circuit designers can use their CAD/CAM systems. A few notable examples of this work available as integrated systems include the IGEN system (Zachary, Jones, & Taylor, 2002), the Midas system (Laughery & Corker, 1997), Apex (Freed & Remington, 2000), and the Jack

anthropometric system (Badler, Erignic, & Liu, 2002)¹. The commercial systems that are available do not seem extendible by researchers interested in expanding the behavior and capabilities included.

A prototypical example to illustrate what can be done is a model that examines cell phone design (St. Amant, Horton, & Ritter, 2004). Models were created in ACT-R and GOMS that could perform five tasks on ten cell phone designs. These models performed the five tasks across all of the phone designs. The ACT-R model in particular had access to problem solving and interacted with the cell phone images and representations of an interactive environment. Their results were all basically consistent with the empirical data that was later gathered. The resulting models were then used to optimize the design of the phone interfaces, leading to an estimated time savings of about 30%. Given the widespread use of cell phones, this possible time savings cumulated across all users represents several human lifetimes.

The range of human capabilities that can be considered for inclusion in these models is quite broad and much work remains. The current models, while becoming useful when considering many real design decisions, do not yet include very much of human perception and motor output; they do not have very complex or error tolerant models of error correction and recovery when interacting, and they cannot be routinely applied to a complex interface. There is not a lot of sharing of task models and interfaces. This can be put in contrast with the Lisp, Java, and perhaps expert system shell communities, where contributions are shared on a much more regular basis.

This tutorial review examines how to get started with cognitive models in cognitive architectures with some emphasis on using the resulting models to test interfaces. It notes several cognitive architectures that could be or are being used for evaluating interfaces and predicting task time and (often) errors. The article provides practical comments on how to learn to use a cognitive architecture, providing general guidance as well as pointers to specific resources. The article also examines one of the most vexing questions for modelers, that of how to prove or validate the resulting model. The article concludes by noting some of the most exciting, current problems.

2. How to choose an architecture

You will often start your choice of a cognitive architecture with a problem in mind, and this is important. While in the fullness of time the architectures can be expected to become similar because they are all modeling human cognition, they are forces working to keep them different. Like simulations in other domains, their focus on different domains or levels of analysis may keep them somewhat different.

If you are a new researcher, you may be working where an architecture has already been chosen for you. In that case, you may wish to keep in mind its strengths and limitations, and notice your own new research problems.

2.1 Why use a cognitive architecture?

Before proceeding, it is worth noting what cognitive architectures are, and refer to material to explain them and their approach to modeling. Cognitive architectures are

¹ Note that there is a Jack anthropometric system and a JACK intelligent agent system, same first name, but different approaches and different developers (one a US university, the other an Australian company).

an approach to modeling behavior that assumes that there are two components to behavior, the architecture and knowledge. The architecture is composed of cognitive mechanisms that are fixed across tasks and basically fixed across individuals. These mechanisms typically include some form of perception and motor output, some sort of central processor, some working memory or activation of declarative memory, and some way to store and apply procedures. These mechanisms are used to apply task knowledge to generate behavior.

Newell's (1990) book on unified theories of cognition introduces this approach. ACT-R (Anderson & Lebiere, 1998) probably comes the closest to realizing it currently. Newell's book includes a list of reasons for using a cognitive architecture. To briefly summarize some of the most important, a cognitive architecture proposes that the same mechanisms are used for different tasks, which is parsimonious. The cumulation and unification of results to a central source, are aims of science. When the cognitive architecture is realized as a computer program, it supports these aims by using the architecture itself to serve as a focus for unification. The resulting architecture can then be reused and the effort to create it amortized over multiple projects.

The use of a cognitive architecture also allows for model (knowledge) reuse, but this has been done less than I think Newell anticipated. We are finding that reuse of displays of model behavior (Ritter, Jones, & Baxter, 1998; Tor, Ritter, Haynes, & Cohen, 2004) may perhaps be a more approachable way towards reuse. In the cell-phone project (St. Amant et al., 2004), the tasks were reused, but while there were similar models in ACT-R, these had to be created anew here because exact models did not exist.

And finally, the use of a cognitive architecture helps create complete agents opening the way to applications, which is the subject of this article and this special issue. These models can be used to use and thus test interfaces, to serve as opponents or colleagues in synthetic environments, and to run robots.

2.2 Types of architectures

There are several types of architectures that are or that could be used for evaluating interfaces and predicting task time and (often) errors. These include descriptive architectures, symbolic and hybrid architectures, intelligent agent architectures, and connectionist architectures.

The simplest are descriptive architectures like GOMS (John & Kieras, 1996) and the Keystroke-level model (Card, Moran, & Newell, 1983). Models created in these architectures are used to help in system design (e.g., Gray, John, & Atwood, 1993). They are descriptions of behavior rather than generators of behavior for testing interfaces. They can be used to predict the time to do a task, but the actions have to be specified; they do not address problem solving that might be required to do a task. Current work has attempted to make this approach easier to apply to simple interfaces (e.g., Nichols & Ritter, 1995), to automate the more complex behaviors possible (e.g., Freed & Remington, 2000; Matessa, in press), and to unify GOMS and ACT-R creating a higher level language for ACT-R (St. Amant & Ritter, in press). Some of the commercial systems used for interface and system design started with higher level descriptive architectures that computed the time to do larger tasks, but these architectures have tended to migrate towards including information processing and most can now perform the task of interest with an external simulation.

Symbolic and hybrid architectures are most commonly used by researchers in this area. Soar and ACT-R are examples of these two types. Informally, both types can be referred to as cognitive architectures. These architectures support creating knowledge and applying it to situations. The hybrid architectures, ACT-R in particular, tend to provide more action on the level that interface design is currently viewed, that is, reaction times in *ms*, and the possibility of modeling a range of types of errors.

Intelligent agent architectures have been used to explore interfaces (e.g., St. Amant, 2000). They are useful for testing the range of performance of an interface, and for ensuring that an interface can be used to perform a task. They are not designed to make strong predictions about difficulty of use by humans, however. Connectionist architectures have been used extensively in psychology to model behavior, but they have been little used in modeling interaction. They appear to focus on different types of behavior than have been focused on in interface use. They are likely to be useful when modeling the details of perception, and their memory blends and errors are represented to some extent in the hybrid architectures.

2.3 Reviews of architectures

Knowing your potential application will help you choose an appropriate architecture. Architectures have different strengths. Soar, for example, appears to support larger knowledge bases than ACT-R, but does not provide as much support for detailed timing predictions (e.g., Byrne, 2001; Gray & Boehm-Davis, 2000).

There are now several reviews of cognitive architectures that can help you choose an architecture to use. The first review, still helpful although clearly dated, was a special issue of the *SigART Bulletin* (1991). Pew and Mavor's (1998) report is more recent. Their book reviews architectures developed in the US. The architectures they review, such as Soar and ACT-R, are fairly well developed. Further reviews and comparisons of Soar and ACT-R may be helpful, as these are two of the most widely used architectures (Johnson, 1997; Johnson, 1998; Ritter, Shadbolt, Elliman, Young, Gobet, & Baxter, 2003, Appendix B).

Ritter et al.'s (2003) State of the Art Report (as labeled by the publisher, a SOAR report) is an update and extension to Pew and Mavor's report. Their report reviews a set of architectures not included in Pew and Mavor's report, including the Java Agent Construction Kit (JACK: Busetta, Rönquist, Hodgson, & Lucas, 1999), which is a belief-desires-intentions (BDI) architecture implemented in Java; the COGENT meta-architecture (Cooper & Fox, 1998) that has had some success in teaching; and PSI (Detje, 2000; Dörner, 2003), an architecture that includes physiological drives as a basis of emotions. Ritter et al.'s report also includes a review of current problems and directions for research, which is reviewed in the conclusions. A similar report (i.e., including Silverman, Cornwell, & O'Brien, in press) focused on emotion within architectures will be available shortly.

Even these current books do not include several new and a few old architectures that should be mentioned. Langley (1996) and his research group have an architecture that appears to provide a more schema-based approach. Hybrid architectures, such as Clarion (Neveh & Sun, in press; Sun, Merrill, & Peterson, 1998), attempt to create architectures with sub-symbolic and symbolic representations. Further examples include hybrid versions of blending parts of ACT-R, Soar, and EPIC (e.g., Chong, 2001). These are becoming interesting variants in their own right.

There are a variety of architectures being developed for modeling social agents, that is, models that interact with a few to a large number of other agents (e.g., Carley, 1996; Yen, Yin, Ioerger, Miller, Xu, & Volz, 2001). Presentations at the CASOS conference (e.g., www.casos.cs.cmu.edu/events/conferences/conference_2004.html) often explain advances in this area. These social architectures tend to have less information processing capabilities, but appropriately more communication capabilities as well as including more instrumentation to record and analyze the behavior of groups from 10 to 1,000.

AMBR is a large scale project to compare cognitive architectures based on how they interact with a common task. Their results may help you choose a cognitive architecture. The AMBR project has provided two large scale simulations and had models written in a variety of architectures (Gluck & Pew, 2001a, 2001b; Pew & Gluck, in preparation). In addition to ACT-R and a modified version of Soar, these comparisons have included COGNET/iGEN (a commercial architecture from CHI Systems), and D-COG (an architecture developed by the US Air Force).

This section has described a range of architectures to consider and noted several reviews that provide comparisons. The reader will have to choose their own cross to bear, according to what they want to model and the resources available to them. Two architectures were used in the cell-phone example (St. Amant et al., 2004) for comparison. One (GOMS) was chosen because it is commonly and easily used. The other (ACT-R) was chosen because it is commonly used, it supports problem solving, is extendable, and it will be able to use the results of several related projects.

3. How to learn about architectures and models

It is generally acknowledged that learning how to create cognitive models is not a simple process. There are materials to help with this process, organized here by presentation media; they could be organized through the stages of data gathering to model building and testing as well.

3.1 General textbooks on simulation and modeling

There are some general textbooks on simulation and modeling that would be helpful. Pew and Mavor (1998) implicitly provides some overview. A book in the Sage methodology series (Tabor & Timpone, 1996) explicitly provides just an overview. Books on mathematical psychology (e.g., Greeno, 1968; Townsend & Ashby, 1983; Wickens, 1982) offer some guidance, but as their approach is based on theories that typically have a closed form or with much different, simpler assumptions, they do not always offer much guidance. None-the-less, these books do teach some basic assumptions and lessons that are not yet in a cognitive modeling book.

3.2 Textbooks and materials on cognitive modeling

There are some textbooks that attempt to summarize cognitive modeling and in some cases attempt to teach it. These are worth examining. Boden (1988) provides an overview of several major approaches, including Newell and Simon's approach as well as connectionist approaches.

Those working with connectionist architectures have several books to choose from (McClelland & Rumelhart, 1988; McLeod, Plunkett, & Rolls, 1998; O'Reilly, 2000). These books introduce modeling as well as an associated software package. Through worked examples they provide jumping off places for other projects and for further work.

Cooper (2002) has recently published a book on the COGENT system. Like the connectionist books, it too provides a set of examples, along with comments about model building.

The van Someren, Barnard, and Sandberg book (1994) comes closest to providing a book on how to create cognitive models. It is too brief, but starts to touch on many of the important topics, such as how to gather verbal protocols, how to compute inter-rater reliability, and the use of task analysis.

Ericsson and Simon's (1993) book has to be included here. It provides the rationale for using verbal protocols as data, as well as the many limitations of verbal protocols. It also includes some practical advice, but not nearly enough. Work continues on understanding how to use other non-verbal protocols such as mouse moves (Baccino & Kennedy, 1995), eye-gaze (Anderson, Bothell, & Douglass, 2004), and demasking (Seifert, 2001).

Work on sequential data analysis needs to be included as well. Modelers interested in the sequential predictions of their models would be well advised to become a student of sequential data representations (Sun & Giles, 1998), sequential data analysis (Gottman & Roy, 1990), and exploratory sequential data analysis (Sanderson & Fisher, 1994). There are useful tools in this area to help with coding and analyzing data (e.g., MacShapa: Sanderson, James, & Seidler, 1989, which has been updated since). Reviews of techniques and tools seem to be done every few years (Fielding & Lee, 1991; Ritter, 1993; Sanderson & Fisher, 1994).

3.3 Exemplar books and monographs

There are several books and monographs that are worth studying because they teach by example fairly well, not because they often or very directly give proscriptive advice. Many people have learned this way. Newell and Simon's (1972) *Human problem solving* is probably the canonical one. While few people have read it cover to cover (I think I know of one person, and I have read about two-thirds), it provides numerous examples worked out in great detail, and teaches the ethos and spirit of the approach. Baxter's (1997) report is also presented in this way, and provides a more current example for Soar. Some Soar and ACT-R theses also provide examples (e.g., Wiesmeyer, 1992), but often do not provide much help for those interested in learning the process.

There are now several edited books on cognitive modeling (Polk & Seifert, 2002; Rosenbloom, Laird, & Newell, 1992). These do not provide a unified treatment, but do provide numerous lessons and further examples. Simon's (1979, 1989) books of collected works in this area (the checkerboard books) have numerous examples, many of which are still worth building on, and all are worth learning.

3.4 Useful articles

Yost and Newell's (1989) article is helpful, as it attempts to explain the process of building a model, but it is tied to a single architecture and should be more widely read. There are other useful papers, but they are short and do not provide the full story. Interested students of modeling will find them helpful (Kieras, 1985; Ritter & Larkin, 1994; Sun & Ling, 1998). There are numerous examples of models and their fit to the data, but they tend not to explain the steps of model creation from the data – the model tends to appear in final form as Athena did from the head of Zeus.

3.5 Programming style and informal mechanisms

Users of architectures have found that in addition to the architecture and knowledge there is another component that needs to be formalized. Because models within the architectures are theories, how they use the mechanisms – in a uniform or ad hoc basis – are part of the theory.

Newell (1990) noted this when he said that there is more in your architecture than you would expect. Learning how to use an architecture and how not to misuse it has to be learned, not just individually, but as a community. Kieras, Wood, and Meyer (1997) referred to it as rules for creating models. Kieras, 2003 provides an update to this. Newell (1990) and the Psychological Soar Tutorial (<http://acs.ist.psu.edu/nottingham/pst/pst-ftp.html>) refer to it as listening to the architecture and using it appropriately. Some in the Soar group have recently started to formalize how to program Soar in a document where the how-to is referred to as dogma (Nuxoll & Laird, 2003). Similar requirements are already apparent as we create the COJACK architecture (Norling & Ritter, in press).

What all of these authors are referring to is a set of conventions that are adhered to when creating or programming the model. Examples of these conventions for Soar include using only one value per attribute in Soar, not putting too much information in a single state, and not creating operators that are overly complex.

Learning this architectural component is difficult because it is not yet formalized (although model compilers offer the promise of doing this, e.g., St. Amant & Ritter, in press). It is perhaps this type of knowledge that is missing when modelers have difficulties or give up. The old rule of thumb was that you had to visit an established site to absorb this information, and while numerous steps have been taken in recent years to reduce this requirement, such visits are still a good idea.

3.6 Conferences, tutorials, and online materials

Conferences (e.g., the Cognitive Science Conference, International Conference on Cognitive Modeling) and workshops (e.g., the Soar, ACT-R, and Cogent Workshops) offer opportunities to learn current programming (modeling) paradigms and to meet other modelers. Such paradigms have been actively debated in panels at Soar and ACT-R workshops.

There are also more formal places to learn. Tutorials are now often offered at the relevant conferences, and these can serve as useful introductions. The ACT-R summer School and the German Cognitive Autumn School (Herbstschule Kognitionswis-

senschaft), while only available occasionally, offer excellent opportunities to get started and to learn more.

Finally, there are now online materials. On their main web sites both ACT-R (<http://act.psy.cmu.edu/>) and Soar (<http://sitemaker.umich.edu/soar>) now have online tutorials (you print them and do the exercises), and they both have Frequently asked questions lists (<http://acs.ist.psu.edu/act-r-faq>, <http://acs.ist.psu.edu/soar-faq>). Other architectures are likely to have the same, if not now, they will have them soon as the standard of support rises for new and existing users. These tutorials and FAQs also raise issues that most architectures will wish to address, so they should provide value even for users of other architectures.

3.7 Psychology and computer science materials

There are two further areas important for the success of modeling to keep in mind: psychology (the data to be modeled), and computer science (the tools used in modeling). If you are a computer scientist coming to cognitive modeling, you will need an overview of the information processing view of psychology. Anderson's books on psychology (1996) and on learning and memory (1995) are excellent introductions and good overviews. (There are others as well.) If you are interested in more specific areas, textbooks in those areas will also be helpful.

If you wish to have access to a wide range of data useful for modeling, either to help build an architecture or else to provide additional data to extend the coverage of the model, engineering psychology can provide this. Wickens' text books (Wickens, Gordon, & Liu, 1998; Wickens & Hollands, 2000) provide an overview, and the Engineering Data Compendium (Boff, Kaufman, & Thomas, 1986; Boff & Lincoln, 1988) provides a detailed view that is sometimes helpful.

It was first noted by Kieras (1985) that it is very useful, perhaps even essential, that modelers know the language underlying their architecture to assist in modeling. After the model is built, additional apparatus will have to be built to include running the model multiple times (if it is stochastic), to explore variants of the model, to run the model on a variety of tasks, and to provide the model access to a task simulation. For current modelers, this can mean studying Lisp, Tcl/Tk, and Java. Online resources and summaries of learning materials for these languages can be found, for example, in the frequently asked questions lists for the architecture (e.g., <http://acs.ist.psu.edu/act-r-faq>, <http://acs.ist.psu.edu/soar-faq>). Psychologists who just set themselves the task to learn ACT-R and not Lisp will run into difficulties, and will either learn Lisp or quit.

4. How to test your model²

Modelling is always a purpose driven act. Thus every model has a purpose (or set of purposes). Testing and validation has to be done with the purpose of the model in mind. For science, the role of the proposed mechanisms to account for behavior is pretty common. Thus testing is probably the better label for this step. For engineering and design, the usefulness and usability of the model are being considered, so validation is probably a better label for this step.

² An earlier version of this section was presented at a Symposium on Model Fitting and Parameter Estimation at the ACT-R Workshop, 2003.

How to test and validate your model is a problem that has vexed many researchers. I have met several people and I have spent time myself looking for a statistic to prove cognitive models. This is a fruitless search that I suspect is repeated far too often. There is an approach to testing models that I think is productive. Campbell and Bolton (in preparation) provide a longer explanation that will be useful to those particularly interested in this topic. After the theoretical background is provided, some practical suggestions are provided.

4.1 Theoretical background

Grant's (1962) paper on the strategy and tactics of investigating models argued that there were two important aspects for testing a model, that (a) the model was worth taking seriously, and (b) you could see where the model was wrong so that you could improve it. This is consistent with Newell's (1990, p. 507) view of how to develop unified theories of cognition (UTC): what is the current bar (standards), does this theory (or model) raise it, and what are the further regularities to be included in the future? I like this approach as it lets me make progress, or at least be happy. I have seen others trying to prove their model, and they are not and cannot be happy because proving your model is equivalent to accepting the null hypothesis (see any elementary statistics book for a description of the dangers of that).

Taking a model seriously depends on what other models are available and what you want to do with it. What is the current best model? If you look at current theory/data comparisons of task performance models (e.g., ACT-R), the models can typically match a single type of data or a few kinds of data on a single task. For example, the data compared with the model will include one or a few of reaction time means, the sequence of task actions, groupings of task actions into strategies, error rates and types, and trends and variance in all of these. Few models have had their predictions compared to all of these aspects of data. Fewer yet have been compared to data from multiple tasks. Gobet and Ritter (2000) describe this approach; Lovett, Daily, and Reder (2000) independently have provided an example.

Sometimes in a new area of modeling it will be enough report the performance of model, that is, that it can do the task (currently models of teamwork and emotions seem to use this approach). More advanced models may report the correlation between the model's predictions and data, which Simon and Grant both recommend. Correlations currently appear to be a good standard, and they often lead me to take models seriously.

Summarizing the match across these sets of regularities can be done in multiple ways. For example, John (1996) has used a type of bar chart across a set of different types of behavior being matched. Further details of this approach are presented in Newell's (1990) book, and briefly expanded in Ritter (1993) as criterion-based cognitive modeling. This approach, of criterion-based cognitive modeling, is a way to protect models because it defines the range and performance expected from a given model. Schunn and Wallach, (2001) also provide many useful suggestions.

With multiple types of data with multiple values and multiple displays, how can one compare theories? I currently think that Grant's question, "is a theory worth taking seriously?", can be seen at least partly as a social process. Theories will correspond to the data on a number of dimensions. Reducing their fit to a single number for comparison to choose the best model is likely to be difficult when complex models

or complex data sets are considered. Some models are admirable because they do not touch the simulation and offer new worlds to models (but have an unreported or poor fit). Another model may be interesting because it opens up new areas of data to be included into ACT-R or Soar. A third model may be interesting because it shows how to use a genetic algorithm to fairly test a wide range of ways to adjust models to fit a dataset (e.g., what develops in children?, Tor & Ritter, in press).

In each case, the judgment of "is this model interesting" is based on other models, how well the model fits the data, how applicable the theory, how easy the theory is to use, and a host of other factors. Estimates of future applicability is also important. "Science, like politics, is the art of the possible", said Newell, and I rather strongly agree. That means that I take models that I can download and include with my model much more seriously than those that I cannot inspect or that cost \$1,000. (Something interesting is going on here, because except for computer proofs in mathematics, rarely in science are theories cast so strongly as programs, it seems; and estimates of future usability are likely to be inaccurate.)

A recent set of comments (Roberts & Pashler, 2000, 2002; Rodgers & Rowe, 2002) argue that a reader needs to know more about the model predictions to data comparison than just the fit. They argue that readers need to know what kind of data that the theory cannot fit, the variability of the data, and the likelihood of fitting data. Roberts and Pashler's stance appears to be consistent basically with Grant's two step process, but they ask for more details. The details they ask for appear to me to be more relevant for simple models covering well trod but narrow ground rather than broad, approximate theories that current cognitive models often look like. Roberts and Pashler do request a standard that is worth striving for, but they also appear to be overly harsh. Newell (1990) argued for allowing models time to develop (citing Hebb, "don't bit my finger look where I'm pointing"), and to allow them to have success in multiple ways. A model that performed a new type of learning or problem solving would be inappropriately rejected by Robert's and Pashler's criteria.

Roberts and Pashler prefer theories that predict surprising data. I also find much of psychology data surprising, which they do not, and thus I think predicting this data is worthwhile. I know of several theories that do not predict smooth curves and the data matches these non-smooth curves. Finally, I believe that task performance is much more important than fits to data because task performance is a prerequisite for generating behavior and thus for more autonomous predictions and applications.

4.2 Practical recommendations

So, (a) I recommend that you tell us about your model's predictions, what the data look like, and how the model's predictions correspond to the data in detail, enough so that we can see that the model is worth taking seriously. Because the judgement is based on other alternative models (if any), there is no a priori quality required. There is not a value of r^2 that must be satisfied, although the r^2 of competing models are a good yardstick. You might also note a model's other virtues, such as ease of use, and consistency but not yet correlation with large swaths of behavior.

There are also reasons to dismiss a model. If the model would fit any data, then it is not worth taking seriously (but only if such data already exist, hypothesized data need not apply). If I cannot understand the model; if it is a hack; or if I believe it will

not generalize to other data; and I would add now, if it is not part of a UTC, I am less interested.

I also recommend that you (b) Note where the model can be improved. This does not mean including in your paper a laundry list of data that your model does not yet cover because you ran out of time. If you are a reviewer, it certainly does not mean providing 40 pages of comments of places where the authors could extend their model.³

Therefore, include in your reports just enough detail on your model's limitations for readers to know that you know where the remaining problems are, and to indicate that you know enough to improve your model, but not to apologize for tasks it cannot yet do. Thus, for the cell-phone model (St. Amant et al., 2004) both processes were done. The model was tested to find out how it compared to existing scientific models in the area. The models are shown to predict the data using a table and a figure to show the correlation. We also noted where it could be improved in the near term. The model was also validated, in that the results showed that the models were usable and made not only accurate but useful predictions. The effects of redesign could not only be predicted, but indicated that redesign could be very beneficial.

5. How to choose a problem to work on

With an architecture in hand you might then wish to choose an interesting and timely problem to work on. In reality, the problem in front of you is likely to have driven you to desire to work with a cognitive architecture to start with, or may have arisen from your use of a particular cognitive architecture. Thus, this section is not usually the last section in your journey, but it is include it at the end as a summary of what I believe are some of the current areas of interest when using cognitive architectures. Other interesting problems and reviews exist in this area, and this section does not describe, of course, all possible problems.

Newell in his *Desires and Diversions* talk (Newell, 1991)⁴ emphasized the need to work on the most important problem. While I have colleagues who disagree with me on the necessity of this directed approach, what progress I feel seems to come in the same way as Newell described it, as returning to the same problems and staying focused on a line of research as much as possible; working on an important problem; and working on a problem where you have some comparative advantage due to education or access to resources or affinity. Thus, in introducing these areas as interesting, you may note that along with a variety of colleagues I am working on some aspect of several of these problems.

³ These two suggestions are consonant with two comments taken from a document a colleague recently was kind enough to share with me, *Levy's Ten Laws of the Disillusionment of the True Liberal*, findable online with a search engine. That is, Law 4b: Good intentions are far more difficult to cope with than malicious behavior; and Law 8: No amount of genius can overcome a preoccupation with detail.

⁴ This talk is available online. The recommended path is to go to <http://www.ul.cs.cmu.edu/> (Universal Library); click on Multimedia and Lectures, click on Distinguished lectures, click on 1991, and then click on the link to the talk. If you need a hard link, this path, which varies based on machine type and operating system, sometimes resolves to: <http://doi.library.cmu.edu/10.1184/LOCAL/4205> . The slides are available through <http://diva.library.cmu.edu/Newell/>

Pew and Mavor (1998) in the course of their book describe worthwhile projects at the end of each applicable chapter. The problems they note are interesting problems and worth working on. Their descriptions tend to refer to whole areas and to be the size of a multi-year research proposal that might include multiple investigators

Ritter et al.'s (2003) State of the Art report provides a listing of about 23 smaller projects. As a group, these projects are more approachable as PhD, MSc, or class projects in advanced AI, simulation, or modeling classes. These projects are grouped into three main categories, that of extending the coverage of architectures and models, of improving their integration with tasks, and making the models and architectures more usable.

5.1 Projects extending architectures and models

The first project area is to more accurately model human behavior. To highlight just a few of the projects there, one project suggests that including errors in performance will be important. This is of interest to interface designers in safety-critical systems (e.g., Freed & Remington, 1998). Several projects consider learning. The wide varieties of change that learning entails is a large, broad area for work and will be for a long time.

Another interesting project area is including models of emotions, changes of motivation, and changes within and across individuals. Examples of this work are available in my research group (Belavkin & Ritter, 2003; Norling & Ritter, in press; Ritter, Avraamides, & Council, 2002) and elsewhere (Gratch, in press; Hudlicka & McNeese, 2002; Silverman, Cornwell, & O'Brien, in press).

5.2 Projects improving the integration of architectures with tasks

The second project area is integrating models with other systems, broadly defined. In particular, the most interesting problem to me is providing models with access to interfaces in ways that approximate the richness of human perceptual-motor capabilities as well as including the limitations of human capabilities. Providing models access to tasks has been a constant problem for modelers, of how to provide their models access to the task of interest or an interesting task where the phenomenon of interest can be studied. The field has started with providing the task in the modeling language. When that approach became unwieldy, providing the models access to raw information through sockets in a complete domain. There are good reasons to use a micro-world, a simplified simulation of a larger task (Gray, 2002). These approaches have not been entirely satisfactory, and I have seen many projects flounder on this "uninteresting" technical subtask.

In other sciences instrumentation and essential technology support that caused researchers to fail would make the failure point an interesting problem to those disciplines. Thus I think interaction is an interesting problem for cognitive modeling. In the last few years we have been working to create simulated eyes and hands (Lonsdale & Ritter, 2000; Norling & Ritter, 2001; Ritter, Baxter, Jones, & Young, 2000). In particular, we have been working with St. Amant and his students to create a simulated eye and hand that do not need to instrument an interface to interact with it (St. Amant, Horton, & Ritter, 2004; St. Amant & Riedl, 2001).

5.3 Making the models and architectures more usable

The third and final project area is to improve the usability of the resulting models. If the models are too difficult to build or too difficult to understand, use, or apply, then they will not be used. This is an interesting human-computer interaction design task, of creating and explaining intelligent behavior. It is similar in many ways to expert system development, except the systems are not only required to be intelligent, but intelligent like humans.

We have started to gather descriptions of what users want (Councill, Haynes, & Ritter, 2003), and are working on higher level programming languages (St. Amant & Ritter, in press) and displays (Tor, Ritter, Haynes, & Cohen, 2004). Others are working on this problem as well (e.g., Crossman, Jones, Lebiere, & Wray, 2004; also see the upcoming AAAI 2004 Workshop on Intelligent Agent Architectures: Combining the Strengths of Software Engineering and Cognitive Systems).

5.4 Concluding remarks

It is an exciting time for creating and using cognitive models. The technology continues to mature and the science that can be addressed continues to provide interesting problems. We are a long way from being able to routinely create and apply cognitive models in the way that ANOVAs and regressions can be, but the path forward continues to seem clearer and broader with the passage of time. This article may help you on this path, but like nearly all science, progress will be faster with a mentor and being part of a community.

6. Acknowledgements

As this article is about teaching and learning, I would like to take this opportunity to thank my teachers, from the early science ones, to the one who took me aside to teach me logarithms after school, to Newell and Simon who taught by example. Many of my colleagues have taught me in this area. Discussions with Richard Young have been particularly helpful. My students have also had to teach me how to teach this material; I hope this article helps them and makes their learning easier. Rich Carlson, Martin C. Kindsmueller, Emma Norling, Bill Stevenson, and two anonymous reviewers have provided useful comments. Cindy Carroll and the CMU Library's Information Technology staff who created the Newell video reference. Preparation of this report was partially supported ONR, grants N00014-03-1-0248 and N00014-02-1-0021, and partially funded by the Director of Technology Development, Ministry of Defence, Metropole Building, Northumberland Ave, London WC2N 5BP and was carried out under the terms of Contract No RT/COM/3/006.

7. References

- Anderson, J. R. (1995). *Learning and memory*. New York, NY: John Wiley and Sons.
- Anderson, J. R. (1996). *Cognitive psychology and its implications* (3rd ed.). New York, NY: W. H. Freeman.

- Anderson, J. R., Bothell, D., & Douglass, S. (2004). Eye movements do not reflect retrieval processes. *Psychological Science*, 15(4), 225-231.
- Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum.
- Baccino, T., & Kennedy, A. (1995). MICELAB: Spatial processing of mouse movement in Turbo-Pascal. *Behavior Research Methods, Instruments, & Computers*, 27(1), 76-78.
- Badler, N. I., Erignac, C. A., & Liu, Y. (2002). Virtual humans for validating maintenance procedures. *Communications of the ACM*, 45(7), 56-63.
- Baxter, G. D. (1997). *From soup to nuts: Developing a Soar cognitive model of the electronic warfare officer's task*. Working paper No. WP/R3BAIA005/014. Cognitive Modelling unit. Nottingham, UK: Psychology Department, University of Nottingham. <http://acs.ist.psu.edu/nottingham/papers/baxter97.pdf> [link checked 4 June 2004].
- Belavkin, R. V., & Ritter, F. E. (2003). The use of entropy for analysis and control of cognitive models. In F. Detje, D. Dörner, & H. Schaub (Eds.), *Proceedings of the Fifth International Conference on Cognitive Modeling*. 21-26. Bamberg, Germany: Universitäts-Verlag Bamberg.
- Boden, M. A. (1988). *Computer models of mind*. Cambridge, UK: Cambridge U. Press.
- Boff, K. R., Kaufman, L., & Thomas, J. P. (Eds.). (1986). *Handbook of perception and human performance*. New York: John Wiley & Sons.
- Boff, K. R., & Lincoln, J. E. (Eds.). (1988). *Engineering data compendium*. Wright-Patterson Air Force Base, OH: Harry G. Armstrong Aerospace Medical Research Laboratory.
- Burton, R. (1998). Validating and docking: An overview, summary and challenge. In M. Prietula, K. Carley, & L. Gasser (Eds.), *Dynamics of organizations*. 215-228. Menlo Park, CA: AAAI.
- Busetta, P., Rönquist, R., Hodgson, A., & Lucas, A. (1999). JACK intelligent agents - Components for intelligent agents in JAVA. *AgentLink News Letter*, 2(Jan.), <http://www.agent-software.com> [link checked 4 June 2004].
- Byrne, M. D. (2001). ACT-R/PM and menu selection: Applying a cognitive architecture to HCI. *International Journal of Human-Computer Studies*, 55, 41-84.
- Byrne, M. D., Wood, S. D., Sukaviriya, P., Foley, J. D., & Kieras, D. E. (1994). Automating interface evaluation. In *Proceedings of the CHI'94 Conference on Human Factors in Computer Systems*. 232-237. New York, NY: ACM.
- Campbell, G. E., & Bolton, A. E. (in preparation). Implications for model assessment and validation. In R. W. Pew & K. A. Gluck (Eds.), *Modeling human behavior with integrated cognitive architectures: Comparison, evaluation, and validation*. Mahwah, NJ: Lawrence Erlbaum.

- Card, S., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum.
- Carley, K. M. (1996). A comparison of artificial and human organizations. *Journal of Economic Behavior & Organization*, 31, 175-191.
- Chong, R. S. (2001). Low-level behavioral modeling and the HLA: An EPIC-Soar model of an enroute air-traffic control task. In *Proceedings of the 10th Computer Generated Forces and Behavioral Representation Conference*. 27-35. 10TH-CGF-072. Orlando, FL: Division of Continuing Education, University of Central Florida. <http://www.sisostds.org/cgf-br/10th/> [link checked 4 June 2004].
- Cooper, R. P. (2002). *Modelling high-level cognitive processes*. Mahwah, NJ: Lawrence Erlbaum.
- Cooper, R., & Fox, J. (1998). COGENT: A visual design environment for cognitive modelling. *Behavior Research Methods, Instruments, and Computers*, 30, 553-564.
- Crossman, J., Wray, R., Jones, R., & Lebiere, C. (2004). A high level symbolic representation for behavior modeling. In *Proceedings of the 13th Conference on Behavior Representation in Modeling and Simulation*. 04-BRIMS-051. <http://www.sisostds.org/cgf-br/04Brims/> [link checked 4 June 2004]. Orlando, FL: U. of Central Florida.
- Councill, I. G., Haynes, S. R., & Ritter, F. E. (2003). Explaining Soar: Analysis of existing tools and user information requirements. In F. Detje, D. Dörner, & H. Schaub (Eds.), *Proceedings of the Fifth International Conference on Cognitive Modeling*. 63-68. Bamberg, Germany: Universitäts-Verlag Bamberg.
- Detje, F. (2000). Comparison of the PSI-theory with human behaviour in a complex task. In N. Taatgen & J. Aasman (Eds.), *Proceedings of the Third International Conference on Cognitive Modelling*. 86-93. Veenendaal, The Netherlands: Universal Press.
- Deutsch, S. E. (1998). Interdisciplinary foundations for multiple-task human performance modeling in OMAR. In *Proceedings of the Twentieth Annual Meeting of the Cognitive Science Society*. Mahwah, NJ: Lawrence Erlbaum.
- Dörner, D. (2003). The mathematics of emotions. In F. Detje, D. Dörner, & H. Schaub (Eds.), *Proceedings of the Fifth International Conference on Cognitive Modelling*. 75-80. Bamberg, Germany: Universitäts-Verlag Bamberg.
- Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data*. Cambridge, MA: The MIT Press.
- Fielding, N. G., & Lee, R. M. (Eds.). (1991). *Using computers in qualitative research*. London & Beverly Hills, CA: Sage.
- Freed, M., & Remington, R. (1998). A conceptual framework for predicting error in complex human-machine environments. In M. A. Gernsbacher & S. J. Derry (Eds.), *Proceedings of the 20th Annual Conference of the Cognitive Science Society*. 356-361. Mahwah, NJ: Lawrence Erlbaum.

- Freed, M., & Remington, R. (2000). Making human-machine system simulation a practical engineering tool: An APEX overview. In N. Taatgen & J. Aasman (Eds.), *Proceedings of the 3rd International Conference on Cognitive Modelling*. 110-117. Veenendaal, The Netherlands: Universal Press.
- Gluck, K. A., & Pew, R. W. (2001a). Lessons learned and future directions for the AMBR model comparison project. In *Proceedings of the 10th Computer Generated Forces and Behavioral Representation Conference*. 10TH-CGF-067. 113-121. Orlando, FL: Division of Continuing Education, University of Central Florida. <http://www.sisostds.org/cgf-br/10th/> [link checked 4 June 2004].
- Gluck, K. A., & Pew, R. W. (2001b). Overview of the agent-based modeling and behavior representation (AMBR) model comparison project. In *Proceedings of the 10th Computer Generated Forces and Behavioral Representation Conference*. 10TH-CGF-066. 3-6. Orlando, FL: Division of Continuing Education, University of Central Florida. <http://www.sisostds.org/cgf-br/10th/> [link checked 4 June 2004].
- Gobet, F., & Ritter, F. E. (2000). Individual Data Analysis and Unified Theories of Cognition: A methodological proposal. In N. Taatgen & J. Aasman (Eds.), *Proceedings of the 3rd International Conference on Cognitive Modelling*. 150-157. Veenendaal, The Netherlands: Universal Press.
- Gottman, J. M., & Roy, A. K. (1990). *Sequential analysis: A guide for behavioral researchers*. Cambridge, UK: Cambridge University Press.
- Grant, D. A. (1962). Testing the null hypothesis and the strategy and tactics of investigating theoretical models. *Psychological Review*, 69(1), 54-61.
- Gratch, J. (in press). A domain-independent framework for modeling emotion. *Journal of Cognitive Systems Research*.
- Gray, W. D. (2002). Simulated task environments: The role of high-fidelity simulations, scaled worlds, synthetic environments, and microworlds in basic and applied cognitive research. *Cognitive Science Quarterly*, 2(2), 205-227.
- Gray, W. D., & Boehm-Davis, D. A. (2000). Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology: Applied*, 6(4), 322-335.
- Gray, W. D., John, B. E., & Atwood, M. E. (1993). Project Ernestine: Validating a GOMS analysis for predicting and explaining real-world task performance. *Human-Computer Interaction*, 8(3), 237-309.
- Greeno, J. G. (1968). *Elementary theoretical psychology*. Reading, MA: Addison-Wesley.
- Hudlicka, E., & McNeese, M. D. (2002). User affective and belief states: Assessment and user interface adaptation. *Journal of User Modeling and User Adapted Interaction*, 12, 1-47.
- John, B. E. (1996). TYPIST: A theory of performance in skilled typing. *Human Computer Interaction*, 11(4), 321-355.

- 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), 320-351.
- Johnson, T. R. (1997). Control in ACT-R and Soar. In M. Shafto & P. Langley (Eds.), *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*. 343-348. Hillsdale, NJ: Lawrence Erlbaum.
- Johnson, T. R. (1998). A comparison of ACT-R and Soar. In U. Schmid, J. Krems, & F. Wysotzki (Eds.), *Mind modeling - A cognitive science approach to reasoning, learning and discovery*. 17-38. Lengerich, Germany: Pabst Scientific Publishing.
- Kieras, D. E. (1985). The why, when, and how of cognitive simulation. *Behavior Research Methods, Instrumentation, and Computers*, 17, 279-285.
- Kieras, D. (2003). Model-based evaluation. In J. Jacko & A. Sears (Eds.), *Handbook for human-computer interaction*. 1139-1151. Mahwah, NJ: Lawrence Erlbaum.
- Kieras, D. E., Wood, S. D., Abotel, K., & Hornof, A. (1995). GLEAN: A computer-based tool for rapid GOMS model usability evaluation of user interface designs. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'95)*. 91-100. New York, NY: ACM.
- Kieras, D. E., Wood, S. D., & Meyer, D. E. (1997). Predictive engineering models based on the EPIC architecture for a multimodal high-performance human-computer interaction task. *Transactions on Computer-Human Interaction*, 4(3), 230-275.
- Langley, P. (1996). An abstract computational model of learning selective sensing skills. In *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society*. 385-390. Hillsdale, NJ: Lawrence Erlbaum.
- Laughery, K. R., & Corker, K. M. (1997). Computer modeling and simulation of human/system performance. In G. Salvendy (Ed.), *Handbook of human factors*. New York, NY: John Wiley and Sons.
- Lehman, J. F., Laird, J. E., & Rosenbloom, P. S. (1996). A gentle introduction to Soar, an architecture for human cognition. In S. Sternberg & D. Scarborough (Eds.), *Invitation to cognitive science, vol. 4*. Cambridge, MA: MIT Press.
- Lonsdale, P. R., & Ritter, F. E. (2000). Soar/Tcl-PM: Extending the Soar architecture to include a widely applicable virtual eye and hand. In N. Taatgen & J. Aasman (Eds.), *Proceedings of the 3rd International Conference on Cognitive Modeling*. 202-209. Veenendaal, The Netherlands: Universal Press.
- Lovett, M. C., Daily, L. Z., & Reder, L. M. (2000). A source activation theory of working memory: Cross-task prediction of performance in ACT-R. *Journal of Cognitive Systems Research*, 1, 99-118.
- Matessa, M. (in press). Anticipatory eye movements in interleaving templates of human behavior. In *Proceedings of the International Conference on Cognitive Modeling*. Mahwah, NJ: Lawrence Erlbaum.

- McClelland, J. L., & Rumelhart, D. E. (1988). *Explorations in parallel distributed processing: A handbook of models, programs, and exercises*. Cambridge, MA: MIT Press.
- McLeod, P., Plunkett, K., & Rolls, E. T. (1998). *Introduction to connectionist modeling of cognitive processes*. Oxford, UK: Oxford University Press.
- Naveh, I., & Sun, R. (in press). Simulating a simple case of organizational decision making. In R. Sun (Ed.), *Cognition and multi-agent interaction: From cognitive modeling to social simulation*. New York, NY: Cambridge U. Press.
- Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.
- Newell, A. (1991). *Desires and diversions: Carnegie-Mellon University School of Computer Science Distinguished Lecture*. Palo Alto, CA: University Communications. 64 min. video.
- Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.
- Nichols, S., & Ritter, F. E. (1995). A theoretically motivated tool for automatically generating command aliases. In *Proceedings of the CHI'95 Conference on Human Factors in Computer Systems*. 393-400. New York, NY: ACM.
- Norling, E., & Ritter, F. E. (2001). Embodying the JACK agent architecture. In M. Stumptner, D. Corbett, & M. Brooks (Eds.), *AI 2001: Advances in Artificial Intelligence. Proceedings of the 14th Australian Joint Conference on Artificial Intelligence*. 368-366. Berlin: Springer.
- Norling, E., & Ritter, F. E. (in press). A parameter set to support psychologically plausible variability in agent-based human modelling. To appear in *Proceedings of The Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS04)*.
- Nuxoll, A., & Laird, J. (2003). Soar Design Dogma. Version 0.6, <http://ai.eecs.umich.edu/soar/sitemaker/docs/misc/dogma.pdf> [link checked 4 June 2004].
- O'Reilly, R. C., & Munakata, Y. (2000). *Computational explorations in cognitive neuroscience: Understanding the mind by simulating the brain*. Cambridge, MA: The MIT Press.
- Pew, R. W., & Gluck, K. A. (Eds.). (in preparation). *Modeling human behavior with integrated cognitive architectures: Comparison, evaluation, and validation*. Mahwah, NJ: Lawrence Erlbaum.
- Pew, R. W., & Mavor, A. S. (Eds.). (1998). *Modeling human and organizational behavior: Application to military simulations*. Washington, DC: National Academy Press. <http://books.nap.edu/catalog/6173.html> [link checked 4 June 2004].
- Polk, T. A., & Seifert, C. M. (2002). *Cognitive modeling*. Cambridge, MA: MIT Press.

- Ritter, F. E. (1993). *TBPA: A methodology and software environment for testing process models' sequential predictions with protocols*. Technical Report No. CMU-CS-93-101. Pittsburgh, PA: School of Computer Science, Carnegie-Mellon University. <http://reports-archive.adm.cs.cmu.edu/anon/1994/CMU-CS-94-102.ps> [link checked 4 June 2004].
- Ritter, F. E., Avraamides, M., & Councill, I. G. (2002). An approach for accurately modeling the effects of behavior moderators. In *Proceedings of the 11th Computer Generated Forces Conference*. 29-40, 02-CGF-002. Orlando, FL: U. of Central Florida.
- Ritter, F. E., Baxter, G. D., Jones, G., & Young, R. M. (2000). Supporting cognitive models as users. *ACM Transactions on Computer-Human Interaction*, 7(2), 141-173.
- Ritter, F. E., Jones, R. M., & Baxter, G. D. (1998). Reusable models and graphical interfaces: Realising the potential of a unified theory of cognition. In U. Schmid, J. Krems, & F. Wysotzki (Eds.), *Mind modeling - A cognitive science approach to reasoning, learning and discovery*. 83-109. Lengerich, Germany: Pabst Scientific Publishing.
- Ritter, F. E., & Larkin, J. H. (1994). Using process models to summarize sequences of human actions. *Human-Computer Interaction*, 9(3), 345-383.
- Ritter, F. E., Shadbolt, N. R., Elliman, D., Young, R., Gobet, F., & Baxter, G. D. (2003). *Techniques for modeling human performance in synthetic environments: A supplementary review*. Wright-Patterson Air Force Base, OH: Human Systems Information Analysis Center (HSIAC), formerly known as the Crew System Ergonomics Information Analysis Center (CSERIAC). <http://iac.dtic.mil/hsiac/S-docs/SOAR-Jun03.pdf> [link checked 4 June 2004].
- Ritter, F. E., Van Rooy, D., & St. Amant, R. (2002). A user modeling design tool based on a cognitive architecture for comparing interfaces. In C. Kolski & J. Vanderdonck (Eds.), *Computer-Aided Design of User Interfaces III, Proceedings of the 4th International Conference on Computer-Aided Design of User Interfaces CADUI'2002*. 111-118. Dordrecht, NL: Kluwer Academic Publisher.
- Roberts, S., & Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. *Psychological Review*, 107(2), 358-367.
- Roberts, S., & Pashler, H. (2002). Reply to Rodgers and Rowe (2002). *Psychological Review*, 109(3), 605-607.
- Rodgers, J. L., & Rowe, D. C. (2002). Theory development should begin (but not end) with good empirical fits: A comment on Roberts and Pashler (2000). *Psychological Review*, 109(3), 599-604.
- Rosenbloom, P. S., Laird, J. E., & Newell, A. (1992). *The Soar Papers: Research on integrated intelligence*. Cambridge, MA: MIT Press.
- Sanderson, P. M., & Fisher, C. A. (1994). Exploratory sequential data analysis: Foundations. *Human-Computer Interaction*, 9(3&4), 251-317.

- Sanderson, P. M., James, J. M., & Seidler, K. S. (1989). SHAPA: an interactive software environment for protocol analysis. *Ergonomics*, 32(11), 1271-1302.
- Schunn, C. D., & Wallach, D. (2001). Evaluating goodness-of-fit in comparison of models to data. Online manuscript. Retrieved 1 April 2004 from www.lrdc.pitt.edu/schunn/gof/GOF.doc.
- Seifert, K. (2001). Comparison of methods for studying air traffic controller's information strategies. *MMI-Interaktiv*, 5, Juli 2001, http://www.useworld.net/servlet/handlearticle?obj_id=573&cat_id=40&vie=true [link checked 4 June 2004].
- SIGArt (1991). Special section on integrated cognitive architectures. *Sigart Bulletin*, 2(4).
- Silverman, B. G., Cornwell, J., & O'Brien, K. (in press). Human performance simulation. In J. W. Ness, D. R. Ritzer, & V. Tepe (Eds.), *The science and simulation of human performance*. Amsterdam: Elsevier. <http://www.seas.upenn.edu/~barryg/Ch9-final.pdf> [link checked 4 June 2004].
- Simon, H. A. (1979). *Models of thought*. New Haven, CT: Yale University Press.
- Simon, H. A. (1989). *Models of thought, Volume II*. New Haven, CT: Yale University Press.
- St. Amant, R. (2000). Interface agents as surrogate users. *intelligence*, 11(Summer 2000), 29-38.
- St. Amant, R., Horton, T. E., & Ritter, F. E. (2004). Model-based evaluation of cell phone menu interaction. In *Proceedings of the CHI'04 Conference on Human Factors in Computer Systems*. 343-350. New York, NY: ACM.
- St. Amant, R., & Riedl, M. O. (2001). A perception/action substrate for cognitive modeling in HCI. *International Journal of Human-Computer Studies*, 55, 15-39.
- St. Amant, R., & Ritter, F. E. (in press). Automated GOMS to ACT-R model generation. In *submitted to the International Conference on Cognitive Modeling*. Mahwah, NJ: Lawrence Erlbaum.
- Sun, R., & Giles, C. L. (1998). *Sequence learning*. Berlin, Germany: Springer.
- Sun, R., & Ling, C. X. (1998). Computational cognitive modeling, the source of power, and other related issues. *AI Magazine*, 19(2), 113-120.
- Sun, R., Merrill, E., & Peterson, T. (1998). Skill learning using a bottom-up hybrid model. In F. E. Ritter & R. M. Young (Eds.), *Proceedings of the 2nd European Conference on Cognitive Modelling*. 23-29. Thrumpton, Nottingham: Nottingham University Press.
- Tabor, C. S., & Timpone, R. J. (1996). *Computational modelling*. Thousand Oaks, CA: Sage.

- Tor, K., & Ritter, F. E. (in press). Using a genetic algorithm to optimize the fit of cognitive models. In *Proceedings of the Sixth International Conference on Cognitive Modeling*. Mahwah, NJ: Lawrence Erlbaum.
- Tor, K., Ritter, F. E., Haynes, S. R., & Cohen, M. A. (2004). CaDaDis: A tool for displaying the behavior of cognitive models and agents. In *Proceedings of the 13th Conference on Behavior Representation in Modeling and Simulation*. 04-BRIMS-032. 192-200. Orlando, FL: U. of Central Florida. <http://www.sisostds.org/cgf-br/04Brims> [link checked 4 June 2004].
- Townsend, J. T., & Ashby, F. G. (1983). *Stochastic modeling of elementary psychological processes*. Cambridge, UK: Cambridge University Press.
- van Someren, M. W., Barnard, Y. F., & Sandberg, J. A. C. (1994). *The Think Aloud Method: A practical guide to modelling cognitive processes*. London and San Diego: Academic Press.
- Wickens, C. D., Gordon, S. E., & Liu, Y. (1998). *An introduction to human factors engineering* (3rd ed.). Addison-Wesley: New York, NY.
- Wickens, C. D., & Hollands, J. G. (2000). *Engineering psychology and human performance* (3rd ed.). Prentice-Hall: Upper Saddle River, NJ.
- Wickens, T. D. (1982). *Models for behavior: Stochastic processes in psychology*. San Francisco: Freeman.
- Wiesmeyer, M. D. (1992). *An operator-based model of human covert visual attention*. Ph.D. thesis, U. of Michigan.
- Yen, J., Yin, J., Ioerger, T. R., Miller, M. S., Xu, D., & Volz, R. A. (2001). CAST: Collaborative agents for simulating teamwork. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)*. 1135-1142. Los Altos, CA: Morgan Kaufmann.
- Yost, G. R., & Newell, A. (1989). A problem space approach to expert system specification. In *Eleventh International Joint Conference on Artificial Intelligence*. 621-627.
- Zachary, W., Jones, R. M., & Taylor, G. (2002). How to communicate to users what is inside a cognitive model. In *Proceedings of the 11th Computer Generated Forces Conference*. 375-382, 02-CGF-114. Orlando, FL: U. of Central Florida.
- Also see <http://acs.ist.psu.edu/papers/> for online copies of papers by Ritter.*

Extending ACT-R for modeling dynamics and timing for operating human-machine systems

SANDRO LEUCHTER & LEON URBAS

*MoDyS Research Group, Center of Human-Machine Systems,
Technische Universität Berlin*

Keywords: cognitive modelling, dynamics, human-machine-systems, cognitive architecture, ACT-R

1. Introduction

An important current trend for the ACT-R community is to apply cognitive modeling to real world problems like HCI (Byrne 2001) or human machine interaction (Gluck 2002). Important developments of the architecture that are necessary to do so are PM for integration to the task environment and new ways of goal management. An area that still needs attention is timing for operating human-machine systems. With this contribution we want to promote the application of cognitive architectures for engineering applications in dynamic human-machine systems (HMS). In this section the notion of HMS and their dynamics are defined and the objective of this work is presented in detail.

1.1 Human-Machine Systems

Many machines (i.e. technical systems) are used by humans. The term human-machine system denotes not only systems in which at least one human operates a technical system, but emphasizes the interaction between human and machine. Typically the technical system is fairly complex and shows a continuous dynamic behaviour that is influenced by interventions of its operator. While former research on HMS has focused on physical and ergonomic characteristics of the interaction for optimizing construction and force feedback properties, today the main topic is estimating consequences of automation. Since technical systems are getting more complex often cognition, memory span, and mental models are being of concern. Some typical examples of systems that are analysed and developed from an HMS viewpoint are in high risk environments like the aviation domain, energy and power management (nuclear power plant), and chemical process operation.

1.2 Objective

The objective for modeling behavior of HMS operators is to facilitate simulation based design support, training, and deployment in assistive technology.

1.2.1 Simulation based design support

In the development process of HMS' it is cost efficient to detect design flaws as early as possible. Thus tests are conducted with prototypes or even mockups instead of finished products. The use of simulations of the technical system is common practice in testing e.g. in automobile industry. Additionally simulations of cognitive capabilities of the potential user are effective for questions about human reliability (Amalberti 2002). They can be efficient in large scale multi-user scenarios (computer generated forces: Jones et al 1999) or when a single prototype test is very expensive (e.g. aircraft cockpit automation: Lüdtkke 2002).

1.2.2 Training

Insights from simulations of different strategies in cognitive models can be measures for "difficulty of learning each strategy, efficiency of using each strategy once learned, generality of each strategy to the range of [...] problems, retention of the strategies, and transfer" (Rittle-Johnson & Koedinger 2001). Although stated for simple arithmetic tasks these measures can be used to decide which strategy to train for operating HMS.

1.2.3 Deployment in assistive technology

Besides using simulations of cognitive processes as knowledge based support system (perfect cognitive models making no errors are *expert systems*) realistic (error making) cognitive models have the potential to be deployed in adaptive automation systems (Parasuraman et al 2000). Such systems adapt their behavior to external conditions normally detected in the technical system or its environment. Taking not only the technical or environmental state but also that of the operator into account leads to a more effective task allocation between automation and human operator. Realistic cognitive models running parallel to the HMS can be used to predict current or future operator states. A famous example for this kind of automation are intelligent tutoring systems (e.g. Leuchter & Urbas 2002).

The rationale of the use of the cognitive architecture ACT-R is to make the modeling process more efficient by introducing a priori constraints on memory and processing.

1.3 Dynamics

The need for timing arises from the dynamics of the operated system. Thus for this report the most important property of an HMS is its dynamics. Many approaches for conceptions on dynamics of HMS exist. Often they are directed by control theory or complex problem solving. Since they do not fully fit in modeling timing in this paper a scheme is proposed that is derived from experience gained modeling cognitive processes of air traffic controllers (Niessen et al. 1998) and process control.

This approach is presented object-oriented. Figure 1 shows the framework as a UML diagram (unified modeling language, common used in software engineering).

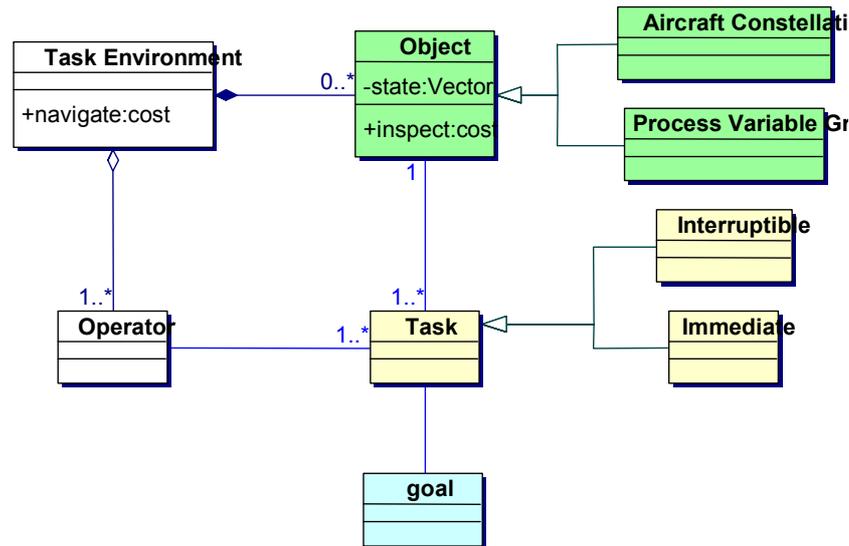


Figure 1: Framework for dynamics.

The diagram shows three columns. The left one is the defines the system boundary: The environment and the modeled subject(s). The middle column shows the concept itself. The right column shows concretizations. Above: in two domains (green): en-route air traffic control (ATC) and process control (PC e.g. chemical plant), below: concretizations of tasks.

The task environment consists of a number of objects. Example for objects are aircraft or their constellations (ATC) and process variables such as temperature or pressure (PC). One or more operators are responsible. An operator is assigned one or more tasks to fulfill. Every task is associated with exactly one object. A task is consequently associated with a goal. Tasks or more exact fulfillment of a task can either interruptible by other tasks or has to be immediately worked off.

Dynamics results first from new objects or tasks appearing in the situation. But for most tasks in HMS the state of the objects is also needed. It can be acquired by the HMS interface (often a screen). To avoid confusion not all information is normally displayed. But for the execution of a certain task an unusual information can be needed. Then the object has to be inspected and that information gathered (e.g. transponder code on ATC screen). Some interfaces are designed in several screens (e.g. control room displays for PC) so that the operator has possibly to navigate in the task environment to the needed information's display. The second reason for dynamics is the changing of objects state. Change can be initiated from actions of the operator and from environmental interference. Both result in a more or less delayed change behavior.

The treatment of this framework is that a task environment has a certain number of objects at a time and following a number of tasks/goals that are valid and have to be fulfilled at any one time (homeostatic and parallel goals, Aasman 1995). The characteristics is the amount of them (interruptible separated from immediate), the change rate and expenses for navigation/inspection. One application of this framework is a comparison between different domains/tasks to estimate the need to use working memory in favor for external memory (HMS interface screen) on the basis of inspection/navigation cost and the number of continuous tasks.

This framework of describing dynamics is used in the following section to discriminate between certain task environments.

2. Task Environments

In this section the dynamics of several interactive task environments is discussed: car driving, en-route air traffic control, human-computer interaction, aircraft piloting, interactive computer games, and process control in chemical plants. The consequences for scheduling of subtasks are deduced.

Car driving: Although driving is not a top-down activity it is composed of several subtasks that are all active at the same time (“homeostatic”, Aasmann 1995). Cnossen (2000, p. 40) and Salvucci (2001) enumerate subtasks of driving as: speed control, steering, visual search, and distance keeping. Car driving is a highly dynamic task because the state of the vehicle and its environment changes constantly possibly without prior warning.

Apart from modern automation like adaptive cruise control there is not much need for a memory for situation awareness because the situation can be easily directly perceived at environment (objects are primarily other cars) and instruments.

En-route air traffic control: Controllers monitor the movement of aircraft on radar screens. Although the traffic is planned in advance weather conditions or other unexpected events influence plan achievement. Thus controllers have to command pilots for other direction, speed or altitude to avoid dangerous approximation. The situation is dynamic: New objects (aircraft) enter the sector, the state of the objects (position, altitude, speed, direction) changes normally upon request of the controller. The change rate of tasks is rather low, because new aircraft enter a controlled area about every some minutes. Nevertheless it is necessary to monitor aircraft because pilots could not exactly perform commands or deviate from their routes. Controllers need a representation of the current situation in order to achieve anticipation, conflict resolution and monitoring (Niessen et al. 1998).

Human-computer interaction: HCI does not necessary include the notion of dynamics: Objects that are observed are GUI-elements that display values. Depending on the model that is displayed they are changing and have to be monitored. Although many dynamic human-machine-systems have computer mediated interfaces or can be used in PC based simulations there are no direct results for cognitive models of HCI.

Aircraft piloting: Through the introduction of *fly by wire* concepts and automation technology in the modern *glass cockpit* pilots mostly fulfill a supervisory control task. Objects are the measure instruments and displays. The subtasks are precisely in hierarchical standard operating procedures defined. A major part of all human factors errors in aviation results in lack of situation awareness in that the current mode of the automated system is misinterpreted (e.g. Lüdtkke et al. 2002). An interpretation on the basis of the proposed dynamic modeling framework is that procedures have to be treated as objects which was the way how an existing ACT-R pilot model was realized (Schoppek et al. 2000).

Interactive computer games: Similar to car driving new objects in interactive computer games can occur and change their state without the player’s interaction. In first person shooter games (Laird 2001) the most important feature is the position of ob-

jects (enemies or other characters). Since navigation is achieved through movement in virtual space inspection cost for objects are high. Because of this and to make deliberative behavior possible it is necessary to maintain a mental representation of the virtual world.

Process control in chemical plants: In contrast to computer games, air traffic control and car driving the super structure of the environment does in general not change, i.e. the pure number of objects does normally not change. Nevertheless, due to start up and shut down of single unit operations and the possibility to change the interconnection between the unit operations the relation between the objects and the resulting dynamic behavior of the process is subject to change. Furthermore the course of variables in time is most often not predictable by linear extrapolation. In consequence a mental representation of the current relations between the objects and a image of the history of the process is necessary for the supervisory control tasks.

Thus operators need not only a mental representation of the objects' values but also of their dynamical features as their state. Operators typically have several displays in the control room and one monitor can display several different views. They use structural presentations of the chemical plant where the values of process variables are shown and trend pictures of the development of a variable over time.

2.1 Requirements for Scheduling

There are different characteristics of dynamic task environments that demand different processing of the dynamics. Within the proposed framework dynamics is handled through composition of goals, the scheduling of their execution, memory for objects' state (situation awareness), and update of objects' state (which can be seen as one of the goals).

Since providing a memory with features such as adaptive decay and partial matching for a situation's objects' state can be easily achieved with modern production systems like ACT-R (Anderson & Lebiere 1998) we focus on goal scheduling.

The task characteristics of HMS environments is either more reactive or more deliberative. In reactive task environments objects and associated tasks are perceived, new goals are generated, and executed. In the deliberative case subtasks can be postponed and subtasks have different urgency and priority. A memory for postponed tasks and strategies for scheduling them is needed.

The resulting multiple tasking is not the same as that of Lee & Taatgen (2002). They report on multi-tasking in the time gaps between action in asynchronous cognitive subsystems. Also scheduling as a task itself (e.g. Nellen 2002) is not the scope of this treatment. The time resolution is much less fine than in the task described by Gray et al. (2000).

For **car-driving** scheduling of tasks/goals does not need sophisticated algorithms because they are equally important. If secondary tasks like cell-phone dialing are added breakpoints between the tasks have to be defined (Salvucci 2001).

In **en-route air traffic control** controllers have too limited resources to monitor every aircraft on the screen. Thus they have to schedule their updating sequence, anticipating, and conflict resolution. Niessen et al. (1998) suggest an algorithm that takes inferred importance and two timing measures into account: Every object stores a

timestamp when it was last updated, and during anticipation a duration until an event (e.g. time to collision) is computed. Other examples for cognitive models with multi-tasking exist (AMBR: Gluck & Young 2001, and Lee & Taatgen 2002)

Most **HCI** problems can be formalized in hierarchical task analysis models like GOMS. Thus there is no need to schedule between subtasks. The situation changes when there are several primary tasks (e.g. Salvucci 2001) or when the displays show the state of a dynamic human-machine-system (e.g. air traffic control: Freed 1998)

In **aircraft piloting** there are procedures to be executed. Multi-tasking and scheduling are only needed on a very top level (sequencing of procedures: starting, following way points, initiating landing). One additional requirement is the handling of asynchronous air traffic control commands.

Interactive computer games: A multitude of parallel goals exist that have to be examined and according to the current situation postponed or pre-drawn. To do so the player has to take both his or her memory and the perceived situation into account.

Process control in chemical plants: Depending on the inferred situation changes of goals have to be made. A major concern is to be able to suspend and to resume tasks. An operator model has to store the points where operation on one task was left and another was resumed.

2.1.1 Timed Conditions

If cognitive models are applied in engineering it is usually the aim to predict frequencies of erroneous production selection or memory slips or execution or learning durations for the whole task (e.g. within the GOMS framework). But there are also errors according timing and thus the need not only to model adaptive sequences of production selection but also the use of time and duration in conditions of productions.

An example for such a modeling demand is in process control: Sometimes a task has to be abandoned when too much time has elapsed. Thus duration has to be recalled in productions' conditions. To achieve this the system's real time would have to be retrieved and stretched or compressed according to the current workload.

2.1.2 Objects Getting More Important Over Time

In (en-route) air traffic control scheduling depends on timing: It is important to update the state (mostly position and altitude) of the objects as often as possible. But due to the other subtasks aircraft can only be monitored from time to time. But the need to update an objects' features in the mental representation gets bigger the longer the object has not been modified. Thus decay of activation depending on accesses to chunks is a contrary concept.

Such chunks representing such special elements of the situation under supervisory control have to be used in a certain way: Special productions have to refresh their activation depending on the current need to update it. Niessen et al. achieve this behavior through direct manipulation of activation parameters from outside ACT-R in the production-cycle-hook.

3. Conclusions: Extensions to the Architecture

On the basis of a brief review of these tasks and their requirements for scheduling and multi-tasking some needs for ACT-R can be drafted to fit it to modeling operator cognition in complex dynamic human-machine systems. We are currently implementing these extensions for modeling process control of chemical plants.

3.1 Linking and Embedding to Task Environment

Although ACT-R/PM made it possible to connect a model with a task environment there are problems for engineering: Normally there exists a big simulation or an API not accessible from LISP or making it hard to create a GUI within the LISP process. While one can cope with this restriction (e.g. Ritter et al. 2002) it would be efficient to embed an ACT-R model into a high-level framework for the “normal” control of a system and only execute a specialized ACT-R sub model for questions like memory errors from there. This would help controlling a real world situation with a simple outer model (without ACT-R) and only pay attention to special situations for that a more precise ACT-R model would be built.

Inside the ACT-R model there were less need for multi-tasking and scheduling and additionally communication with the task environment could be achieved through appropriate instantiation of chunks in the model during its start-up.

3.2 Recall of Duration

A new function for recalling duration information has to be added. Setting a named reference point that is stored as chunks in the working memory and thus can be forgotten or confused allows for retrieving elapsed time since setting it.

Perception of time is depended to the workload and the “mode”: If concentrating on duration measurement high workload leads to underestimating, if recall is retrospective high workload leads to overestimating elapsed duration. Recall has to be possible in both modes and must be stretched or compressed according to subgoalings.

3.3 Scheduling

Tasks are to be represented and executed as chunks from a ACT-R “middleware” such as ACT-GOMS (Schoppek et al. 2000). But in contrast to ACT-GOMS it must include a scheduler for subtasks. They are to be marked interruptible and immediate during modeling time. Priority and urgency like in PDL could further guide the scheduling process.

An additional requirement in chemical plant control is that some tasks may not be carried out in parallel with others and that there are other dependencies possible. But a scheduler should not take also such information into account but it had to be modeled explicitly because this is an important source of control errors.

4. Acknowledgments

This work has been supported by Volkswagen Foundation in the program “Junior Research Groups at German Universities”.

This report is an extended version of a presentation given at the Fifth International Conference on Cognitive Modeling (ICCM) in Bamberg, Germany on April 10 - 12, 2003.

5. References

- Aasmann, J. (1995). *Modelling Driver Behaviour In Soar*. Leidschendam: KPN Research.
- Amalberti, R. (2002). Use and Misuse of Safety Models in Design. In F. Grandoni (Ed.), *Dependable Computing EDCC-4. 4th European Dependable Computing Conference, Toulouse, France, October 23-25, 2002. Proceedings* (p. 1). Berlin: Springer-Verlag (LNCS; 2485).
- Anderson, J. R., & Lebiere, C. (1998). *Atomic Components of Thought*. Hillsdale, N.J.: Erlbaum.
- Byrne, M. D. (2001). ACT-R /PM and menu selection: applying a cognitive architecture to HCI. *International Journal Human-Computer Studies*, 55 1-44.
- Cnossen, F. (2000). *Adaptive strategies and goal management in car driving*. Proefschrift Rijksuniversiteit Groningen. Retrieved Oct. 30, 2002 from <http://www.ub.rug.nl/eldoc/dis/ppsw/f.cnossen/thesis.pdf>.
- Freed, M. A. (1998). *Simulating Human Performance in Complex, Dynamic Environments*. Evanston, Illinois: Phil. Diss., Northwestern University. Retrieved Oct. 30, 2002 from <http://www.andrew.cmu.edu/~bj07/apex/docs/dissertation/dissertation.doc>.
- Gluck, K. A. (2002). *Proceedings of the Winter Workshop on ACT-R Models of Human System Interaction*. Retrieved Oct. 29, 2002 from <http://www.dtic.mil/AFRL/afri.html>.
- Gluck, K. A., & Young, M. J. (2001). The AMBR Model Comparison Project: Multitasking, the Icarus Federation, and Concept Learning. In J. D. Moore, & K. Stenning (Eds.), *Proceedings of the 23rd Annual Conference of the Cognitive Science Society, August 1-4, 2001, Edinburgh, Scotland*. Mahwah, NJ: Lawrence Erlbaum.
- Gray, W. D., Schoelles, M. J., & Fu, W.-T. (2000). Modeling a continuous dynamic task. In N. Taatgen & J. Aasman (Eds.), *Proceedings of the Third International Conference on Cognitive Modeling* (pp. 158-168). Veenendaal, NL: Universal Press.
- Jones, R. M., Laird, J. E., Nielsen, P. E., Coulter, K. J., Kenny, P., & Koss, F. V. (1999). Automated Intelligent Pilots for Combat Flight Simulation. *AI Magazine*, 20(1) 27-41.
- Laird, J. E. (2001). It knows what you're going to do: Adding anticipation to a Quakebot. In *Proceedings of the fifth international conference on Autonomous agents 2001* (pp. 385-392). New York, NY: ACM Press.

- Lee, F. J., & Taatgen, N. A. (2002). Multitasking as Skill Acquisition. In W. D. Gray, & C. Schunn (Eds.), *Proceedings of the 24th Annual Conference of the Cognitive Science Society* (pp. 572-577). Hillsdale, NJ: Erlbaum.
- Leuchter, S., & Urbas, L. (2002). Simulation Based Situation Awareness Training for Control of Human-Machine-Systems. In Valery Petrushin, Piet Kommers, Kinshuk, & Ildar Galeev (Eds.), *IEEE International Conference on Advanced Learning Technologies. Media and the Culture of Learning. Kazan, Russia: Sep 9-12, 2002* (pp. 34-39). Palmerston North, New Zealand: IEEE Learning Technology Task Force.
- Lüdtke, A., Möbus, C., & Thole, H.-J. (2002). Cognitive Modelling Approach to Diagnose Over-Simplification in Simulation-Based Training. In S. A. Cerri, G. Gouarderes & F. Paraguacu (Eds.), *ITS 2002* (pp. 469-506). Berlin: Springer-Verlag (LNCS; 2363).
- Nellen, S. (2002). *How Humans solve Scheduling Problems: Analysis of Human Behavior in the Plan-A-Day task* (Diploma Thesis) Department of Psychology, Ruprecht Karls Universität, Heidelberg. Retrieved Oct 30, 2002 from <http://www.ub.uni-heidelberg.de/archiv/2129>.
- Niessen, C., Leuchter, S., & Eyferth, K. (1998). A psychological model of air traffic control and its implementation. In F. E. Ritter & R. M. Young (Eds.), *Proceedings of the Second European Conference on Cognitive Modelling (ECCM-98)* (pp. 104-111). Nottingham: Nottingham University Press.
- Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, 30(3) 286-297.
- Ritter, F. E., van Roy, D., & St. Amant, R. (2002). A User Modeling Design Tool Based on a Cognitive Architecture for Comparing Interfaces. In C. Kolski, & J. Vanderdonckt (Eds.), *Computer-Aided Design of User Interfaces III, Proceedings of the 4th International Conference on Computer-Aided Design of User Interfaces CADUI'2002* (pp. 111-118). Dordrecht: Kluwer Academic Publisher.
- Rittle-Johnson, B., & Koedinger, K. R. (2001). Using cognitive models to guide instructional design: The case of fraction division. In *Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society* (pp. 857-862). Mahwah, NJ: Erlbaum.
- Salvucci, D. D. (2001). Predicting the effects of in-car interface use on driver performance: an integrated model approach. *International Journal of Human-Computer Studies*, 55(1) 85-107.
- Schoppek, W., Boehm-Davis, D.A., Diez, M., Hansberger, J., & Holt, R.W. (2000). Letting ACT-R fly - A model of the interaction between trained airline pilots and the flight management system. In *Proceedings of the Seventh Annual ACT-R Workshop 2000*, Carnegie Mellon University, Pittsburgh, PA.

Opportunities and Challenges of Modeling User Behavior in Complex Real World Tasks

WOLFGANG SCHOPPEK & DEBORAH A. BOEHM-DAVIS

University of Bayreuth, Germany

George Mason University, Fairfax, VA

Keywords: cognitive modeling, human-machine interaction, automation, task analysis, human error

1. Introduction

In basic research, cognitive modeling has proven a valuable methodology for developing theoretical assumptions, testing their dynamic interactions, and exploring the scope of theories. Cognitive architectures such as ACT-R (Anderson & Lebiere, 1998) or Soar (Rosenbloom, Laird, & Newell, 1992) provide a common basis for different models and enhance communication and exchange of solutions. In applied contexts, modeling of real tasks and operators could further the understanding of human-machine systems; validated models could provide an objective guide to design and training decisions. However, they only provide a first step towards rapid development of valid real world models, because the architectures do not constrain many decisions the modeler has to make. This paper will illustrate this issue through a case study of developing and validating a model of complex real-world behavior – that of flying an aircraft.

1.1 Requirements for modeling complex real world tasks

Models of real world tasks have requirements that are not always applicable to models used solely for research purposes. Specifically, to be useful, models first must be developed in reasonable amounts of time. Although what is regarded as reasonable will depend on the specific task and the questions the model should answer, the typical development time of one man year is too long for most applied cases. Second, models must be valid. Depending on the application or area of interest, there are different kinds of validity for models: structural, predictive, etc.. We claim that first of all, models must make correct predictions about some aspects of performance, ideally without the need for empirical testing. Considering how rare it is for models to

make accurate predictions even for basic research, this requirement may be even more ambitious than the first.

In this paper, we argue that, in general, cognitive architectures can help shorten development time as they come with a built-in structure for representing tasks that can help form the foundation for the model-building activity. We further argue that cognitive architectures can also help address the second requirement – that of ensuring validity. Most cognitive architectures are based on assumptions that have been developed and tested empirically. For example, within the ACT-R cognitive architecture, early theories of skill acquisition were compared against empirical data (Anderson, 1982). More recently, models of skill acquisition have been applied to and validated in a broader range of applications (Taatgen, 1999; Lee & Anderson, 2001; Taatgen & Lee, 2003). As a consequence, standard techniques for modeling learning of procedures from written instructions are now available.

The problem with architectures, however, is that they are content independent and do not sufficiently constrain the modeling of specific tasks. Table 1 illustrates this for the example of decision making. Assuming that decisions can be described by some form of mapping from actions to conditions, and that conditions, actions, and the decision rule itself are mentally represented, there are a number of possibilities for these representations, which may occur in many different combinations. All the possibilities shown in Table 1 can be implemented in an architecture like ACT-R, but the architecture does not suggest what combinations a researcher should select for modeling a given task.

To leverage cognitive architectures for modeling real world tasks, standard solutions that are architecturally grounded are needed to represent entities such as “decision making”, “acquisition of cognitive skills”, or “intention management”. Theoretical standard solutions would be formulated in the language of a specific architecture and empirically tested (not ad hoc) in a range of applications, thus having the status of theories.

Table 1: Framework for a symbolic theory of decision making that would support modeling of complex real world tasks. The sinuous lines indicate that various combinations between representations of conditions, decision rules, and actions are possible. Solutions used in ACT-Fly are printed in italics.

Condition	Decision rule	Action
Class of situations - classification by similarity - classification by rules	Similarity matching	<i>Method</i> - represented procedurally - represented declaratively
<i>Specific parameters of a situation</i>	<i>Fixed symbolic rule</i>	<i>Single step</i>
Combination of the former two (“situation plus exception“)	Deduction of rule from general knowledge	

Such theories can best be developed through tight cooperation between applied and basic researchers. Modeling in basic research often aims at developing architecturally grounded theories of certain processes, but it needs to be informed by the uses to which it will be put in order to be useful to applied researchers. Basic models often ignore how the processes they represent are influenced by real-world context. Thus,

from the view of the basic researcher, models of complex real world tasks are beneficial because the applied model is a good test bed for the value of the theory. Also, if the modeler of a real-world task carefully specifies what parts of the model are derived from the architecture and for what parts new solutions had to be developed, the basic researcher is informed about ways in which the architecture needs to be extended.

Thus, we are arguing here that the efficiency of modeling behavior in real world tasks can be improved when models are developed within established architectures even though not all solutions to specific modeling problems can be derived from the architecture. For our project, we chose ACT-R as our architecture, because it is a widely accepted psychological theory with a broad empirical basis and a modeling language. On its symbolic level, ACT-R distinguishes between declarative memory, made up of a network of typed “chunks”, and procedural memory, consisting of production rules. ACT-R also assumes a subsymbolic level ascribing continuous parameters such as activation or utility to each symbol. In memory retrievals, for example, both levels determine what is retrieved. On the symbolic level, the memory element must match the symbolic specification of the retrieval command; given one or more chunks are matching the specification, the subsymbolic level selects the most active of these chunks and determines the retrieval latency.

We used the ACT-R architecture to develop a model, called “ACT-Fly”, that simulates the interaction between airline pilots and the flight management system when completing a descent. The goals for the modeling project were to (1) gain an idea of how long it might take to produce a model of this complex real-world task using a cognitive architecture and (2) determine how well this model could predict human performance. In developing the model, we started with GOMS task analyses and then implemented them in ACT-R 4.0. Some problems posed by the task could neither be solved with GOMS nor with standard ACT-R; thus we added minor extensions and modifications to ACT-R, which are described below.

1.2 Characteristics of our modeled task

Until recently, typical tasks modeled with ACT-R were laboratory experiments such as memorizing lists of words (Anderson, Bothell, Lebiere, & Matessa, 1998) or discriminating previously learned statements from distracters (Anderson & Reder, 1999); tasks characterized by clear, shallow goal hierarchies, a repetition of very short trials with the same underlying structure, a static environment, and low demands for prior knowledge. Flying, in contrast, involves heterogeneous goals that may compete for limited resources. For example, the goal of monitoring the plane passing a critical waypoint (location in space) competes with the goal of encoding a new clearance from air traffic control (ATC). Also, much prior knowledge must be brought to the flying task.

The specific task we chose to model is flying a simulated Boeing 747-400 from the end of the cruise phase to the initial approach fix using the aircraft’s automated flight management system. A number of different specific scenarios were flown to include a variety of conditions, such as different ATC clearances or descent profiles. The task was a vertical navigation task, chosen because this is one of most error-prone aspects of automated flying (see, for example, Sarter & Woods, 1992).

Two basic modes of automation are available for accomplishing the task: a fully automated mode - called VNAV - where the autopilot receives most of the reference values from a preprogrammed flight plan; and semiautomatic modes (referred to as FLCH and V/S), where the reference values must be provided by the pilot. VNAV is generally the preferred mode of operation because it optimizes the flight profile and fuel consumption. However, if ATC requires quick changes to the flight plan, the pilot can respond more flexibly using semiautomatic modes. In all of these modes, the behavior of automation and aircraft must be monitored and set points must be provided on time.

Even though the range of tasks modeled within the ACT-R framework has increased considerably in the past several years, this task presented a number of challenges. For example, although the challenge of modeling dynamic tasks has been addressed in the past few years, resulting in models of air traffic control (Lee & Anderson, 2001) and driving tasks (Salvucci, 2001), the dynamics involved in the present task are different (relatively slow). Users typically have to wait for minutes until they can judge the success of a certain intervention. Because new and rather unpredictable demands can arise during the waiting time, multiple interleaved streams of behavior are the result. Thus, some of the features of the present task required us to find new solutions that are not obvious in the architecture and cannot be derived from existing models that successfully predict behavior in other tasks.

The flying task is also difficult to validate as it is a typical supervisory control task, where the user must mainly observe what the automation is doing, resulting in very sparse records of observable behavior. This makes the comparison between simulated and real behavior difficult.

2. ACT-Fly model

In developing our model, we drew theoretical background from a combination of GOMS (Card, Moran & Newell, 1983) and ACT-R. We conducted task analyses based on flight manuals, interviews with subject matter experts, and interactions with the flight simulator. These analyses produced a set of methods by which pilots use the flight management system to fly a descent which we represented using GOMS elements in an NGOMSL (Kieras, 1997) framework. We then had to develop ACT-R representations of those GOMS elements and NGOMSL methods and a way of translating those analyses to ACT-R code. The translation from the NGOMSL code to ACT-R code was accomplished through an Excel spreadsheet tool. This idea is similar to Salvucci & Lee's (2003) "ACT-Simple", which allows the translation of a simple modeling syntax into ACT-R. Compared to ACT-Simple with its focus on key-stroke level processing, our solution is targeted at modeling decision making through a representation of higher level procedures.

With early versions of the model, we found that when we relied entirely on methods, the model was too rigid to respond to unexpected events. Specifically, we found that the sequential structure of methods often did not match the less predictable order of events in the environment. Another problem with the method-only controlled version was the lack of situation awareness. The scope of a method is typically limited to local aspects of a task, and so there was no inherent need to create a "big picture". To achieve more flexibility and better situation awareness, we introduced an additional level to the control structure that operates in a non-sequential, rule based manner. In

the terms of Table 1, situations at this level are classified by rules, decisions are also made by a fixed symbolic rule, and the actions taken are either single steps or the execution of a method.

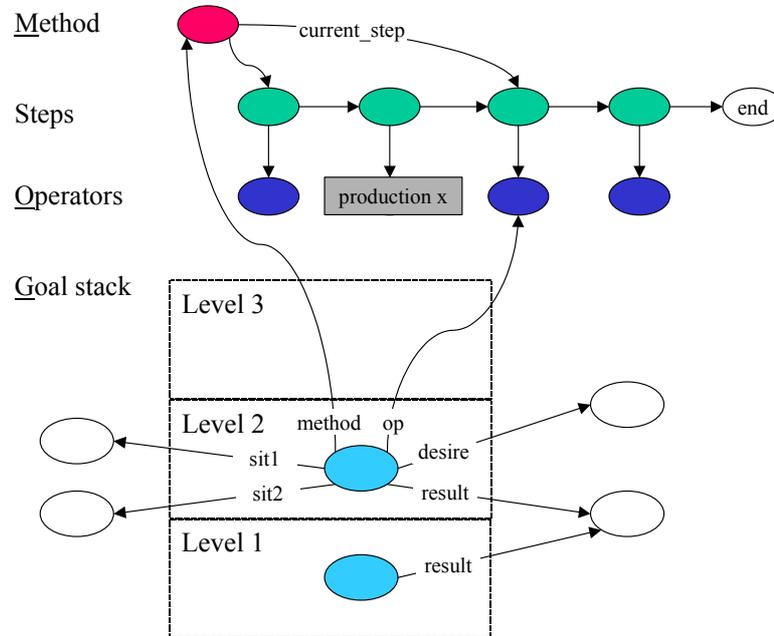


Figure 1: Declarative structures of ACT-Fly. The structure at the top represents a method. The structure at the bottom shows the goal stack. Level 1 and Level 2 are holding one goal chunk each. The goal chunk on Level 2 has its slots filled with chunks representing the situation (e.g. the current method) and current setpoints ("desire"). Results are passed across the levels.

The model communicates via a TCP/IP socket connection with a PS1 747-400 desktop flight simulator manufactured by Aerowinx. The model writes its commands and requests to a socket and reads answers from it. Likewise, the connection software reads commands and requests from the socket, forwards the commands to the simulator, gets the requested values from the simulator, and writes them to the socket.

Technically, the model is divided into three parts. A domain independent part contains generic operators and functions that handle the execution of hierarchical methods and the management of intentions. A domain specific part contains domain knowledge such as methods and rules. The third part contains the interface to the external environment. The benefits of this separation are that the general part can be used for modeling other tasks (done e.g. by Holt, Hansberger, Chong, & Boehm-Davis, 2002, and Schoppek, 2002), and that, to a certain extent, the general part can be improved without changing the domain specific parts.

The activities demanded from the pilots range from situation-specific decision making (e.g. deciding which mode to use for a specific leg¹) to the execution of standard

¹ leg: the flight path between two waypoints

procedures (e.g. entering an altitude restriction into the flight management computer). To account for this variety of actions, ACT-Fly's control structure was based on a goal stack limited to three levels with a clear division of responsibilities among the levels. Each of the levels is represented by a goal chunk of a specific type (chunk is the ACT-R term for a declarative memory element).

Level 1 is the bottom level. It can be characterized as the decision making level. At this level, rule-based decisions are made as to what goals are pursued and what methods are selected to accomplish these goals. Level 1 also serves as the goal manager for Level 2. Finally, Level 1 contains some basic problem solving productions. The goal chunk representing this level stores molar information about the situation, such as the phase of flight, the position of the aircraft in the flight plan, or the status of ATC clearances.

Level 2 can be characterized as the method level. It is the level of operation described by frameworks like GOMS. Similar to GOMS, our methods consist of operators, subgoals, and decision steps. Level 2 can execute hierarchical methods of virtually any depth on one level. This is possible because subgoals are not stacked on top of each other. Rather, superordinate goals are released to memory and retrieved later on. Storage and retrieval of goals are handled on Level 1. This design has several advantages. First, the concept of a goal stack has been criticized for providing unrealistically perfect memory for goals (Altmann & Trafton, 2002). In ACT-Fly, goals do not simply appear on top of the goal stack once the previous goal has been popped, but must be retrieved from memory - a process that can fail and can predict certain types of errors. Second, as control is returned to Level 1 after the execution of each submethod, the course of action can be corrected during the execution of a long, nested method. With a more traditional goal stack, the system would be "blocked" for the time such a method is executed. Although Altmann & Trafton (2002) also argue that suspended goals are subject to decay, their model does not explicitly allow a mechanism for interrupting the execution of a current task to perform an alternate task. Thus, our solution makes the model more flexible and ready to handle interruptions.

Most steps within methods are represented as declarative chunks linked through associations. Thus, the retrieval of the next step is cued by the current method and the previous step, but is not constrained symbolically. That enables the model to produce errors of omission and of commission in the execution of methods. Another advantage of the associative linking of steps is that methods are learned "by doing", using ACT-R's associative learning mechanism. This representation of procedures is not standard in ACT-R modeling, but there is some evidence supporting this assumption. For example, Byrne and Bovair (1997) explained their findings on the "post-completion error" with a spread of activation from one step to the next; successful models of associative sequence learning also exist (Altmann, 2000; Lebiere & Wallach, 2001). Using the execution of one step to cue execution of the next step through associations between the two steps corresponds best to an intermediate state of proficiency as it is expressed by Rasmussen's (1986) rule-based behavior. However, that is not the only mechanism available for representing procedures in our model; it was also possible to integrate completely proceduralized sections into a method to be executed as a whole. The ACT-Fly methods serve different functions. There are methods that perform input operations to the automation, methods that do mental calculations with flight parameters to support decisions, and methods that return classifications of the current situation to maintain situation awareness.

We assume three basic types of operators, which are part of every step. (The separation of steps and operators allows the occurrence of the same operator in multiple steps.) Internal operators perform memory operations, comparisons, or mental calculations; they are represented as single production rule. External operators perform actions such as pressing buttons or dialing values. Perceptual operators provide representations of the environment such as current values read from displays or ATC clearances.

Level 3 represents the interface between central cognition and peripheral systems. Since ACT-Fly does not simulate perceptual or motor processes in detail, input-output operations are modeled on an abstract level. When the model requests information from the environment, a specialized chunk is pushed on Level 3, completed with the requested information (through the TCP/IP-socket connection with the flight simulator), and the results are transferred to the goal chunk of Level 2. Similar steps are performed for motor commands. After being cleared from the goal stack, the I-O-chunks remain as episodic traces in memory.

Our solution of a goal stack with three levels has similarities with the “memory for goals” approach by Altmann & Trafton (2002) (which in the meantime has become a standard modeling technique in ACT-R). In particular, goal processing at Level 2 complies with Altmann & Trafton’s approach. However, assuming separate levels for problem solving, procedural processing, and perceptual-motor processing made modeling of our complex task much easier. We believe it reasonable to assume that goals corresponding to different levels do not interfere with each other when a pending goal is retrieved from memory. Moreover, the goals of Level 3 never stay there for longer short periods of time, making it plausible that the pending goal on Level 2 is virtually always retrieved correctly.

3. ACT-Fly's predictions and performance

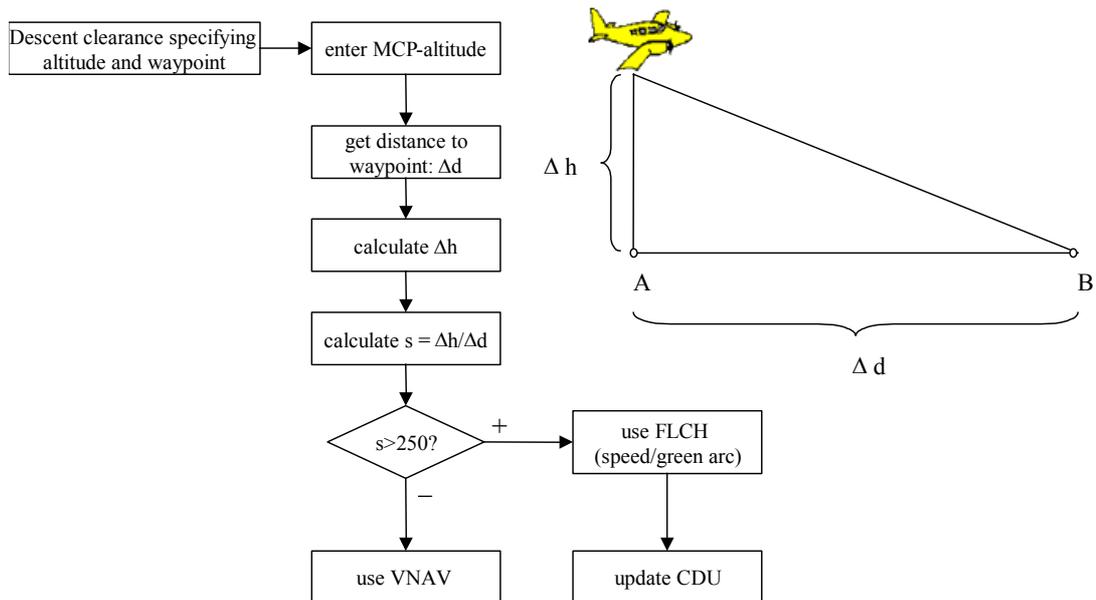


Figure 2: Flow chart of the central method of ACT-Fly that decides what mode to use for a given descent leg. The boxes represent goals for which ACT-Fly has appropriate methods.

Central to the behavior of the ACT-Fly model is a method that decides what mode of the flight management system to use for a given leg. A flow chart of this method is depicted in Figure 2. This was an area where it became clear that existing models in current architectures could not provide guidance about how to represent this task. For our model, we interviewed several subject matter experts and discovered several alternative methods that could have been used. From these reports, we abstracted the method shown in Figure 1, which is more explicit and simpler than any individual model described by an expert. ATC-clearances and situations in which a waypoint gets close can trigger execution of this method. Our goal was to determine how well the use of this single abstracted model could account for typical pilot behavior.

Performance of the ACT-Fly model was assessed by flying informal as well as standardized simulated scenarios. In these scenarios, the model ran on one computer, the flight simulator on a second computer. Most of the communication between the programs was handled through an TCP/IP socket connection. Only a few values from the flight simulator had to be entered manually on request of the model; ATC clearances were entered manually, too. Most scenarios started with the simulated aircraft being close to the “top of descent point” calculated by the flight management system. ACT-Fly recognizes the situation on its own initiative. Time synchronization was accomplished by letting ACT-Fly, which usually runs faster than the flight simulator, wait for the simulator when its simulated time was more than five seconds ahead of real time.

Figure 3 shows the results of simulations of two descent scenarios. For each scenario, we ran six simulations. As can be seen from the mean deviations around each data point, there was little variability in model performance. Generally, the model flew the desired paths precisely. In the scenario without ACT-clearances, the model se-

lected the VNAV mode for most of the legs, except for the leg between BOLDR and MENLO, where FLCH was used. In the second scenario, where a clearance was issued after the top-of-descent point (T/D) had been passed (making VNAV an inappropriate mode choice), the model selected the semiautomatic FLCH mode right after the clearance.

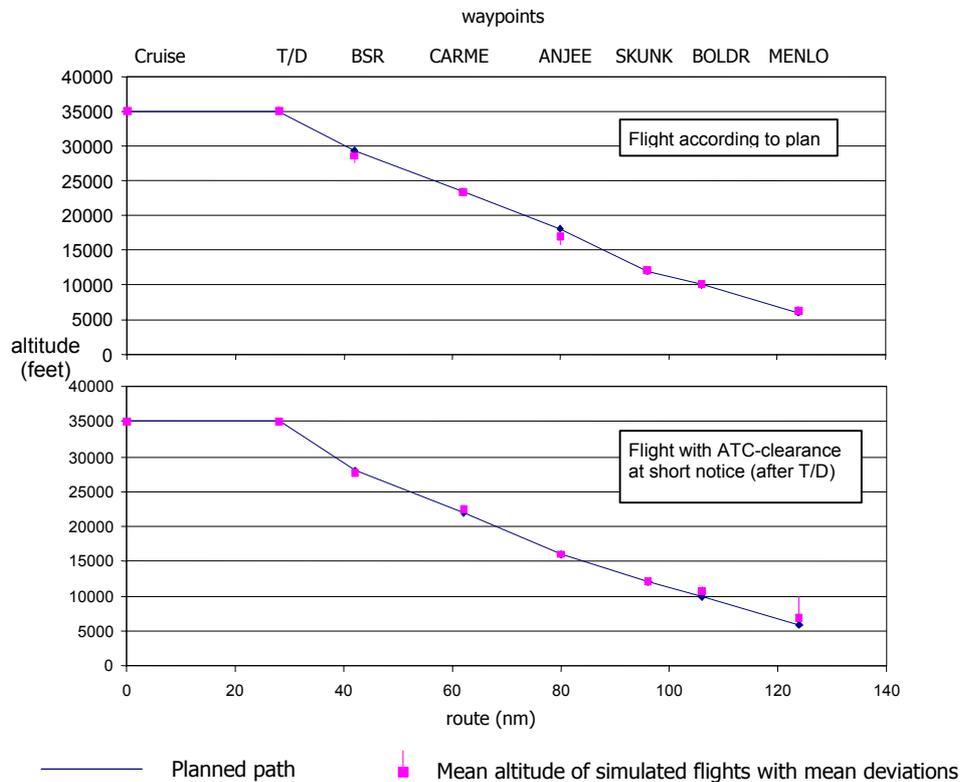


Figure 3: Flight paths of simulated flights in two scenarios (six simulations in each scenario). The upper panel shows data from a scenario without ATC clearances, the lower panel data from a scenario with the clearance “Cross ANJEE at 16000”.

In all twelve simulations, each consisting of approximately 16 simulated minutes and 1300 production firing cycles, four errors of omission occurred. One of these caused the large deviation at waypoint MENLO visible in the lower panel of Figure 2, because the model could not recover from the error and failed to update the altitudes in the mode control panel. Although the single complete failure to recover out of four errors of omission is too low for a stable estimation, we think that this proportion is much worse than what is found in humans (Reason, 1990). Observations from our informal testing support this view.

The model committed a number of other errors; these impaired performance more gradually. For example, there were three cases where the model failed to resume a deferred action (intention). In general, error recovery was best when the method in which the error occurred was directly triggered by the identification of a relatively stable situation (initiated by Level 1 of the control structure). Error recovery was worse when the affected method was triggered by another method (initiated by Level 2), and when it was triggered by a transient situation.

To compare performance of real pilots with the model, we conducted a study with five commercial pilots, certified to fly the Boeing 777 aircraft (which has a flight management system very similar to the Boeing 747-400). Four of the pilots served as first officers on the 777; one served as a captain. The participants flew two scenarios with the aforementioned PS1 flight simulator. Flight performance, prompted recall of flight parameters, and eye-tracking data were recorded. (For results of the eye-tracking aspect of this work, see Diez, Boehm-Davis, Holt, Pinney, Hansberger, & Schoppek, 2001). As the results in Table 2 indicate, the pilots' performance was unexpectedly poor. In the first scenario, all pilots violated the altitude restriction of 11000 feet at waypoint PANZE. The altitude restriction was much better met in the second scenario. However, four of five pilots initiated an early descent, which was unnecessary in Scenario 2 (and increases fuel consumption). The frequent mode changes commanded by the pilots are also remarkable.

	Scenario 1: Sea Isle approach			Scenario 2: Big Sur approach		
	early descent?	mode (sequence)	crosses PANZE at	early descent?	mode (sequence)	crosses ANJEE at
Good solution	yes	V/S or FLCH	11,000	no	VNAV	16,000
Subject 1	no	V/S	19,500	yes	V/S, VNAV V/S, VNAV	16,000
Subject 2	yes	VNAV	16,200	yes	VNAV	16,000
Subject 3	yes	FLCH, VNAV, FLCH, VNAV	13,000	yes	VNAV, FLCH, V/S, VNAV, FLCH, HOLD	16,200
Subject 4	no	VNAV	15,700	no	VNAV	16,000
Subject 5	yes	V/S, FLCH, VNAV, FLCH	11,000	yes	VNAV	16,000

Although this poor performance is troubling, we believe that it does not closely reflect the pilots' line performance and should be attributed, at least in part, to the setting of our study: The pilots reported that the simulation functioned reasonably well in simulating the responses of a real aircraft; however, there were two important differences between the simulated situation and a real cockpit. Our participants had to accomplish the flying task alone, whereas in reality, there is a distinct division of labor between two crew members. Second, the interface required keyboard and mouse entries rather than the use of knobs and dials. The combination of these factors likely influenced pilot performance.

4. Discussion

The process of designing the ACT-Fly model revealed a number of phenomena that appear to be common, or even typical for complex tasks, but for which there are no standard solutions in extant ACT-R models. We developed ad hoc solutions with no claim that they represent theories. For all these phenomena, theoretical solutions that account for a broad range of documented effects would advance both the efficiency of modeling complex tasks, and the scope of the ACT-R framework.

Deferring actions: In dynamic systems, effects of actions can unfold slowly. In these cases, checking the success of an action must be deferred, while other things are being done. For example, in VNAV mode, after entering a new reference value for alti-

tude, the pilot must wait until the next waypoint has been passed before deciding how to continue. The problem for modeling is how the deferred intention is remembered on time. To simulate intention memory, we used a mechanism that inhibits the chunks representing the deferred actions for a certain time. After each completion of a method, the model tried to retrieve deferred goals from memory, which fails as long as these are inhibited. After the inhibition has terminated, there is a chance to retrieve the chunks, but only for a limited time, after which they may be forgotten.

Expectations: One undesired type of event that can lead to errors is “automation surprise” (Sarter & Woods, 1995). It occurs when the behavior of the automation does not match the pilots’ expectation (e.g. when the aircraft suddenly levels off when the pilots expect it to continue the descent). We included two mechanisms to model expectations. One involves the retrieval of a chunk that represents a situation-action-situation sequence. The other models expectations through production rules that respond to outcomes not associated with specific behavior. The first mechanism requires a controlled action demanding central resources; the second mechanism does not really form expectations to be compared with the actual outcomes. Rather, it responds to unfamiliar situations. We think that both solutions do not sufficiently reflect the character of the process of forming expectations as an autonomous background mechanism, but since input and output can be connected exclusively through the central production selection process in ACT-R, such mechanisms cannot be implemented in principle. This strong claim of the ACT-R architecture might be reconsidered in future developments.

Estimation of time: We identified several processes that rely on estimation of time: the resumption of intentions, the periodic repetition of monitoring behaviors, and the decision to try a different method when one method fails after some repeated applications. We simulated time estimation simply by using the numerical output of the time function provided by ACT-R. Phenomena like the dependence of subjective time on workload could not be produced with this solution (Leuchter & Urbas, 2003). If one assumes internal timers whose signals are processed depending on, e.g., workload, there is again the question if this is done by the central production selection mechanism or by some other background processes.

Judging the results of our modeling effort, we can state that some of the objectives have been met, some have not. Following is a list of strengths of the model.

1. Errors of omission and errors of commission could be produced. The underlying dynamic is such that errors of omission are more likely under high working memory load (Schoppek, 2000).
2. We showed that a large fraction of (even flexible) behavior in the use of the flight management system can be explained by (predefined) procedures in the absence of declarative memory about the system.
3. The model can be used to test the effectiveness and flexibility of alternative procedures. In our model, the ACT-Fly method we developed for deciding what mode to use for a given descent profile proved effective in a range of scenarios. It would have been easy to develop and test alternative methods for making this decision by inserting a new method in its place.
4. The separation of general and domain specific parts allows for improvements in the general part with few changes to the specific part of the model. Further,

the procedural aspects of new tasks can be modeled quickly without needing to work at the general part.

5. We discovered mechanisms common for supervisory control tasks that are in principle difficult to implement in ACT-R (see above).

On the other hand, the model has a number of shortcomings:

1. Once the model has produced an error of omission (or commission), it is – unlike humans – not good at recovering from the error. In some simulations, the whole subsequent behavior of the model was disturbed by one error.
2. We did not succeed in modeling continuous mode awareness and automation surprise. The first is due to the strictly frame-oriented structure of the goal chunk in ACT-R (which corresponds largely to working memory in other frameworks). Once a piece of information is stored in one of the slots of the goal chunk, it can reliably be retrieved as long as the goal is active, or unless the slot is changed deliberately. The alternative of storing the information about the current mode in declarative memory rather than in the goal chunk does not produce the desired effects either, because in ACT-R, declarative memory can only be accessed in a controlled manner returning “conscious” feedback about the result of the retrieval attempt. Thus, we did not find a satisfactory solution for modeling inadvertent forgetting of mode information.
3. The model generally has poor situation awareness, supposedly because it has too little general knowledge about flying. We believe that real pilots classify (and identify) situations much more by similarity to known situations than by rules, as ACT-Fly does. It is possible that the procedure we developed for the descent decision is tailored too much to the support of decision-making and too little to the support of situation awareness. To test procedures more realistically, their implications for decisions and situation awareness should be judged.
4. We were able to build this model over a period of 10 months. This is a fairly long interval. However, it is shorter than what likely would have been required had we not been working with an existing cognitive architecture.

Most aspects of the task for which ACT-R did not provide enough constraints followed from the task's dynamic and the requirement to interleave subtasks during long time intervals. Although we were able to identify potential solutions to the identified problems, they represent only crude approximations to long-term theories that could be reused. Nonetheless, they can be regarded as hints how the scope of ACT-R could be extended to reasoning and action in more complex and dynamic environments.

5. Acknowledgments

This research has been supported by grants NAG 2-1289 from the NASA and 99-G-010 from the FAA. We thank Melanie Diez for her help in developing the scenarios for the simulator and for collecting empirical data from pilots.

6. References

- Altmann, E. M. (2000). Memory in chains: Modeling primacy and recency effects in memory for order. *Proceedings of the 22nd annual conference of the Cognitive Science Society* (pp. 31-36). Hillsdale, NJ: Erlbaum.
- Altmann, E. M. & Trafton, J. G. (2002). Memory for goals: An activation-based model. *Cognitive Science*, 26, 39-83.
- Anderson J.R. (1982). Acquisition of cognitive skill. *Psychological Review*, 89, 369 - 406.
- Anderson, J.R., & Lebiere, C. (1998). *Atomic components of thought*. Mahwah, NJ: Erlbaum.
- Anderson, J.R., Bothell, D., Lebiere, C. & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language*, 38, 341 -380.
- Anderson, J.R. & Reder, L.M. (1999). The fan effect: New Results and new theories. *Journal of Experimental Psychology: General*, 128, 186 -197.
- Byrne M.D., & Bovair S. (1997). A working memory model of a common procedural error. *Cognitive Science*, 21, 31 - 61.
- Card, S.K., Moran, T.P. & Newell, A. (1983). *The Psychology of Human - Computer Interaction*. Hillsdale, NJ: Erlbaum.
- Diez, M., Boehm-Davis, D.A., Holt, R.W., Pinney, M.E., Hansberger, J.T., Schoppek, W. (2001). *Tracking pilot interactions with flight management systems through eye movements*. Paper presented at the 11th International Symposium on Aviation Psychology, Columbus, Ohio.
[<http://hfac.gmu.edu/~mdiez/MelPubs/Tracking%20pilot%20interactions.PDF>]
- Holt, R. W., Hansberger, J., Chong, R., & Boehm-Davis, D. A. (2002). Modeling aviation crew interaction using a cognitive architecture. In *Proceedings of the 24th Annual Meeting of the Cognitive Science Society*, Fairfax, VA.
- Kieras, D. (1997). A guide to GOMS model usability evaluation using NGOMSL. In M. Helander, T. K. Landauer, & P. Prabhu (Eds.), *Handbook of Human-Computer Interaction*, (Second ed., pp. 733-766). New York: Elsevier.
- Lebiere C., & Wallach D. (2001). Sequence learning in the ACT-R cognitive architecture: Empirical analysis of a hybrid model. In C. L. Giles R. Sun (Eds.), *Sequence Learning: Paradigms, Algorithms, and Applications Lecture Notes in Computer Science* (pp. 188 - 212). Heidelberg: Springer.
- Lee F.J., & Anderson J.R. (2001). Does learning of a complex task have to be complex? A study in learning decomposition. *Cognitive Psychology*, 42, 267 - 316.
- Leuchter, S. & Urbas, L. (2003). Modeling dynamics and timing for operating human-machine systems. In F. Detje, D. Dörner, & H. Schaub (Eds.), *The Logic of Cognitive Systems. Proceedings of the Fifth International Conference on Cognitive Modeling* (pp. 279-280). Bamberg: Universitäts-Verlag.

- Rasmussen, J. (1986). *Information processing and human-machine interaction*. Amsterdam: North Holland.
- Reason, J. (1990). *Human error*. Cambridge: University Press.
- Rosenbloom, P. S., Laird, J. E., & Newell, A. (Eds.). (1992). *The SOAR Papers: Research on Integrated Intelligence*. Cambridge, MA: MIT Press.
- Salvucci, D. D. (2001). Predicting the effects of in-car interface use on driver performance: An integrated model approach. *International Journal of Human-Computer Studies*, 55, 85 - 107.
- Sarter N.B., & Woods D.D. (1992). Pilot interaction with cockpit automation I: Operational experiences with the flight management system. *International Journal of Aviation Psychology*, 2, 1 - 28.
- Sarter, N. B. & Woods, D. D. (1995). How in the world did we ever get into that mode? Mode error and awareness in supervisory control. *Human Factors*, 37, 5 -19.
- Schoppek, W. (2000). Learning and performance of sequential action under different workload conditions: An ACT-R model. In N. Taatgen; J. Aasman (eds.), *Proceedings of the Third International Conference on Cognitive Modeling* (pp. 295 – 296). Veenendaal: Universal Press.
- Schoppek W. (2002). Examples, rules, and strategies in the control of dynamic systems. *Cognitive Science Quarterly*, 2, 63 - 92.
- Taatgen N.A. (1999). A model of learning task-specific knowledge for a new task. *Proceedings of the 21th annual conference of the cognitive science society* (pp. 730 - 735). Mahwah, NJ: Erlbaum.
- Taatgen N.A., & Lee F.J. (2003). Production compilation: A simple mechanism to model complex skill acquisition. *Human Factors*, 45, 61 - 76.

Antizipative Modellierung des Benutzerverhaltens mit Hilfe von Aktionsvorhersage- Algorithmen

ALEXANDER KÜNZER, FRANK OHMANN & LUDGER SCHMIDT

Institut für Arbeitswissenschaft, RWTH Aachen

Schlüsselwörter: Adaptive Benutzungsschnittstellen, Aktionsvorhersage-Algorithmen, Benutzermodellierung

1. Einleitung

In den letzten Jahren ist der Funktionsumfang von Softwareanwendungen immer stärker angestiegen. Solche hochfunktionellen Anwendungen (HFA) sind sehr komplex geworden, weil sie die Bedürfnisse großer und heterogener Gruppen von Benutzern mit unterschiedlichem Kenntnisstand, unterschiedlicher Erfahrung und auch für verschiedene Aufgabenbereiche zufrieden stellen müssen (Fischer 2001). Trotzdem sind leicht benutzbare Benutzungsschnittstellen (UIs) nicht weit verbreitet – trotz der Berücksichtigung eines benutzerzentrierten Entwicklungsprozesses (DIN EN ISO 13407) oder allgemeiner Gestaltungskriterien für interaktive Softwaresysteme (DIN EN ISO 9241-10). Daher besteht weiterhin ein Bedarf an intelligenteren Benutzungsschnittstellen, die Novizen ebenso wie Experten unterstützen, ihre Aufgaben durchzuführen (Lau 1999).

Im Bereich der Benutzermodellierung wurden verschiedene Ansätze entwickelt, z. B. Kommandovorschlagsliste (Greenberg 1993, Davison & Hirsh 1998), intelligente Hilfe- oder Tutorsysteme (IHS/ITS, Encarnação und Stoev 1999) oder sogenannte „glaubwürdige“ Agenten (Maes 1994, Horvitz et al. 1998). Diese Systeme basieren alle auf Benutzermodellen, mit denen eine Vorhersage des Benutzerverhaltens möglich ist. Das bedeutet, dass der Computer die Interaktion des Benutzers mit der Anwendung beobachtet und so zu ermitteln versucht, welche Ziele und Aufgaben der Benutzer verfolgt (z. B. „Was macht er oder sie gerade?“, „Gibt es irgendwelche Probleme?“).

In diesem Artikel wird die Entwicklung eines Benutzermodells vorgestellt, welches auf einer Analyse von Aktionssequenzen beruht. Mit Hilfe dieses Ansatzes wird anschließend ein Unterstützungssystem für den Anwender realisiert. Dazu werden sogenannte Aktionsvorhersage-Algorithmen implementiert, die problemlos in – auch bereits bestehende – Anwendungen integriert werden können. Außerdem erlaubt die Verwendung eines solchen standardisierten Ansatzes den einfachen Vergleich verschiedener Aktionsvorhersage-Algorithmen mit unterschiedlicher Vorhersagequalität. Als Anwendungsfall dient eine selbstentwickelte multimodale Benutzungsschnittstelle zur Steuerung des 3D-Laserschweißens in zukünftigen Autonomen Produktionszellen (APZ). Verschiedene aus der Literatur bekannte aber auch eigene Implementierungen von Aktionsvorhersage-Algorithmen wurden anhand eines realen Interaktionsszenarios verglichen, um Unterschiede und den möglichen Nutzen solcher einfachen Unterstützungssysteme zu bewerten.

2. Hintergründe

2.1 Adaptive Benutzungsschnittstellen

Obwohl es in den letzten Jahrzehnten vielfältige Entwicklungen im Bereich der Mensch-Rechner-Interaktion gab, besitzen die meisten Software-Anwendungen immer noch starre Schnittstellen, welche die großen Unterschiede menschlicher Interaktionstypen und -stile nicht berücksichtigen. Anpassbare Benutzungsschnittstellen erlauben es dem Benutzer zwar, die Benutzungsschnittstelle nach seinen eigenen Bedürfnissen zu modifizieren, normalerweise besitzt der Benutzer jedoch nicht genügend Kenntnisse der Anwendung und über seine eigenen Bedürfnisse, so dass diese Art der Individualisierung meist nicht genutzt wird (Ross 2000). Einen besseren Ansatz bieten sogenannte adaptive Benutzerschnittstellen (AUI), welche die Bedürfnisse des Anwenders vorherzusehen versuchen, um die Benutzungsschnittstelle entsprechend anzupassen.

Adaptive User Interfaces (AUI) können als eine Art „intelligente“ Schnittstelle angesehen werden, bei der die Benutzungsschnittstelle mit Techniken der Künstlichen Intelligenz (KI) versucht, die Ziele und Bedürfnisse des Anwenders zu antizipieren und sich diesen anzupassen (Dieterich et al. 1993). Eine andere Definition von AUIs stammt von Langley (1997).

“An adaptive user interface is a software artifact that improves its ability to interact with a user by constructing a user model based on partial experience with that user.”

Ein Benutzermodell kann eingesetzt werden, um Ziele und Präferenzen des Benutzers vorherzusagen oder Verhaltensmuster zu entdecken, z.B. zur spekulativen Befehlsausführung, Eingabe vervollständigung, Ausführungsbeschleunigung oder Unterstützung. Beispiele hierzu sind z. B. das Lumière-Projekt (Horvitz et al. 1998), welches später kommerziell in den MS Office Produkten eingesetzt wurde. Ein anderes Beispiel stammt von Davison und Hirsh (1998), die ein System zur Vorhersage von Unix-Kommandos entwickelten, das von Korvemaker und Greiner (2000) auf vollständige Unix-Kommandos inkl. Parameter ausgedehnt wurde.

2.2 Generische Benutzermodellierung durch Beobachtung von Handlungssequenzen

Intelligente Benutzungsschnittstellen wurden bereits in vielen Projekten erforscht. Im jeweiligen Anwendungsumfeld erreichten sie dabei oftmals beeindruckende Ergebnisse. Jedoch führte die Übertragung auf andere Anwendungsfelder zu neuen Herausforderungen (Gorniak & Pool 2000). Viele der manuell modellierten Ansätze (z. B. Aufgaben- und Benutzermodelle) sind zu aufwändig und unflexibel (Carberry 2001), wie z. B. im Lumière Projekt, in dem Experten das Benutzerverhalten mit Bayes'schen Netzen modellierten oder wie in VAMPIRE ("Visual Model-based Pick-And-Place InterFace Editor", Eisenstein and Rich 2002), bei welchem Aufgaben feingranular in Form von Makros beschrieben werden müssen. Es besteht daher ein Bedarf an generischen Ansätzen, die von der Domäne unabhängig sind und allein aus dem bisherigen Verhalten des Benutzers lernen können (Kobsa 2001).

In dieser Veröffentlichung werden verschiedene Ansätze betrachtet, die weitgehend unabhängig von einer speziellen Domäne oder Anwendung sind. Zu diesem Zweck werden Benutzeraktionen mit Domänen- bzw. Aufgabenbezug beobachtet, d. h. die Aktionen sind im Rahmen des semiotischen Modells (Foley et al. 1990) auf der semantischen Ebene anzusiedeln. Diese Verdichtung der Aktionen reduziert die tatsächlich zu berücksichtigenden Aktionen auf diejenigen mit Aufgabenbezug, welche dann im weiteren als syntaktische Interaktionssequenz kontextfrei weiter betrachtet werden (Hilbert & Redmiles 1999). In diesen Ansatz kann auch das Zustandsmodell der Anwendung berücksichtigt werden, in dem die beobachtete Benutzerinteraktion durch entsprechende Ereignisse der Anwendung angereichert werden.

Ursprung dieser Aktionen sind die physikalische Ereignisse, welche über die Eingabegeräte zeitlich verteilt, nacheinander in das Betriebssystem gelangen (vgl. Abbildung 1). Hier führen diese Ereignisse zur entsprechenden Ereignissen des Betriebssystems und werden dann über eine abstrakte Interaktionsebene bis möglicherweise zur Ebene mit Domänen- bzw. Aufgabenbezug weitergeleitet. Allerdings kann dabei auf jeder Ebene ein Filterprozess stattfinden, falls das Ereignis nicht auf höheren Ebenen benötigt wird (bspw. müssen die meisten Anwendungen keine Informationen über Mauszeigerbewegungen erhalten). Der linke Aktionsstrang in Abbildung 1 zeigt eine Aktion welche keine Auswirkungen auf die eigentliche Aufgabe besitzt (z. B. kann es sich um das Zoomen einer Abbildung handeln) und daher vor der obersten Ebene herausgefiltert wird.

Auch die Benutzungsschnittstelle kann wie im rechten Aktionsstrang Ereignisse auslösen, um etwa eine Nachricht anzuzeigen. Neben dem Filtern besteht auch die Möglichkeit der Aggregation von Aktionen. Dadurch können auch komplexere Aktionsmuster oder -sequenzen auf die für die Beobachtung wesentlichen Aktionen reduziert bzw. umgewandelt werden. Im Beispiel in Abbildung 1 ist das Drucken über einen Druckdialog, welcher dann mit OK bestätigt wird gezeigt – alternativ könnte die resultierende Aktion `Document.Print` auch über eine Werkzeugleiste direkt ausgelöst werden.

Basierend auf diesen Aktionssequenzen ist es dann möglich, mit adäquaten Benutzermodellen die nächsten möglichen Aktionen des Benutzers vorherzusagen. Einerseits können diese Vorhersagen das Interaktionsverhalten von Experten berücksichtigen, so dass ein Unterstützungssystem für Novizen realisiert werden kann. Andererseits ist es aber auch möglich, die Interessen des Anwenders selbst als Grundlage zu

verwenden, um Shortcuts vorzuschlagen bzw. eine individuelle Anpassung der Oberfläche durchzuführen. Eine Kombination beider Ansätze ist ebenfalls möglich.

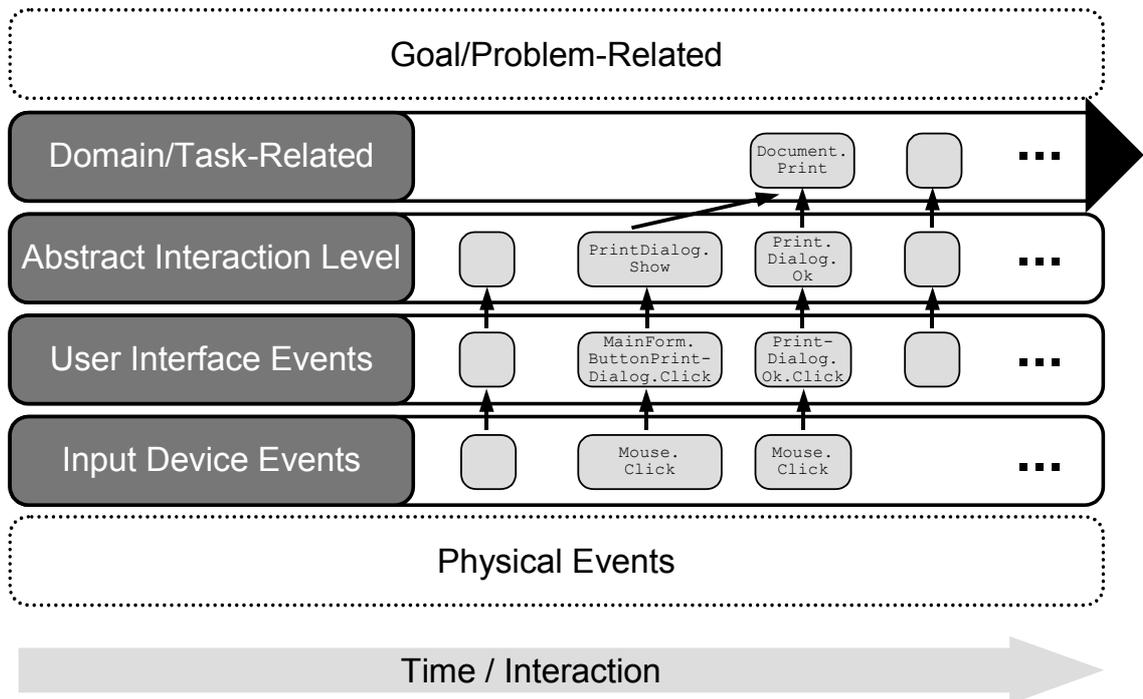


Abbildung 1: Ebenen der Abstraktion in Benutzerschnittstellen nach Hilbert und Redmiles (1999) mit einem Filter- und Sammlungsprozess zur Ermittlung domänen-/aufgabenorientierter Aktionen (kleine Boxen = Aktionen).

2.3 Aktionsvorhersage-Algorithmen (AVAs)

Die eigentliche Vorhersage der Benutzeraktionen wird mit Hilfe eines Aktionsvorhersage-Algorithmus (AVA, engl. Action Prediction Algorithm) durchgeführt. Dabei werden die möglichen nächsten Aktionen samt der Wahrscheinlichkeiten prognostiziert, indem die vorherigen Aktionen als Eingabe dienen. Abbildung 2 zeigt die Funktionsweise eines AVAs schematisch. AVAs können in verschiedener Weise implementiert werden, z. B. durch (Hidden) Markov Modelle (Cook & Wolf. 1998; Zukerman et al. 1999) oder Bayes'sche Netze (Horvitz et al. 1998; Schlick et al. 2002). Nachteilig ist dabei häufig, dass viele Lernfälle benötigt werden, um die Modelle adäquat anlernen zu können (Fischer 2001). Dies kann problematisch sein, falls für eine Anwendung entsprechend viele Trainingsdaten nur schwer zu erhalten sind.

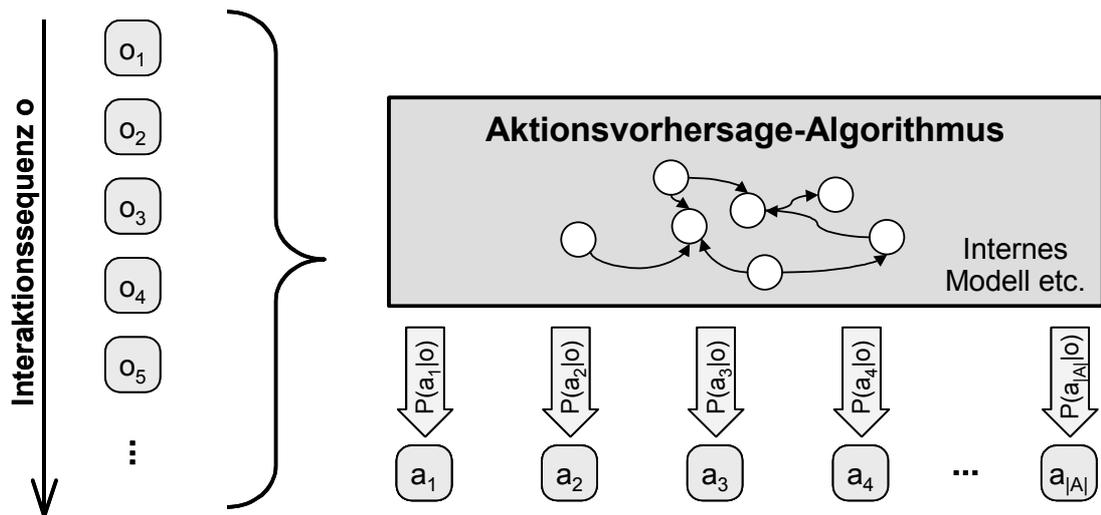


Abbildung 2: Schematische Darstellung eines Aktionsvorhersage-Algorithmus.

Sun (2000) gibt die folgende Formulierung für Sequenz-Vorhersagen: Sei $s_i \in S$ ein Sequenz-Element, dann kann das Problem beschrieben werden als eine Wahrscheinlichkeitsverteilung für die Elemente der Sequenz mit $p(s_{j+1} | s_i, s_{i+1}, \dots, s_j)$, wobei $1 \leq i \leq j < \infty$. Wenn $i=1$ ist, basiert die Vorhersage auf allen vorhergehenden Elementen. Gemäß der Idee eines idealen Online-Lern-Algorithmus (IOLA) von Davison und Hirsh (1998) und eigenen Erfahrungen auf diesem Gebiet kann ein AVA wie in Formel 1 angegeben definiert werden, wobei A^* die Menge aller endlichen Aktionssequenzen über dem endlichen Alphabet A , also der Menge aller möglichen Aktionen der Anwendung, ist.

$$\phi: A^* \rightarrow R^{|A|} : o_{1:m} \mapsto (p_1, p_2, \dots, p_{|A|}) \quad (1)$$

mit $p_i = P(a_i | o_{1:m})$ und $a_{1 \leq i \leq |A|}, o_{1 \leq j \leq m} \in A$

3. AVA-basierte Benutzer-Unterstützungssysteme

3.1 Vorgehen zur AVA-basierten Unterstützung

Der Ablauf der Unterstützung mithilfe eines AVA ist in Abbildung 3 visualisiert und ergibt sich direkt aus der Funktionsweise eines AVA. Basierend auf den beobachtbaren Aktionen des Benutzers wird zuerst der Filter- und Aggregationsprozess erforderlich, um die abstrakten Aktionen auf der semantischen Ebene zu aggregieren. Diese Interaktionssequenz stellt dann die Eingabe für den verwendeten AVA dar und kann mögliche Folgeaktionen samt Wahrscheinlichkeiten vorhersagen, die dann zur adaptiven Anpassung der Anwendung bzw. der Unterstützungskomponente genutzt werden.

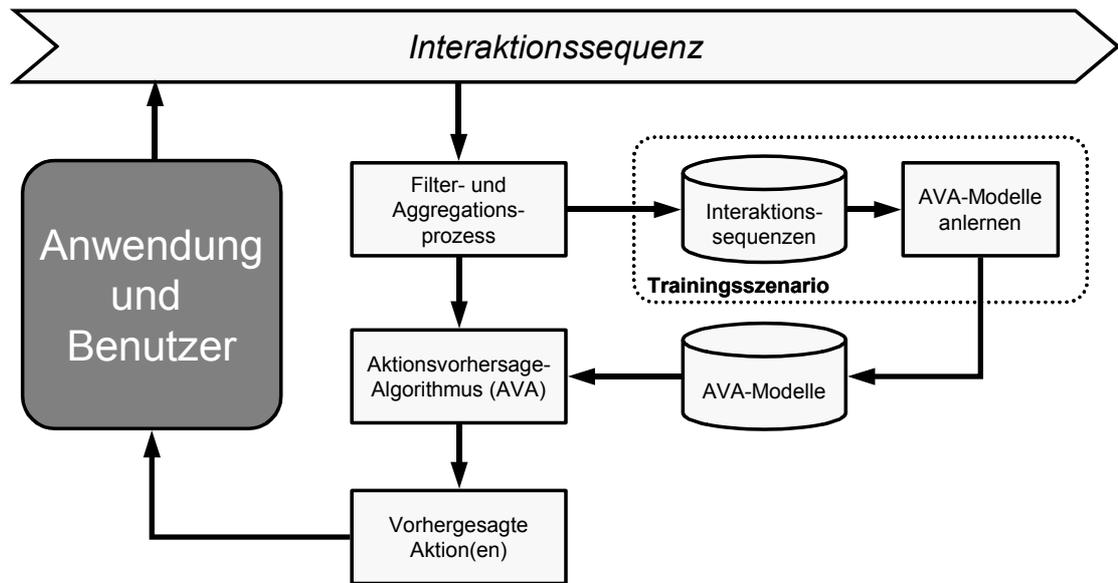


Abbildung 3: Schematisches Ablaufdiagramm zur adaptiven Unterstützung mittels eines AVA

Im Trainingsfall können mit demselben Mechanismus auch Interaktionssequenzen aufgezeichnet und gesammelt werden, die dann zum Anlernen des AVA-Modells genutzt werden.

3.2 Adaptive Hilfe- und adaptive Tutorsysteme

Es wurden zwei Unterstützungssysteme entwickelt, die auf AVAs basieren. Abbildung 4 zeigt eine Bildschirmdarstellung des adaptiven Tutors (links) und der adaptiven Hilfe (rechts). Der adaptive Tutor ermöglicht dem Benutzer eine aktivere Interaktion, während die adaptive Hilfe passiv gestaltet ist. In beiden Unterstützungssystemen lassen sich aber Statusinformationen anzeigen (1), welche dem Benutzer aktuelle Vorgänge erläutern können. Der adaptive Tutor zeigt die vorgeschlagenen Aktionen (2) samt entsprechendem Hilfetext an. Die Hyperlinks erlauben es dem Benutzer, die angezeigte Aktion direkt über den Tutor auszuführen. Das adaptive Hilfesystem (rechts) hat ähnliche Elemente, erlaubt aber keine direkte Interaktion.

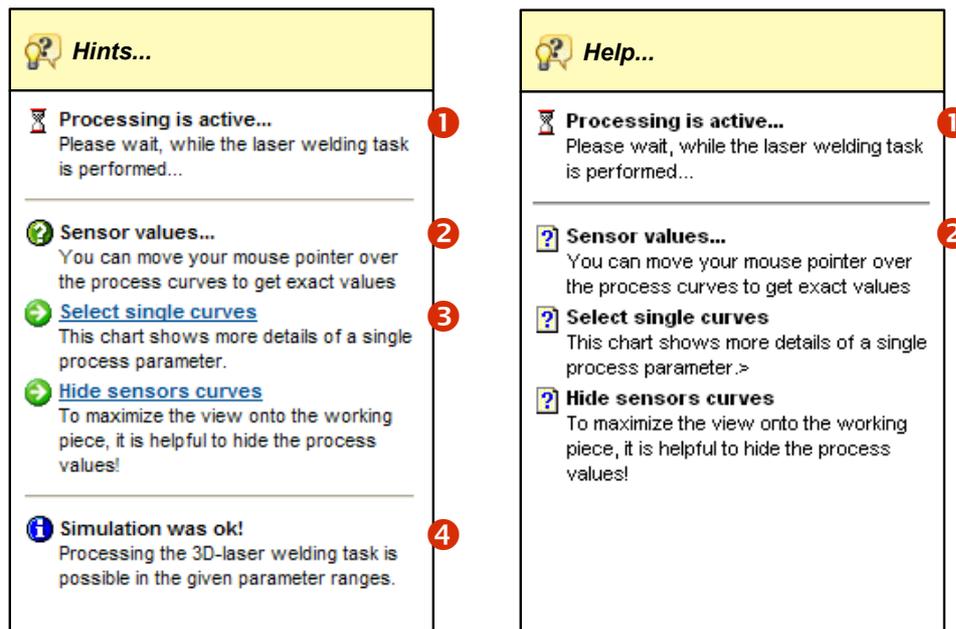


Abbildung 4: Beispiel des adaptiven Tutors und der adaptiven Hilfe.

Sowohl die adaptive Hilfe als auch der adaptive Tutor zeigen die vorgeschlagenen Aktionen und Hinweise basierend auf dem bisherigen Verhalten des Benutzers an. Das initiale Aufgabenmodell der Anwendung wird aus der Beobachtung von erfahrenen Benutzern konstruiert und enthält so deren Wissen bzgl. der Applikationsnutzung. Für den Fall, dass das Interaktionsverhalten eines einzelnen Benutzers zu Grunde gelegt wird, kann die Hilfe bzw. der Tutor lernen, persönliche Interessen und Vorlieben des Benutzers zu unterstützen, so dass Eingabevervollständigung, Shortcuts aber auch Unterstützung realisierbar ist. Eine spekulative Befehlsausführung, d. h. der Computer führt die Aktionen des Benutzers pro-aktiv aus, wird nicht verfolgt, da absolut korrekte Vorhersagen nicht garantiert werden können.

Der generische Ansatz erlaubt eine einfache – auch nachträgliche – Integration in vorhandene Anwendungen. Eine explizite Modellierung des Domänen- bzw. Anwendungswissens ist nicht erforderlich. Die Anpassung basiert auf einem AVA, der die wahrscheinlichsten Aktionen vorhersagt. Somit ist die Hilfe / der Tutor unabhängig von einer speziellen Implementierung und erlaubt es, den verwendeten AVA auszutauschen, z. B. aus Leistungsgründen. Dadurch kann sowohl der adaptive Tutor als auch die adaptive Hilfe in beliebige Anwendungen integriert werden. Ansonsten können die Vorhersagen auch benutzt werden, um andere Unterstützungsarten zu implementieren.

Um den Nutzen des adaptiven Hilfesystems zu erhöhen, werden nicht nur die vorgeschlagenen Aktionen, sondern auch einige zusätzliche Elementtypen verwendet:

Konzepte. Die Ansammlung verschiedener (auch identischer) Aktionen (z.B. Bewegen und Positionieren eines 3D-Objektes) kann durch ein Konzept wie in Abb. 3 (2) zusammengefasst werden. Konzepte werden ebenfalls bei der Prädiktion von Aktionen berücksichtigt.

Status. Statusmitteilungen (z.B. „Aufgabe wird erledigt“) werden zusätzlich im Tutor angezeigt (siehe Abbildung 4 (1)). Im Falle der intensiven Nutzung des Tutors ist

dadurch sichergestellt, dass Mitteilungen der Anwendung vom Benutzer nicht übersehen werden.

Information. Informationselemente werden benutzt, um den Benutzer über wichtige Situationen innerhalb der Anwendung zu informieren (ähnlich wie bei den Status-Elementen), wie in Abb. 4 (4). Informationselemente werden aber im Gegensatz zu Statusmitteilungen bei der Prognose weiterer Aktionen berücksichtigt.

3.3 Integrationsanforderungen für aktionsbasierte Unterstützungssysteme

Eine Integration dieses generischen Ansatzes für die adaptive Hilfe bzw. den adaptiven Tutor ist einfach und erfordert das Einfügen des Vorhersagemoduls sowie einer Modifikation der vorhandenen Applikation in folgender Hinsicht:

Aktionen. Alle Aktionen, welche relevante Aspekte der Anwendung abdecken, müssen definiert werden. Dazu müssen Bezeichnungen und Hinweise angegeben werden, welche in der Hilfe bzw. dem Tutor angezeigt werden. Heutzutage unterstützen Betriebs- und Programmiersysteme Ereignis- (z. B. MS Windows) und Aktionsbehandlungen (z.B. Borland's Delphi), auf die zurückgegriffen werden kann.

Logging. Die Anwendung muss fähig sein, alle definierten Aktionen aufzuzeichnen. Dadurch werden Interaktionssequenzen gewonnen, welche die Basis zur Vorhersage weiterer Aktionen bilden.

Remoting. Um dem Benutzer die Navigation über den Tutor zu ermöglichen, ist es nötig, dass die Funktionen der Anwendung durch den Tutor aufgerufen werden können. Eine Makrofähigkeit bzw. ein Nachrichtensystem (z. B. wie in Windows) können dazu verwendet werden.

3.4 Einsatzbereiche für AVA-basierte Unterstützungsfunktionen

Ein AVA-basierter Unterstützungsansatz bietet sich vor allem für Mensch-Maschine-Systeme an, welche einen hohen Anteil an prozessorientierter Navigation aufweisen. Die im folgenden vorgestellte Anwendung ACTIVE-UI erfüllt diese Anforderungen, genauso wie bspw. die Interaktion auf Webseiten. Mensch-Maschine-Systeme mit hohem inhaltlichen Ausrichtung und geringerer Frequenz der Interaktionsereignisse sind hingegen für den AVA-Ansatz weniger geeignet (z. B. Konstruktion via CAD-System, Operator-Schnittstellen). Allerdings kann generell der Zustandsraum eines Mensch-Maschine-Systems durch Berücksichtigung entsprechender Systemereignisse auch bei AVA-Ansätzen berücksichtigt werden.

4. Implementierung der Aktionsvorhersage-Algorithmen

4.1 Aus der Literatur bekannte Aktionsvorhersage-Algorithmen

Um einen geeigneten AVA für das adaptive Unterstützungssystem auszuwählen, wurden verschiedene Modelle mit dem Ziel implementiert, Unterschiede und Nutzen vergleichen zu können. Markov-basierte Algorithmen haben eine geringe Vorhersagegüte und benötigen umfangreiche Trainingsdatenbestände (Luczak et al. 2001, Künzer & Luczak 2003).

4.1.1 Der IPAM-Algorithmus

Der IPAM-Algorithmus (Incremental Probabilistic Action Modeling) wurde von Davison und Hirsh (1998) vorgestellt. Es ist ein einfacher AVA, der aber gute Ergebnisse liefert und dazu ein einfaches Markov-Modell verwendet, das durch ein spezielles Online-Lernverfahren aktualisiert wird. Der Parameter alpha des IPAM-Algorithmus beschreibt die Gewichtung der zuletzt aufgetretenen Aktionen in den Vorhersagen und wird von Davison und Hirsh zwischen 0,6 und 0,8 empfohlen. Während der ursprüngliche IPAM-Algorithmus auf einem Markov-Modell erster Ordnung beruht, haben die Autoren IPAM für höhere Ordnungen erweitert (genannt IPAM n –Algorithmus).

4.1.2 Die PPM-Algorithmen

Der PPM-Algorithmus (Prediction by Partial Matching) entstand ursprünglich aus einem Kompressionsalgorithmus (Ziv & Lempel 1977), der finite Kontexte benutzt und auch zur Vorhersage von Benutzeraktionen verwendet werden kann (Cleary & Witten 1984). Ein PPM-Algorithmus der Ordnung n besteht aus n+1 Markov-Vorhersagern. PPM+ ist eine Erweiterung von PPM, die zusätzlich einen Startvektor für das erste Ereignis einer Sequenz verwendet, anstatt dort die absoluten Häufigkeiten zu berücksichtigen. PPMU+ benutzt eine spezielle Anlerntechnik, bei der nur die längste gefundene Sequenz beim Anlernen aktualisiert wird (Howard und Vitter 1994). PPM* besitzt keine maximal vorgegebene Markov-Ordnung und wurde von Cleary et al. (1995) vorgestellt. PPMC+ berücksichtigt bei der Prognose auch kürzere Sequenzen als die maximal gefundene und wurde von Moffat (1990) präsentiert.

4.2 Neuentwickelte Aktionsvorhersage-Algorithmen

4.2.1 Der Aktionsvorhersage-Algorithmus LEV

Es wurde ein eigener AVA entwickelt, der auf der Ähnlichkeit von Sequenzen basiert. Als Ähnlichkeitsmaß wird die Levenshtein-Distanz (Levenshtein 1966) herangezogen. Da der vorliegende AVA dieses verbreitete Ähnlichkeitsmaß verwendet, wurde er „LEV-Algorithmus“ genannt. Dabei ist die Grundidee, auch dann gültige Vorhersagen zu treffen, wenn der Benutzer Fehler oder unnötige Schritte während der Interaktion im existierenden System macht. Dazu versucht der Algorithmus zunächst eine Vorhersage aufgrund von exakten Übereinstimmungen mit Sequenzen aus den Trainingssequenzen zu treffen. Wenn keine exakten Übereinstimmungen existieren, versucht der Algorithmus Übereinstimmungen mit der Levenshtein-Distanz 1 zu finden, dann mit Distanz 2 usw. Für den Fall, dass exakte Übereinstimmungen vorhanden sind, ist das vom Algorithmus gelieferte Ergebnis identisch mit dem eines Markov-Modells der vorgegebenen Ordnung. Ansonsten wird die Suche ausgeweitet auf die Suche nach ähnlichen Treffern. Abbildung 5 zeigt den Pseudo-Quellcode des LEV-Algorithmus.

```

function SeqCount(o: Sequence; LevDist: Integer): Integer;
begin
  Result := 0;
  for each o'_{Subseqs of len|o|} in O do
    if CalcLevenshteinDistance(o',o) = LevDist then
      inc(Result);
  end;

function CalcPrediction(o: Sequence): Vector;
begin
  Result := ZeroVector;
  PredLen := Min(MarkovOrder, |o|);
  for i := 0 to PredLen + 1 do
    begin
      for each a in A do
        Result[a] := SeqCount(o_{|o|-PredLen+1:|o|}a, i);
        if Result <> ZeroVector then break;
      end;
      Normalize(Result);
    end;
  end;
end;

```

Abbildung 5: Pseudo-Quellcode für den Levenshtein-Algorithmus.

4.2.2 Der Aktionsvorhersage- Algorithmus KO

Nach der Handlungsregulationstheorie (z. B. Hacker 1978) finden Benutzerinteraktionen in hierarchisch-sequentiell strukturierten Handlungssequenzen statt (Abb. 5). Daher wurde ein weiterer AVA entwickelt, welcher als „KO-Algorithmus“ bezeichnet wurde (da er andere AVAs in Bezug auf die Vorhersagequalität AVAs „ausschaltete“). Der KO-Algorithmus berücksichtigt nicht nur die längste gefundene Sequenz zur Prognose (wie bspw. PPM), sondern zieht dazu auch kürzere Sequenzen heran. Besteht eine beobachtete Sequenz wie im Beispiel in Abbildung 6 aus zwei Untersequenzen H_2H_1 , so gibt es drei mögliche Vorhersagen in der Trainingssequenz: H_2H_1 aber auch zwei Mal die Untersequenzen H_1 . Dadurch kann der KO-Algorithmus die Aktion Y zusätzlich zur Aktion X vorhersagen.

Der KO-Algorithmus kombiniert den Ansatz der mindestens vorkommenden Untersequenzen (z. B. Mannila et al. 1997) mit einer Nearest-Neighborhood-Technik, wobei alle geeigneten Teilsequenzen genutzt werden, um Vorhersagen zu treffen. Dieser selbstentwickelte AVA benutzt zusätzlich Gewichte $w(i)$, abhängig von der Länge der zu vergleichenden Teilsequenz. Gute Ergebnisse wurden dabei mit der Gewichtungsfunktion $w(i) = i^{19}$ erzielt. Abbildung 7 zeigt den Pseudocode für den KO-Algorithmus.

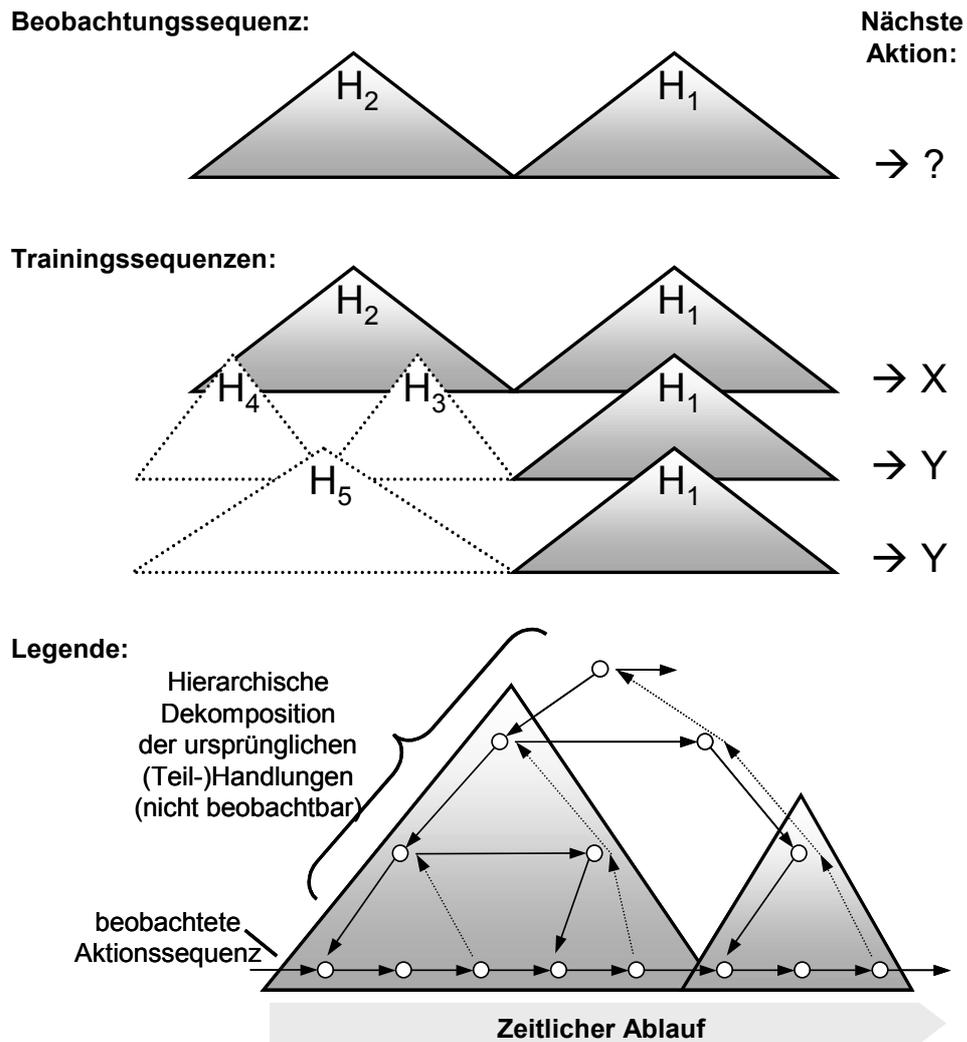


Abbildung 6: Beispiel für die Vorhersage einer Aktion basierend auf hierarchisch-sequentiellen Handlungssequenzen nach Hacker (1978)¹.

Eine weiterführende Betrachtung des KO-Algorithmus wurde von Künzer (2004) durchgeführt, wobei unterschiedliche Gewichtungsfunktionen und – durch eine verbesserte Implementierung – auch höhere Exponenten im Rahmen einer Grenzwertbetrachtung untersucht wurden. Dabei erwies sich der KO-Algorithmus ein als universeller AVA-Ansatz, wobei allerdings die ursprüngliche Hypothese in Bezug auf die Berücksichtigung unterschiedlicher Sequenzlängen aber verworfen werden musste. Vielmehr konvergiert der KO-Ansatz gegen eine neue Variante des PPM-Ansatzes, bei welchem die höheren Vorhersagewahrscheinlichkeiten auf den initialen Filterprozess der Mindestfrequenzen zurückzuführen sind. Die über den KO-Algorithmus hergeleitete neue Variante des PPM*-Algorithmus mit Pruning wird daher als PPMP* bezeichnet. Es zeigte sich aber, dass die Wahl der Gewichtungsfunktion $w(i) = i^{19}$ Vorhersagewahrscheinlichkeiten erzielt, welche annähernd den maximalen, vom PPMP*-Algorithmus zu erzielenden Werten, entspricht.

¹ Die hierarchisch-sequentiellen Handlungsstrukturen sind im Beispiel durch die Dreiecke nur angedeutet. Während im zeitlichen Ablauf lediglich eine Aktionssequenz zu beobachten ist (Legende, unten), stecken dahinter eigentlich die hierarchisch-verfeinerten Pläne. Auch die Handlungen (Dreiecken) können sich ihrerseits auch aus Teil-Handlungen zusammensetzen.

```

function SubseqCount(o: Sequence): integer;
begin
  Result := 0;
  for each o' in O do
    Result := Result + CountSubseqsInSeq(o, o');
    if Result < MinimumFrequency then Result := 0;
  end;

function CalcPrediction(o: Sequence): Vector;
begin
  Result := ZeroVector;
  for each a in A do
    begin
      Result[a] := SubseqCount(a) * w(0);
      for i := 1 to min(|o|, SearchDepth - 1) do
        Result[a] := Result[a] + SubseqCount(o|o|-i+1:|o|a) * w(i);
      end;
      Normalize(Result);
    end;
  end;
end;

```

Abbildung 7: Pseudocode zur Vorhersage mit dem KO-Algorithmus.

5. Voruntersuchung: Früher Prototyp von ACTIVE-UI

5.1 Fallstudie

Als Fallstudie diente eine selbstentwickelte multimodale Benutzungsschnittstelle namens ACTIVE-UI (“Autonomous Production Cells’ Multimodal and Adaptive – User Interface”, Schlick 2000) zur Überwachung von 3D-Laserschweißvorgängen in künftigen Autonomen Produktionszellen. Damit ACTIVE-UI sowohl für Novizen als auch für Experten besser benutzbar ist, soll ACTIVE-UI durch ein adaptives Hilfesystem auf AVA-Basis ergänzt werden. Um hierzu von realen Benutzern typische Interaktionssequenzen zu sammeln, wurde ein Experiment konzipiert, bei dem 30 Benutzer mit Erfahrung im Umgang mit ACTIVE-UI ohne Tutor ein typisches Interaktionsszenario zu bearbeiten hatten (Künzer et al. 2001). Als Interaktionsfall sollte ein Schweißauftrag durchgeführt werden (Konfigurieren der Sensoren, Auswahl eines numerischen Kontrollprogramms, Simulation und Prozessanalyse, sowie Überprüfen von wichtigen Prozesswerten).

5.2 Anlernen der Aktionsvorhersage-Algorithmen

Die Algorithmen benötigen verschiedene Parameter, u. a. die Ordnung der Markoketten. Die Auswahl dieser Werte kann durch Kreuzvalidierung und einer Anzahl gegebener Trainingsfälle systematisch getestet werden. Kommen neue Anwendungsfälle hinzu, können die Parameter später erneut geschätzt werden, so dass ständig eine Anpassung an das aktuelle Benutzerverhalten erfolgt. In der hier vorgestellten Untersuchung wurden immer die optimalen Parameter für jeden Algorithmus verwendet.

5.3 Verfahren

Die aufgezeichneten Interaktionssequenzen von 30 Benutzern wurden verwendet, um verschiedene AVAs als unabhängige Variable zu bewerten. Die Vorhersagewahrscheinlichkeit als abhängige Variable wurde durch die mittlere Vorhersagewahrscheinlichkeit (Mean Prediction Probability, MPP) operationalisiert, wie sie in Formel 2 und 3 angegeben sind (Künzer et al. 2001).

Zusätzlich wurde die mittlere Prognosegenauigkeit (Mean Prediction Accuracy, MPA) benutzt, die ähnlich berechnet wird, nur dass korrekte Vorhersagen mit 1 bewertet werden und falsche Vorhersagen mit 0. Obwohl die MPA weniger aussagekräftig ist (Albrecht et al. 1998), wird sie doch in anderen Publikationen zu diesem Thema oft herangezogen.

$$PP(o, \phi) = \frac{1}{|o|} \sum_{i=1}^{|o|} P_{\phi}(o_i | o_{1:i-1}) \quad (2)$$

$$MPP(O, \phi) = \frac{1}{|O|} \sum_{o \in O} PP(o, \phi) \quad (3)$$

Sowohl MPP (und bei den anderen Untersuchungen auch die MPA) werden mit einer sechsfachen Kreuzvalidierung berechnet, wobei alle Interaktionssequenzen auf sechs Gruppen verteilt werden. Fünf dieser Gruppen wurden als Trainingsbasis benutzt, um den AVA anzulernen, die sechste Gruppe dient dann zur Evaluation des AVAs. Um den Effekt einer ungünstigen Verteilung der einzelnen Sequenzen auf die sechs Gruppen zu vermeiden, werden die Berechnungen mit insgesamt fünf Replikationen wiederholt. Jede dieser fünf Replikationen enthält dabei eine zufällige Permutation der Trainingsbasis.

5.4 Ergebnisse

Die durchschnittliche Dauer der Aufgabenbearbeitung betrug 346 Sekunden (zwischen 208 und 598 Sekunden). Während dieser Zeit benötigten die Benutzer 14 bis 29 Schritte (im Mittel 23) und nutzten 14 bis 26 verschiedene Aktionen (durchschnittlich 22) von insgesamt 34 vorhandenen.

Anschließend wird eine einfaktorielle ANOVA für die Variable *MPP* durchgeführt. Die Nullhypothese ist dabei, dass die *MPP* der stochastischen Modelle gleich ist. Die alternative Hypothese postuliert einen signifikanten Unterschied von wenigstens zwei AVAs ($\alpha_{ANOVA} = 0.05$). Da die Nullhypothese verworfen werden musste, wurden mit dem Post-hoc Student-Newman-Keul Test die signifikanten Unterschiede zwischen den AVAs überprüft ($\alpha_{Posthoc} = 0.05$).

Die Ergebnisse der Untersuchung sind in Abbildung 8 dargestellt. Die 95%-Konfidenzintervalle zeigen eine überlegene *MPP* von KO 3/19 (*MPP*=0,5464), vor PPM+ 4 (*MPP*=0,5305) und PPMU+ 4 (*MPP*=0,5233). Der PPMC+ 4 Algorithmus und alle IPAM-Implementierungen hatten eine niedrigere *MPP*.

Die einfaktorielle ANOVA ergab einen F-Wert von $F_{ANOVA}=65,093$ ($df=9$). Da der Wahrscheinlichkeitswert des F-Tests die kritische Marke von $p_{ANOVA}<0,001$ unterschreitet, wurde die Nullhypothese verworfen. Der Post-hoc Test ergab, dass die Unterschiede zwischen KO 3/19, PPM+ 4 und PPMU+ 4 nicht signifikant sind. Alle homogenen Untergruppen sind in Abbildung 8 eingezeichnet. Die *MPP* des neuen LEV-Algorithmus ist nicht unter den höchsten Werten, hatte aber dennoch einen höheren Wert als die IPAM- und PPMC+ Algorithmen.

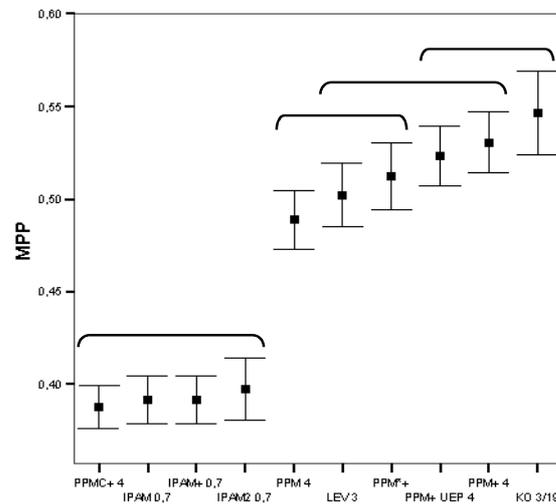


Abbildung 8: 95% Konfidenzintervalle von *MPP* mit signifikanten homogenen Untergruppen ($\alpha_{\text{Posthoc}} = 0,05$).

6. Voruntersuchung: Greenberg Referenzdateien

Um diese Ergebnisse besser mit anderen Publikationen vergleichen zu können, wurde die Vorhersagewahrscheinlichkeit und -genauigkeit der einzelnen Algorithmen ebenfalls anhand der Greenberg-Dateien getestet (Greenberg 1988). Die acht von Greenberg zur Verfügung gestellten Dateien enthalten jeweils zwei Dateien aus vier unterschiedlichen Gruppen. Diese Dateien wurden in einzelne Sessions unterteilt, die Kommandoparameter entfernt und durch einheitliche Aktions-IDs innerhalb jeder Datei umkodiert.

Die Auswertung erfolgte wieder über die bereits beschriebene sechsfache Kreuzvalidierung. Tabelle 1 zeigt die *MPP* und Tabelle 2 die *MPA* dieser Interaktionssequenzen. Zusätzlich wurde ebenfalls für jede Datei eine einfaktorielle ANOVA gerechnet. In den Tabellen wurde die signifikant beste homogene Untergruppe gemäß des Student-Newman-Keul Tests mit einem Sternchen (*) versehen, während der höchste Wert jeder Zeile fett gedruckt ist.

Der KO3/18-Algorithmus war immer in der Untergruppe mit den besten *MPPs* zu finden, und in sieben von neun Fällen erreichte er sogar die höchste *MPP* von allen Algorithmen. Verglichen mit den Ergebnissen von Abbildung 8 kamen PPM+4 und PPMU+4 nur in vier bzw. drei von neun Fällen in die Spitzengruppe. Das zweite interessante Ergebnis ist, dass die *MPA*, die hauptsächlich in anderen Publikationen verwendet wurde, zu ähnlichen Ergebnissen führt. Wieder kam KO3/19 in acht von neun Fällen in die Spitzengruppe.

Tabelle 1: Vergleich der Vorhersagewahrscheinlichkeiten (MPP) für unterschiedliche AVAs bei den Greenberg-Referenzdateien.

	o	A	IPAM/0,7	IPAM2/0,7	KO3/18	LEV3	PPM4	PPM+4	PPMU+4	PPMC4	PPM*	PPM*+
novice-programmers-15	71	14	0,3283	0,3724	*0,4653	0,4262	0,4247	*0,4381	0,4309	0,3854	0,4277	*0,4411
novice-programmers-20	54	18	*0,2596	*0,2602	*0,2972	*0,2906	*0,2984	*0,2953	*0,2863	*0,2682	*0,2944	*0,2913
experienced-programmers-14	101	51	0,1428	0,1423	*0,2075	0,1808	0,168	0,1879	0,1794	0,1279	0,1617	0,1815
experienced-programmers-28	438	70	0,2032	0,2123	*0,2485	0,2211	0,1778	0,2272	0,2235	0,1445	0,1730	0,2225
computer-scientists-11	37	24	*0,2236	*0,1913	*0,2139	*0,2263	0,1489	*0,2238	*0,2234	0,1151	0,1372	*0,2121
computer-scientists-26	123	55	0,1461	0,1350	*0,1889	0,1676	0,1247	0,1755	0,1709	0,1008	0,1216	0,1724
non-programmers-16	46	32	0,2817	0,2613	*0,3395	0,2918	0,2581	0,3085	0,2988	0,2093	0,2451	0,2955
non-programmers-24	34	25	*0,3517	0,3394	*0,3945	*0,3754	0,3233	*0,3686	*0,3574	0,2748	0,3239	*0,3693
combined-references-file	904	132	0,1357	0,1549	*0,1983	0,1732	0,1647	0,1791	0,1723	0,1338	0,1598	0,1742
rank 1-5					1	4		2	5			3

Tabelle 2: Vergleich der Vorhersagegenauigkeiten (MPA) für unterschiedliche AVAs bei den Greenberg-Referenzdateien.

	o	A	IPAM/0,7	IPAM2/0,7	KO3/18	LEV3	PPM4	PPM+4	PPMU+4	PPMC4	PPM*	PPM*+
novice-programmers-15	71	14	0,4299	*0,5572	0,5069	0,4972	*0,6177	0,5045	0,5517	*0,6189	*0,6213	0,5080
novice-programmers-20	54	18	0,3045	0,3834	*0,4807	0,3918	*0,4262	0,3881	0,3846	*0,4292	*0,4342	0,3962
experienced-programmers-14	101	51	0,2106	*0,3339	*0,3309	0,2812	0,2502	0,2728	0,2768	0,2517	0,2621	0,2848
experienced-programmers-28	438	70	0,2702	*0,3239	*0,3377	*0,3198	0,2445	*0,3217	*0,3326	0,2449	0,2511	*0,3283
computer-scientists-11	37	24	0,2986	0,3948	*0,5179	0,4112	0,1855	0,4118	0,4295	0,1859	0,1827	0,409
computer-scientists-26	123	55	0,1947	0,3223	*0,3623	0,2988	0,1608	0,3099	0,3118	0,1631	0,1651	0,3142
non-programmers-16	46	32	0,4200	*0,4767	*0,4839	*0,4274	0,3695	*0,4364	*0,4443	0,3688	0,3623	*0,4291
non-programmers-24	34	25	*0,4608	*0,4949	*0,5258	*0,5064	*0,5032	*0,5032	*0,5298	*0,4946	*0,5184	*0,5184
combined-references-file	904	132	0,1719	0,2456	*0,2767	*0,2630	0,2405	*0,2628	*0,2736	0,2408	0,2460	*0,2683
rank 1-5				3	1			5	2			4

7. Untersuchung: Unterstützungsfunktion in ACTIVE-UI

7.1 Versuchsdurchführung

Abschließend wurde eine Untersuchung mit der aktuellen Version von ACTIVE-UI durchgeführt, um den Nutzen adaptiver Unterstützungssysteme sowie auch die Effekte zwischen den AVAs und der Art des Unterstützungssystems zu betrachten.

Dazu wurden die bereits beschriebene adaptive Hilfe und der adaptive Tutor in ACTIVE-UI integriert. Als dritte Bedingung wurde eine kontextsensitive Hilfe geschaffen, und die vierte Bedingung war ACTIVE-UI ohne jegliche Unterstützung.

Die Testpersonen ohne Erfahrung mit der Benutzung von ACTIVE-UI hatten insgesamt drei verschiedene Schweißaufträge durchzuführen, wobei der Schwierigkeitsgrad dabei sukzessive anstieg.

Abhängige Variablen waren die Art der Unterstützung und ein vorher durchgeführtes Training des Benutzers. Die beiden adaptiven Systeme wurden mit neun Interaktionssequenzen von erfahrenen ACTIVE-UI-Benutzern bei Durchführung dieser Aufgaben angelernt. Als AVA wurde PPM+4 gewählt, weil sein Verhalten gut bekannt ist und in den Vorstudien gute Werte bezüglich der *MPP* lieferte.

Die Testreihen wurden von insgesamt 48 Studenten verschiedenen Alters durchgeführt. Das Durchschnittsalter betrug 24,6 Jahre, und alle Teilnehmer waren erfahrene Computer-Anwender. Vor Versuchsbeginn wurde das räumliche Vorstellungsvermögen mittels eines dreidimensionalen Würfeltests (Three-Dimensional Cube Test, Gittler 1999) ermittelt. Der durchschnittlich erzielte Wert betrug +0,89. Werte zwischen -3 und 0 weisen auf ein geringes räumliches Vorstellungsvermögen hin, wohingegen Werte zwischen 0 und +3 für ein hohes räumliches Vorstellungsvermögen stehen.

Tabelle 3 zeigt die Aufteilung der Versuchspersonen auf die unterschiedlichen Hilfesysteme. Im dritten Versuchsdurchlauf durfte die Hälfte der Testpersonen ein Pre-Training durchführen. Das Vorgehen dieser Untersuchung (Between-Subject-Design) ist detaillierter in Künzer et al. (2003) und Ziefle et al. (2004) dargestellt.

Tabelle 3: Aufteilung der Versuchspersonen zu den unterschiedlichen Hilfesystemen.

unabhängige Variable	adaptiver Tutor n = 12		adaptive Hilfe n = 12		kontextsensitive Hilfe n = 12		ohne Hilfe n = 12	
	Ja n = 6	Nein n = 6	Ja n = 6	Nein n = 6	Ja n = 6	Nein n = 6	Ja n = 6	Nein n = 6
Pre-Training								

7.2 Personenbezogene Ergebnisse

Abbildung 9 zeigt die Auswertung aller 48 Versuchspersonen im Hinblick auf den Trainingseffekt (vom ersten bis zum dritten Versuch), wobei es signifikante Unterschiede unter den Versuchspersonen gibt: Für die Lösung einer Aufgabe werden je nach angebotener Hilfe unterschiedliche Bearbeitungszeiten benötigt. Der allgemeine Lerneffekt ist hoch signifikant ($F(2,94) = 197,58$; $p < 0,01$) und zeigt, dass der dritte Versuch deutlich schneller absolviert wurde als der erste Versuch. Weiterhin gibt es einen deutlichen Hinweis zwischen Lerneffekt und angebotenen Hilfssystem ($F(2,6)$

= 2,77; $p < 0,05$). Die höchste Verbesserung der Bearbeitungszeit (64%) zeigten die Testpersonen unter Verwendung des adaptiven Tutors. Die Unterstützungssysteme *kontextbasierte Hilfe* (52%) und sowohl die *adaptive Hilfe*, als auch *ohne Hilfe* erreichten eine zeitliche Verbesserung um 54 %. Interessanterweise hing der Lernerfolg und der Nutzen des Hilfesystems davon ab, ob die Testpersonen einen anfänglichen Probeversuch durchgeführt hatten ($F(1,3) = 3,22$; $p < 0,05$) und so in einer Eingewöhnungsphase die Programmstruktur erlernen konnten. Einen anderen starken Einfluss auf die Effektivität hatte das räumliche Vorstellungsvermögen der Testpersonen (Abbildung 10). Je geringer dieses war, desto geringer war die Anzahl gelöster Aufgaben bei gleichzeitig höherer Bearbeitungszeit.

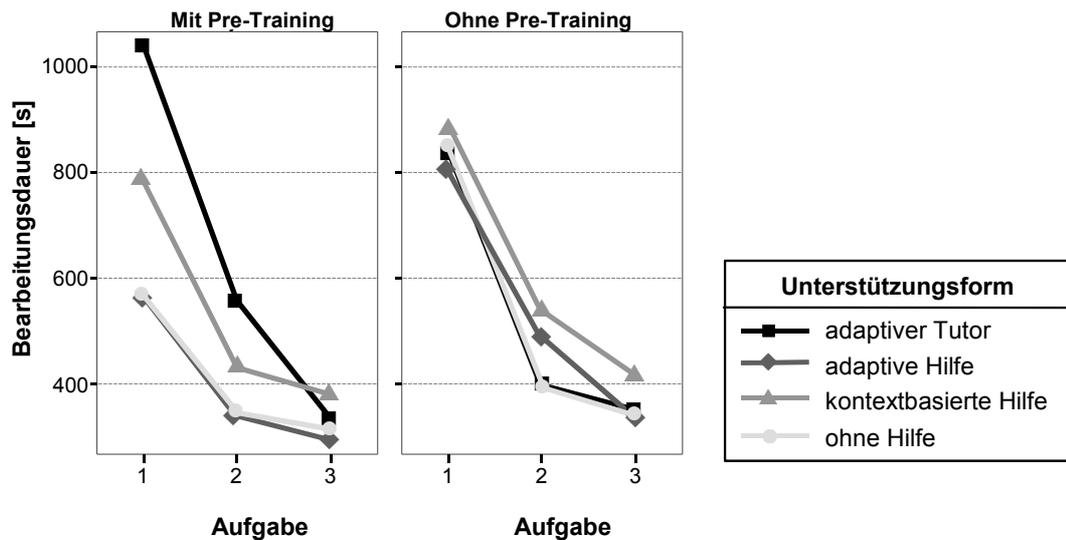


Abbildung 9: Einfluss des Trainingseffekts auf die benötigte Gesamtzeit in Abhängigkeit von der Versuchsanzahl und des verwendeten Hilfesystems.

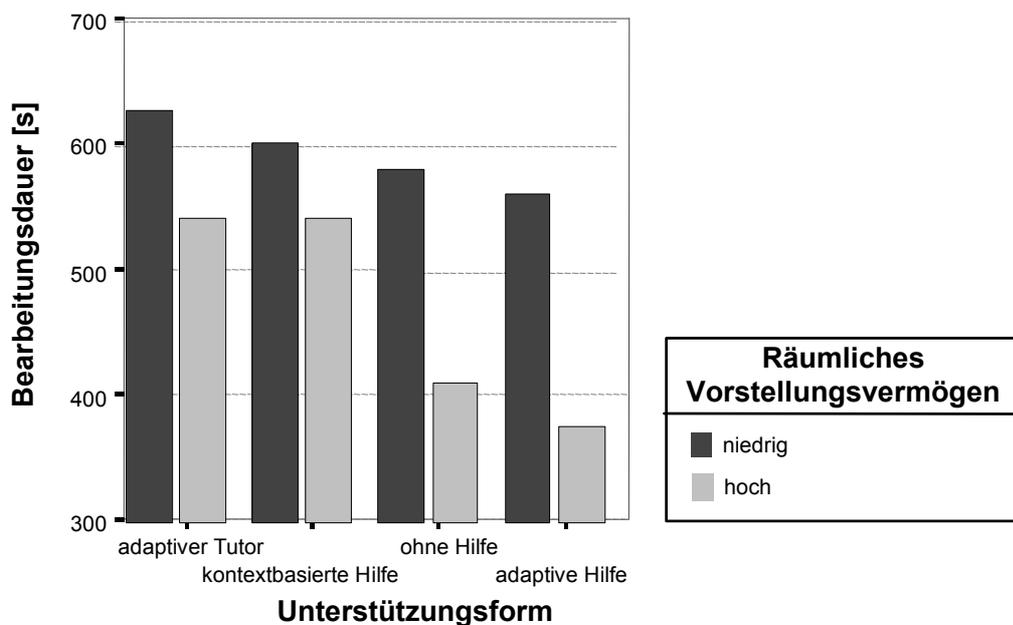


Abbildung 10: Einfluss des räumlichen Vorstellungsvermögens und der Unterstützungsf orm auf die benötigte Gesamtzeit.

In den Abbildungen 11 und 12 sind die Anzahl der gelösten Aufgaben und die Anzahl von Umwegschritten in Bezug auf den optimalen Pfad für die erste und dritte Aufgabe aller Tutorengruppen in Abhängigkeit vom räumlichen Vorstellungsvermögen dargestellt. Versuchspersonen ohne Hilfesystem lösten die geringste Anzahl an Aufgaben, wohingegen bei Unterstützung durch den adaptiven Tutor die besten Resultate erzielt wurden. Dies konnte auch bei Messungen der benötigten Umwegschritte signifikant bestätigt werden. Auch hier zeigt die Verwendung eines adaptiven Tutors die stärkste Abnahme an eingeschlagenen Umwegschritten.

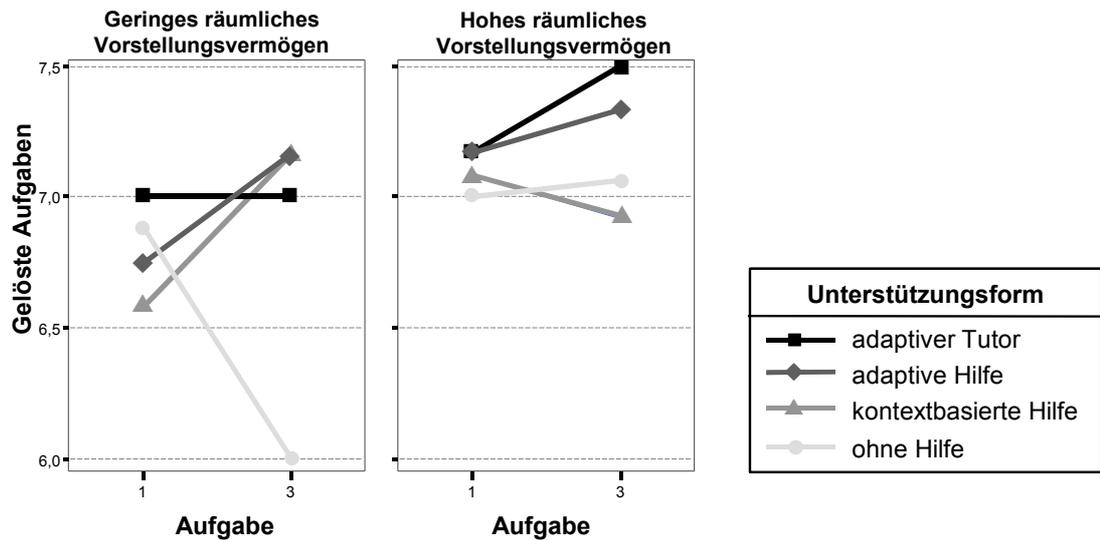


Abbildung 11: Auswertung der Anzahl gelöster Aufgaben in Abhängigkeit vom räumlichen Vorstellungsvermögen.

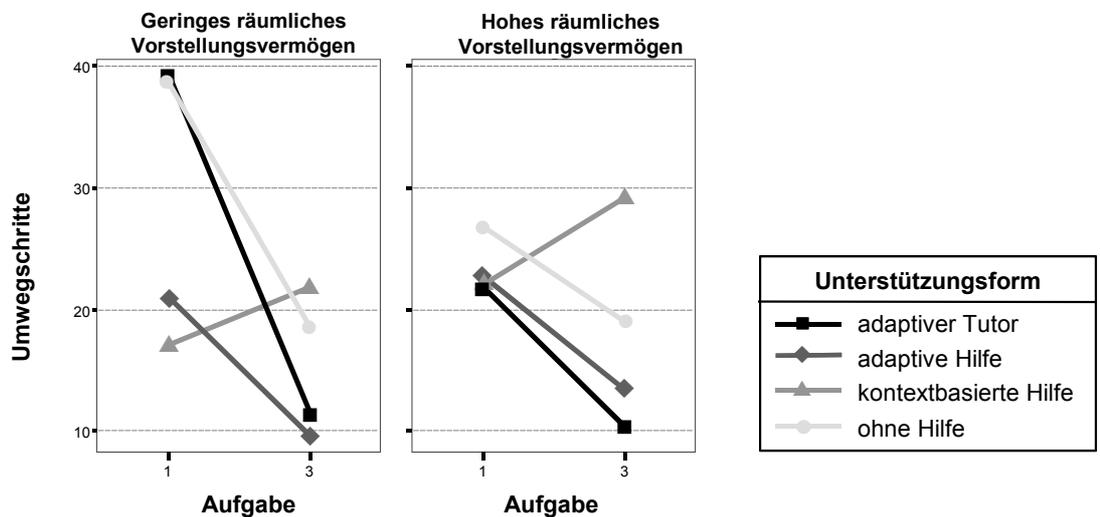


Abbildung 12: Auswertung der Anzahl eingeschlagener Umwegschritte in Abhängigkeit vom räumlichen Vorstellungsvermögen.

7.3 AVA-bezogene Ergebnisse

Obwohl PPM+ 4 als AVA für die adaptiven Unterstützungsförmungen benutzt wurde, ist es natürlich möglich, die MPP aller Algorithmen ohne vorheriges Training zu

berechnen. Das Ergebnis ist in Abbildung 13 grafisch dargestellt. Einerseits zeigte sich, dass die Wahl von PPM+4 sinnvoll war, da die übrigen Algorithmen keine signifikant besseren *MPPs* erzielen konnten. Trotzdem lieferte KO3/18 erneut die höchste *MPP*.

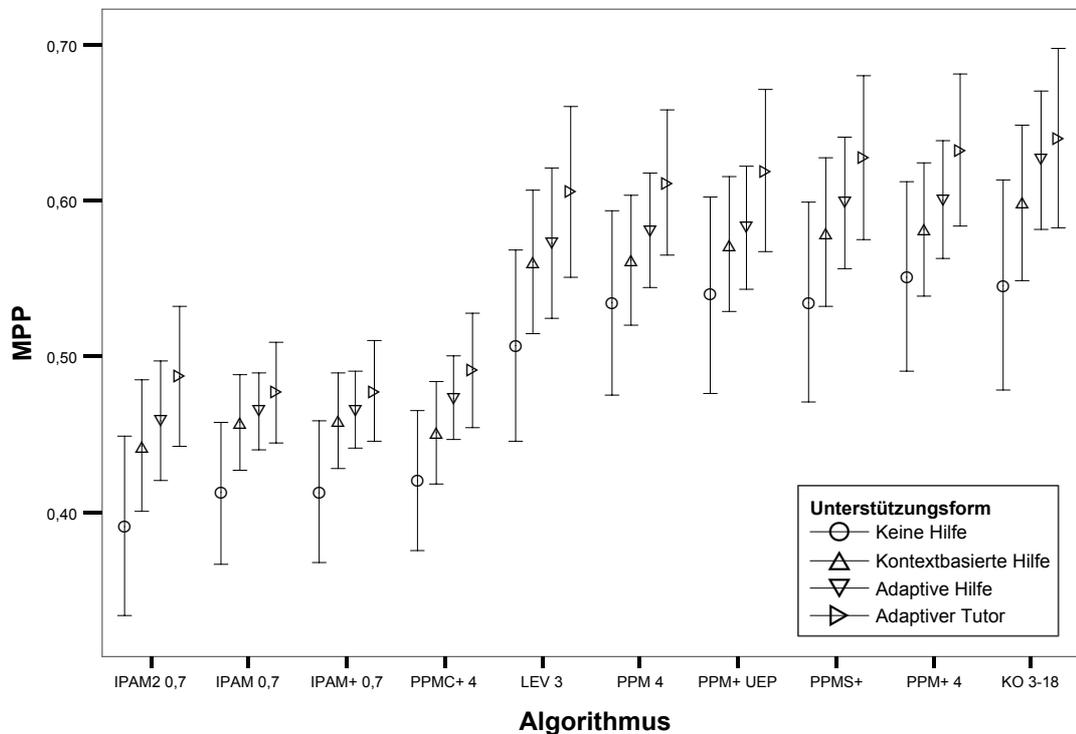


Abbildung 13: Vorhersagewahrscheinlichkeit für die unterschiedlichen Unterstützungssysteme in ACTIVE-UI.

Die *MPPs* können darüber hinaus auch als Maß dafür angesehen werden, wie ähnlich die Interaktionssequenzen der Testpersonen zu denen der Experten sind. Bedingt durch die vom Umfang her kleine Datenbasis gab es aber keine signifikanten Effekte, allerdings lieferte für alle AVAs die Bedingung „keine Hilfe“ bezüglich der *MPP* die niedrigsten Ergebnisse, gefolgt von „kontextsensitiver Hilfe“, „adaptive Hilfe“ und „adaptiver Tutor“, der am besten abschnitt. Daher wurde eine Normalisierung der *MPPs* durchgeführt, mit dem Ziel, die Unterschiede der Level- und Tendenzunterschiede der Werte für jeden AVA anzupassen, wobei die Formel $x_N = (x - x_{\min}) / (x_{\max} - x_{\min})$ zu Grunde gelegt wurde. Eine einfaktorische ANOVA lieferte für die normalisierten Ergebnisse einen F-Wert von $F_{ANOVA}=13,378$ ($df=3$; $p_{ANOVA}<0,001$) und der Post-hoc Student-Newman-Keul Test zeigte, dass die homogene Untergruppe „adaptive Hilfe“ ($MPP=0,6680$) und „adaptiver Tutor“ ($MPP=0,6405$) signifikant besser ist, als „keine Hilfe“ ($MPP=0,5701$). Dies kann durch den Einfluss des Hilfe- oder Tutorsystems selbst erklärt werden, die dem Benutzer Vorschläge machen, die dieser tatsächlich befolgt. Besonders den Anwendern ohne Unterstützungsfunktion gelang es nicht, dem optimalen Interaktionspfad zur Lösung der Aufgaben zu folgen. Auch wenn die Bearbeitungszeiten am Ende niedrig waren, zeugen der hohe Fehleranteil und der niedrige Anteil an der Aufgabenbearbeitung davon, dass diese Benutzer aufgrund der fehlenden Unterstützung dazu gezwungen waren ein mentales Modell der Anwendung zu entwickeln, wozu sie eine Trial-And-Error-Strategie anwendeten.

8. Zusammenfassung und Ausblick

Dieser Artikel beschreibt die Verwendung von generischen Aktionsvorhersage-Algorithmen (AVAs) zur Entwicklung eines adaptiven Unterstützungssystems. Dazu wurden mit dem LEV- und dem KO-Algorithmus zwei neue AVAs vorgestellt und in drei verschiedenen Szenarios mit anderen AVAs verglichen. Die mittlere Vorhersagewahrscheinlichkeit (Mean Prediction Probability, *MPP*) des KO-Algorithmus lieferte dabei bessere Werte als bekannte Algorithmen wie IPAM oder PPM.

Basierend auf den AVAs wurden eine adaptive Hilfe und ein adaptiver Tutor entwickelt, die durch die Beobachtung von Expertenverhalten angelern werden können. Die adaptiven Unterstützungssysteme können dadurch Anfängern helfen, ihre Aufgaben effektiver durchzuführen, aber auch Experten erhalten aufgaben- und adaptive Hilfe- und Funktionsunterstützung im Rahmen ihrer Interaktionen.

Eine Auswertung dieser Ansätze mit 48 Probanden zeigte, dass die adaptiven Unterstützungsansätze zu einer Steigerung der Effektivität bei den Benutzern führen. Die Verwendung des PPM+-Algorithmus war gut geeignet, da keiner der anderen AVAs eine signifikant höhere *MPP* aufwies. Die Nutzung der Unterstützungssysteme innerhalb von ACTIVE-UI führte darüber hinaus zu Interaktionssequenzen mit einer höheren *MPP* als ohne Unterstützung, was zeigt, dass die Benutzer auch von der Vorschlägen beeinflusst werden, da die *MPP* auch als Ähnlichkeitsmaß interpretiert werden kann.

Trotzdem differiert die *MPP* der verschiedenen AVAs und ist stark von der konkreten Interaktionsaufgabe abhängig. Deshalb sollte zukünftig eine dynamische Auswahl des bestgeeigneten AVA und seiner Parameter dazu führen, die Vorhersagequalität der adaptiven Unterstützungssysteme weiter zu erhöhen.

9. Danksagungen

Die Autoren danken Brian Davison und Ian Witten für ihr Interesse und die hilfreichen Kommentare. Teile dieser Untersuchungen wurden von der Deutschen Forschungsgemeinschaft im Rahmen des Sonderforschungsbereiches 368 „Autonome Produktionszellen“ gefördert.

Literatur

- Albrecht, D.W.; Zukerman, I. & Nicholson, A.E. (1998). Bayesian Models for Keyhole Plan Recognition in an Adventure Game. In: *User Modeling and User-Adapted Interaction*, 8, S. 5-47.
- Carberry, S. (2001). Techniques for Plan Recognition. In: Kobsa, Alfred (Hrsg.). *UMUAI - User Modeling and User-Adapted Interaction*, Bd. 11 (1-2). New York: Springer Verlag, 2001, S. 31-48. Tenth Anniversary Issue of UMUAI.
- Cleary, J.G. & Witten, I.H. (1984). Data compression using adaptive coding and partial string matching. In: *IEEE Transactions on Communications* 32.

- Cleary, J.G.; Teahan, W.J. & Witten, I.H. (1995). Unbounded length contexts for PPM. In: *Proceedings of Data Compression Conference*. Los Alamitos, CA : IEEE Computer Society Press, S. 52-61.
- Cook, J. E. & Wolf, A. (1998). Discovering Models of Software Processes from Event-Based Data. In: *ACM Transactions on Software Engineering and Methodology*. Vol. 7.
- Davison, B. & Hirsch, H. (1998). Predicting sequences of user actions. In: *Notes of the AAAI/ICML 1998 Workshop on Predicting the Future: AI Approaches to Time-Series Analysis*.
- Dieterich, H.; Malinowski, U.; Kühme, T. & Schneider-Hufschmidt, M. (1993). State of the Art in Adaptive User Interfaces. In: Schneider-Hufschmidt, M. (Hrsg.) ; Kühme, T. (Hrsg.) ; Malinowski, U. (Hrsg.). *Adaptive User Interfaces: Principles and Practice*. Amsterdam : North-Holland, S. 13-48.
- Eisenstein, J. & Rich, C. (2002). Agents and GUIs from Task Models. 2002. In: *Conference on Intelligent User Interfaces (IUI 2002)*.
- Encarnação, L.M. & Stoev, S.L. (1999). An Application-Independent Intelligent User Support System Exploiting Action-Sequence Based User Modelling. In: *Proceedings of 7th International Conference on User Modeling*. Wien, New York : Springer-Verlag.
- Fischer, G. (2001). User Modeling in Human-Computer Interaction. In: *User Modeling and User-Adapted Interaction*, Bd. 11. Dordrecht.
- Foley, J.D.; Dam, A.V.; Feiner, S.K. & Hughes, J.F. (1990). *Computer Graphics - Principles and Practice*. Addison-Wesley.
- Gittler, G. (1999). *Three-dimensional Cube Test*. Stuttgart : Beltz.
- Gorniak, P.J. & Poole, D. (2000). Predicting future user actions by observing unmodified applications, *17th National Conference on AI, AAAI-2000*.
- Greenberg, S. (1988). *Using Unix: Collected Traces of 168 users*. In: Research Report 88/333/45. Canada : Department of Computer Science, University of Calgary.
- Greenberg, S. (1993). *The Computer User as Toolsmith: The Use, Reuse, and Organization of Computer-based Tools*.
- Hacker, W. (1978). *Allgemeine Arbeits- und Ingenieurspsychologie: Psychische Struktur und Regulation von Arbeitstätigkeiten*; Schriften zur Arbeitspsychologie. Bd. 20. Bern : Hans Huber.
- Hilbert, D.M. & Redmiles, D.F. (1999). *Extracting Usability Information from User Interface Events*. TR UCI-ICS-99-40, Department of Information and Computer Science, University of California, Irvine.
- Horvitz, E.; Breese, J.; Heckermann, D.; Hovel, D. & Rommelse, K. (1998). The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In: *Fourteenth Conf. on Uncertainty of Artificial Intelligence*, San Francisco, CA.

- Howard, P.G. & Vitter, J.S. (1994). *Design and Analysis of Fast Text Compression Based on Quasi-Arithmetic Coding*. DUKE-TR-1994-11.
- Kobsa, A. (2001). Generic User Modeling Systems. In: Kobsa, Alfred (Hrsg.). *UMUAI User Modeling and User-Adapted Interaction* Bd. 11 (1-2). New York: Springer Verlag, S. 49-63.
- Korvemaker, B. & Greiner, R. (2000). Predicting UNIX command lines: Adjusting to user patterns. In *Adaptive User Interfaces: Papers from the 2000 AAAI Spring Symposium*, S. 59-64.
- Künzer, A. & Luczak, H. (2003). Design of an Adaptive User Interface Based on Action Prediction. In The Ergonomics Society of Korea (Hrsg.). *Ergonomic in the Digital Age. Proceedings of the 15th Triennial Congress of the IEA and the 7th Joint Conference of Ergonomics Society of Korea*. Seoul 2003, S. 1-4.
- Künzer, A. (2004). *Handlungsprädiktion zur Gestaltung einer adaptiven Benutzungsunterstützung in autonomen Produktionszellen*. Shaker-Verlag: Aachen. In Vorbereitung.
- Künzer, A. ; Ziefle, M.; Bodendieck, A. & Luczak, H. (2003). Effects of Different User-Adaptive Help Systems on Task Performance. In Luczak, H.; Zink, K.J. (Hrsg.). *Human Factors in Organizational Design and Management - VII. Proceedings of the Seventh International Symposium on Human Factors in Organizational Design and Management held in Aachen, October 1-2*. IEA Press: Santa Monica, CA, USA 2003, S. 461-467.
- Künzer, A.; Schlick, C.; Ohmann, F.; Schmidt, L. & Luczak, H. (2001). Eine empirische Untersuchung zur Modellierung von Handlungsvorhersagen mit Hilfe dynamischer Bayes-Netze. In: *Human Factors bei der Entwicklung von Fahrzeugen* (DGLR), Bonn.
- Langley, P. (1997). Machine Learning for Adaptive User Interfaces. In: *Proceedings of the 21st German Annual Conference on AI*. Springer: Germany.
- Lau, T. (1999). *A comparison of sequence-learning approaches: implications for intelligent user interfaces*. University of Washington.
- Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions and reversals. In: *Soviet Physics Doklady* Bd. 10(8), S. 707-710.
- Luczak, H.; Schlick, C.; Künzer, A. & Ohmann, F. (2001). Syntactic user modeling with stochastic processes. *Theoretical Issues in Ergonomics Science*. London.
- Maes, P. (1994). Agents that reduce work and information overload. In: *Communications of the ACM* 37 (1994), Nr. 7, S. 30-40.
- Mannila, H.; Toivonen, H. & Verkamo, A. I. (1997). Discovery of frequent episodes in event sequences. In: *Data Mining and Knowledge Discovery*, 1(3), 1997.
- Moffat, A. (1990). Implementing the PPM data compression scheme. In: *IEEE Transactions on Communications*.
- Ross, E. (2000). *Intelligent User Interfaces: Survey and Research Directions*. TR CSTR-00-004, Dep. of Computer Sc., University of Bristol.

- Schlick, C. (2000). Simulation of Rule-Based Behavior for a Multimodal Interaction Task with Stochastic Petri Nets. In: *Proc. of the XIVth Triennial Congress of the IEA*. Santa Monica: CA.
- Schlick, C.; Winkelholz C.; Motz F.; Künzer, A. & Luczak, H. (2002). Stochastic operator models for multiple target search tasks. In: *IEEE International Conference on Systems, Man and Cybernetics*. Hammamet, Tunisia.
- Sun, R. (2000). Introduction to sequence learning. In: *Sequence Learning: Paradigms, algorithms, and Applications*. Springer-Verlag, Berlin.
- Ziefle, M. ; Künzer, A. & Bodendieck, A. (2004). The impact of user characteristics on the utility of adaptive help systems. In: *Proceedings of the WWCS 2004*, Kuala Lumpur, Malaysia. In Druck.
- Ziv, J. & Lempel, A. (1977). A Universal Algorithm for Sequential Data Compression. In: *IEEE Transactions on Information Theory*, S. 337-343.
- Zukerman, I.; Albrecht. W. & Nicholson, A. (1999). Predicting user's request on the WWW. In: *UM99 - Proceedings of the 7th International Conf. on UM*.

Ansätze zur kognitiven Simulation eines Autofahrers

DANIEL KRAJZEWICZ & PETER WAGNER

Institut für Verkehrsforschung, Deutsches Zentrum für Luft- und Raumfahrt

Schlüsselwörter: Fahrermodellierung, Verkehrssimulationen, Verkehrsmodelle, Kognition

1. Zusammenfassung

Das Institut für Verkehrsforschung am Deutschen Zentrum für Luft- und Raumfahrt (IVF/DLR) setzt in vielen Projekten Simulationen des Straßenverkehrs ein, z.B. um Schwachstellen in Verkehrsnetzen zu finden oder um Geräte zur Verkehrskontrolle oder -beeinflussung während ihrer Entwicklung zu bewerten. In der Regel kommen dabei sogenannte mikroskopische Simulationen zum Einsatz, deren betrachtete Größe ein Fahrer-Fahrzeug-Objekt ist, das die Bewegung eines Fahrzeugs im Verkehrsnetz durch wenige Gleichungen beschreibt. Solche Modelle erlauben die Simulation des Straßenverkehrs großer Städte in Echtzeit, allerdings bilden sie den Prozess des Fahrzeugführens nur vereinfacht ab.

Innerhalb eines der Projekte des IVF soll das Verhalten eines einzelnen Autofahrers genauer untersucht und modelliert werden. Während solche Modelle auch für andere Gebiete der Verkehrsforschung interessant sind, z.B. der Forschung zu Fahrsicherheit oder zu Fahrerassistenzsystemen, erhoffen wir uns, so Rückschlüsse auf den Verkehrsfluss ziehen und somit die Qualität mikroskopischer Modelle erhöhen zu können. Im Rahmen dieses Berichts sollen nach einer kurzen Einführung in die Thematik „Verkehrssimulation“ unsere ersten Ansätze zum Aufbau einer in ein simuliertes Verkehrsgeschehen integrierten Simulation der Fahrerkognition gegeben werden.

2. Einleitung

Simulationen sind ein innerhalb der Verkehrsforschung häufig eingesetztes Werkzeug. Bei der Entwicklung neuer Technologien werden Simulationen benutzt um die Auswirkungen dieser Technologien zu berechnen und so deren Vorteile gegenüber konventionellen Methoden vorherzusagen. Ein Beispiel hierfür ist die Simulation „optischer Informationssysteme“ (OIS) im Rahmen des gleichnamigen Projektes des

IVF. Die Kenngrößen simulierter OIS wurden innerhalb der Simulation für eine Lichtsignalsteuerung verwendet. So konnte ohne direkte Eingriffe in den realen Verkehr gezeigt werden, dass die von OIS erzeugten Beschreibungen des Verkehrszustandes, die herkömmliche Detektoren nicht liefern, verbesserte Flusststeuerungen ermöglichen. Im Rahmen des Großprojektes INVENT der deutschen Automobilhersteller werden Simulationen eingesetzt, um die hier entwickelten Verkehrsmanagementstrategien zu bewerten und neue Methoden der Optimierung des Verkehrsflusses zu entwickeln. Ein weiteres, im Jahr 2002 abgeschlossenes Projekt, befasste sich mit der Simulation des Mobilitätsverhaltens einer synthetischen Bevölkerung. Hier wurden statt Fahrzeugen einzelne Personen und ihre Mobilitätswünsche modelliert. Da die Fahrtwünsche auf ein multimodales Verkehrsnetz umgelegt wurden, diente eine Verkehrssimulation als ein mittelbar eingesetztes Werkzeug zum Berechnen des Verkehrszustandes auf der Straße und somit der Attraktivität des straßengebundenen Verkehrs für die synthetische Bevölkerung (Hertkorn 2002). Neben weiteren Faktoren sollte so die Entwicklung der Bevölkerungsstruktur vorhergesagt werden.

Das IVF versucht die Qualität der eingesetzten Modelle zu verbessern. So werden z.B. mit speziell erweiterten Fahrzeugen oder neuen Sensoren Messungen durchgeführt, um mehr über die tatsächliche Dynamik des Verkehrs zu erfahren, neue mikroskopische Modelle entwickelt oder die Parameter bekannter mikroskopischer Verkehrsflussmodelle an Messungen angepasst (Brockfeld, Kühne & Wagner 2002). Das hier vorgestellte „Kognitive Fahrermodell“ versucht ebenfalls die Abbildungsgenauigkeit mikroskopischer Modelle zu verbessern, verfolgt jedoch einen anderen Ansatz. Statt weiterer Arbeit mit mikroskopischen Modellen wird versucht, ein genaueres Denk- und Datenflussmodell eines einzelnen Autofahrers zu erstellen. Solche Modelle werden im Allgemeinen als „sub-mikroskopisch“ bezeichnet. Die aus diesem Modell und den Ergebnissen der mit ihm durchgeführten Simulationen gewonnenen Erkenntnisse sollen dann in mikroskopische Verkehrssimulationen und -modelle eingebettet werden.

In den folgenden Kapiteln wird zunächst das Thema Verkehrssimulationen besprochen, wobei neben einer Vorstellung der Annahmen mikroskopischer Modelle erläutert wird, wofür fein aufgelöste Modelle eines menschlichen Fahrers benutzt werden können. Danach wird das von uns entwickelte kognitive Modell vorgestellt. Im Anschluss wird ein Ausblick auf seine weitere Entwicklung gegeben.

3. Verkehrssimulationen und -modelle

3.1 Mikroskopische Simulationen

In vielen Fällen ist das bei Verkehrssimulationen betrachtete Areal sehr groß, meistens eine ganze Stadt oder ein Stadtzentrum. Von den eingesetzten Modellen wird daher nicht nur eine hohe Qualität, sondern auch eine hohe Ausführungsgeschwindigkeit verlangt. So konnten noch vor einigen Jahren für solche Simulationen nur sogenannte makroskopische Modelle eingesetzt werden. Statt einzelner Fahrzeuge betrachten diese Modelle den Verkehrsfluss – eine aus den einzelnen Fahrzeugbewegungen aggregierte Größe. Durch den Fortschritt der Computertechnik werden diese Modelle jedoch zunehmend durch sogenannte mikroskopische Simulationen ersetzt, die den Verkehrsfluss in einzelne Fahrzeuge aufbrechen und das Verhalten dieser abbilden. Die in mikroskopischen Simulationen benutzten Modelle zur Beschreibung

der Fahrzeugbewegungen sind bewusst einfach gehalten, um eine hohe Ausführungsgeschwindigkeit zu erreichen. So ist die am IVF entwickelte mikroskopische Simulation "SUMO"¹ (siehe Abbildung 1) in der Lage, ca. 1.000.000 Fahrzeugbewegungen pro Sekunde auf einem 1GHz-PC zu berechnen. Jede Fahrzeugbewegung beschreibt dabei das Verhalten des simulierten Fahrzeugs über einen Zeitraum von einer Sekunde.

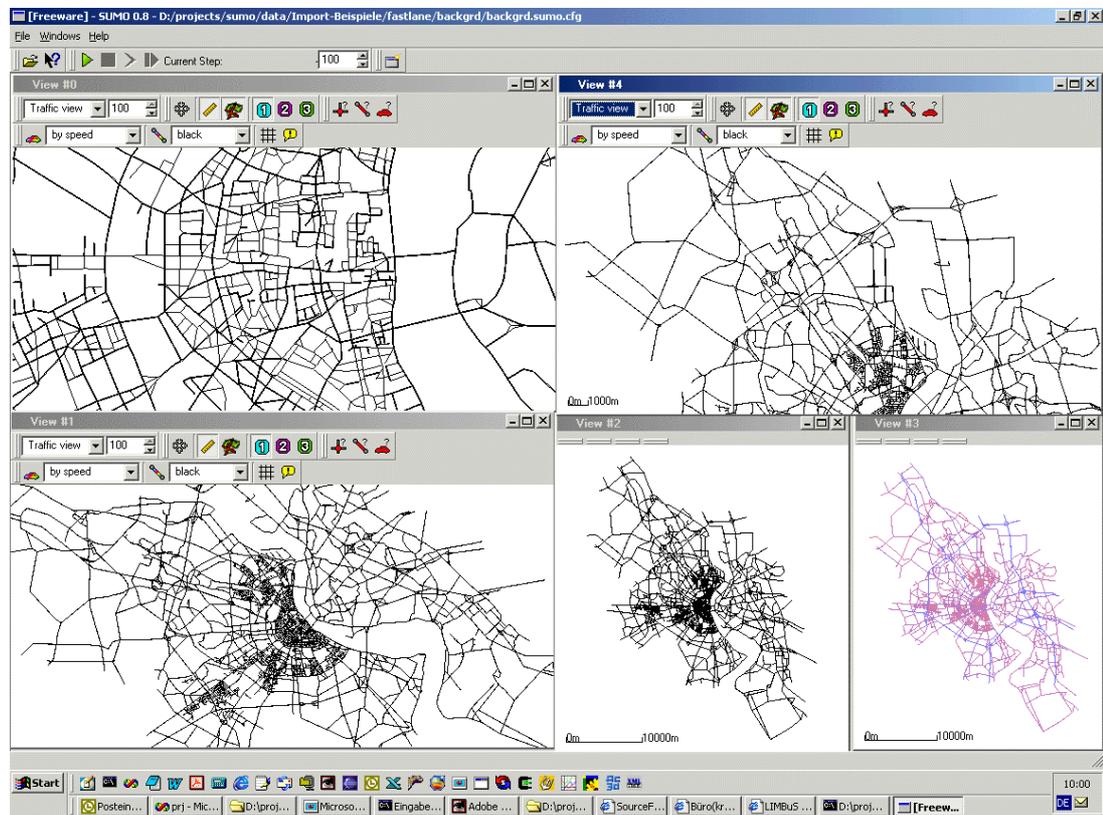


Abbildung 1: Mikroskopische Simulation von Köln mit SUMO.

Innerhalb der Verkehrsforschung existiert eine Vielzahl mikroskopischer Modelle. Sie müssen vor allem in der Lage sein, den Fluss auf einer Straße realistisch abzubilden. Diese Größe hängt hauptsächlich von der Dichte der Fahrzeuge auf der Straße ab, wie in Abbildung 2 zu sehen ist. Auch sollte das Entstehen und das Verschwinden von Staus möglichst korrekt simuliert werden. Aus Beobachtungen ist z.B. bekannt, dass Staus die Eigenschaft haben, sich stromabwärts zu bewegen und spontan zu entstehen. Eine durch mikroskopische Modelle zudem oft umgesetzte Eigenschaft der Fahrzeugdynamik ist das von Wiedemann (1974) entdeckte und formalisierte Pendelverhalten eines Fahrers um einen Fixpunkt hinter dem vorausfahrenden Wagen.

¹ „Simulation of Urban MObility“ (SUMO) ist eine frei erhältliche, mikroskopische Verkehrssimulation; siehe: <http://sumo.sourceforge.net/>

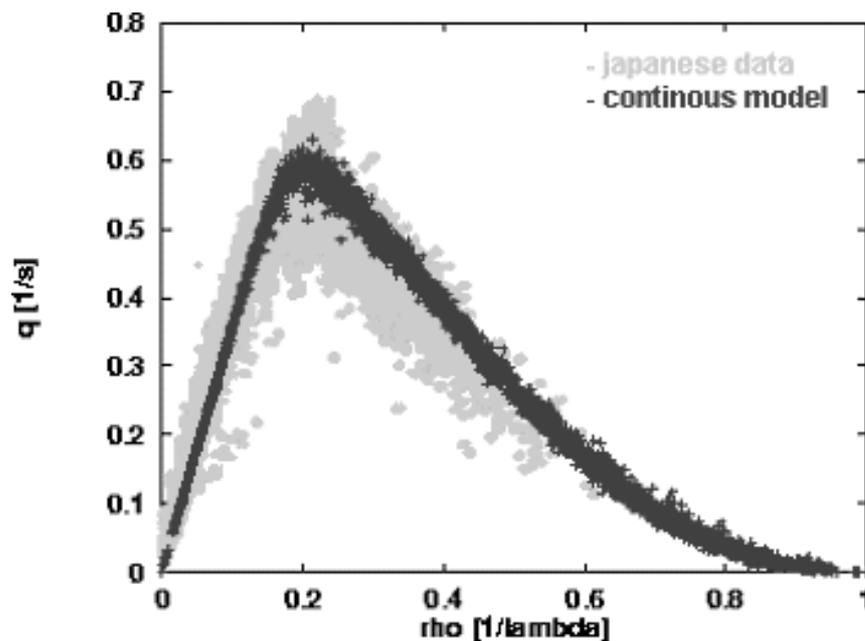


Abbildung 2: Fundamentaldiagramm (Fahrzeugfluss über Fahrzeugdichte).

Mikroskopische Modelle, die die erwähnten Anforderungen erfüllen, können sehr einfach gehalten sein. So besitzt das am IVF eingesetzte, in Krauß (1998) beschriebene Modell nur fünf Parameter: die Fahrzeuglänge, das maximale Bremsvermögen, die maximale Beschleunigung, die Höchstgeschwindigkeit und eine weitere Variable, die das Unvermögen des Fahrers, die gewünschte Geschwindigkeit zu erreichen, beschreibt. Die so genannte „gewünschte“ Geschwindigkeit ergibt sich aus dem Sicherheitsabstand, der aus den Geschwindigkeiten beider Fahrzeuge, dem Abstand zwischen ihnen und ihrem Bremsvermögen errechnet wird – ein Paradigma, das innerhalb der mikroskopischen Modelle sehr häufig benutzt wird, weshalb diese als „Fahrzeugfolgemodelle“ bezeichnet werden. Die tatsächliche Geschwindigkeit des simulierten Fahrzeugs hängt neben der vom Fahrer gewünschten Geschwindigkeit von dem Beschleunigungsvermögen, der Maximalgeschwindigkeit des Fahrzeugs und der das Unvermögen des Fahrers beschreibenden Variable ab.

Das Modell von Krauß beschreibt nur die Längsgeschwindigkeit, wie die meisten mikroskopischen Modelle. Mit dem Einsatz der Modelle in „realen“ Umgebungen, also Wegenetzen, wurde eine Simulation der Spurwechselmanöver nötig. Innerhalb mikroskopischer Simulationen wird der Spurwechsel meistens auf bereits vorhandene mikroskopische Fahrzeugfolgemodelle aufgesetzt. Die zeitliche Dynamik wird kaum betrachtet. Innerhalb eines Schrittes kann das simulierte Fahrzeug von einer Spur auf eine andere wechseln – und im nächsten wieder zurück.

Wie aus ihrer Vorstellung erkennbar, vernachlässigen mikroskopische Modelle viele Elemente der Fahrzeugdynamik. Nur einige beinhalten die mit steigender Fahrzeuggeschwindigkeit abnehmende Beschleunigung. Ebenfalls selten modelliert werden Gangwechsel, die aufgrund von Aufmerksamkeit variierenden Reaktionszeiten des Fahrers, die unter Umständen fehlerbehaftete Aufnahme von Informationen aus der Umwelt, die Wahl der Geschwindigkeit und andere Aspekte der Fahrzeugführung, auch wenn diese Aspekte sicherlich Auswirkungen auf den Verkehrsfluss haben.

3.2 Sub-Mikroskopische Simulationen

Erst seit der zweiten Hälfte der 90er Jahre werden viele mit den Themen „Fahrerassistenzsysteme“ und „Verkehrssicherheit“ verknüpften Modelle entwickelt, die versuchen, den Fahrer genauer abzubilden. Zum Einen wurde diese Entwicklung erst in den letzten Jahren durch Fortschritte der Psychologie, der Forschung zu Mensch-Maschine-Interaktion und durch die gleichzeitig steigende verfügbare Rechenkapazität möglich und berechenbar. Zum Anderen rückt die Thematik selbst verstärkt in den Vordergrund in dem Bemühen, das Fahren sicherer zu gestalten und den Versuchen, dieses durch das Abbilden und Erforschen dieses Prozesses in Simulationen zu erreichen (vgl. Jürgensohn 2001).

Durch eine feinere Auflösung des Fahrer-Fahrzeug-Objektes und die damit einhergehende Reduktion der Simulationsgeschwindigkeit sind sub-mikroskopische Simulationen kaum in der Lage die Bewegung vieler Fahrzeuge in einer vertretbaren Zeit zu berechnen. Somit ändern sich auch die an sie gestellten Aufgaben. Sub-mikroskopische Simulationen werden für die Beantwortung folgender Fragestellungen herangezogen:

- Wie gut sind die fahrzeuginternen Anzeigen erkennbar?
- Wie aufmerksam ist der Fahrer und wo gibt es Engpässe in seiner Aufmerksamkeit?
- Welchen Einfluss haben Fahrerassistenzsysteme und andere Geräte auf den Fahrprozess und die Fahrtüchtigkeit des Fahrers?
- Wodurch entstehen Unfälle und wie können sie verhindert werden?
- Wie entstehen makroskopisch beobachtbare Verkehrszustände (Stau, Stop-And-Go, synchronisierter Verkehr)?
- Wie wird der Verkehr durch das Verhalten einzelner Fahrer bedingt?

Die hier beschriebene Arbeit konzentriert sich vorrangig auf die Beantwortung der letzten beiden Fragen. Klassische Untersuchungen zu Mensch-Maschine-Interaktion werden sekundär behandelt.

4. Modellstruktur

4.1 Grundstruktur der Simulation

Das Modell soll die menschliche Kognition abbilden. Fokussiert werden die für den Vorgang des Fahrzeugführens notwendigen Strukturen und Prozesse. Unter Kognition versteht man nach Neisser „...alle jene Prozesse, durch die der sensorische Input umgesetzt, reduziert, weiter verarbeitet, gespeichert, wieder hervorgeholt und schließlich benutzt wird“ (Neisser 1974 S.19). Ausgehend von der Annahme, der Fahrer würde Informationen aus der Umwelt aufnehmen, in Bedeutungen umwandeln und aus diesen seine Aktionen ableiten, erhält man den in Abb. 3 schematisch dargestellten Informationsfluß in einer stark vereinfachenden Form. Diese Sicht auf den Fahrprozess wird in Ingenieurwissenschaften als die „Fahrer-Fahrzeug-Umwelt-Schleife“ bezeichnet (vgl. Jürgensohn 2001). Das Modell soll den gesamten Vorgang möglichst komplett, wenn auch an einigen Stellen vereinfachend, umfassen.

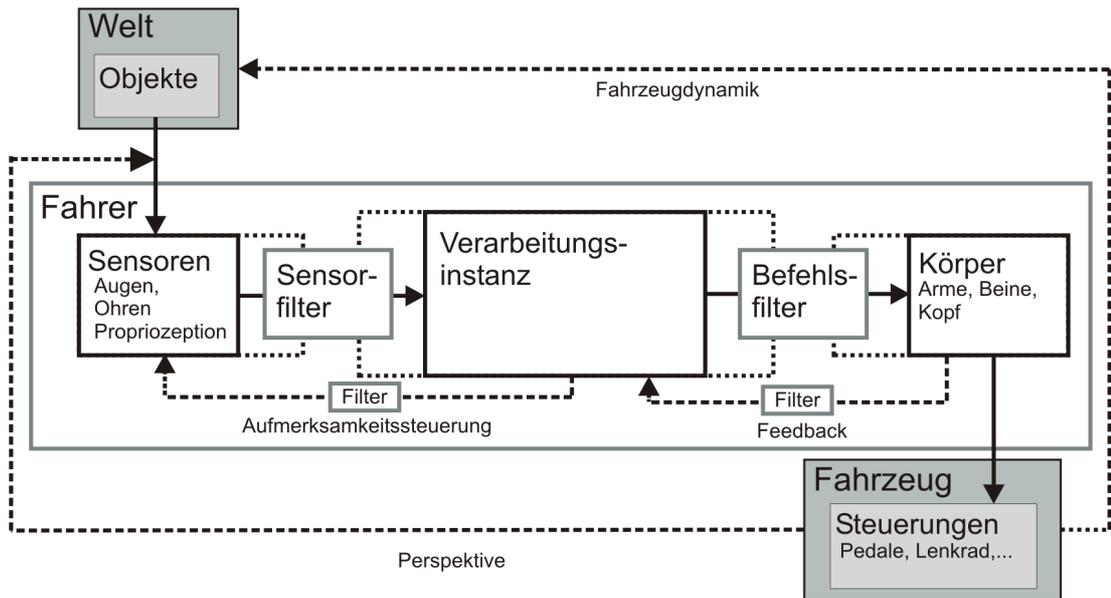


Abbildung 3: Schematische Abbildung des Informationsflusses beim Führen eines Fahrzeugs.

Die Simulation spielt in einem dreidimensionalen Raum. Aus diesem Grund haben alle in der Simulation enthaltenen Objekte ein Aussehen, wodurch nicht nur die Visualisierung der Szene vereinfacht, sondern auch eine realitätsnahe Modellierung der Wahrnehmung möglich wird. Abbildung 4 zeigt die Darstellung einer Simulationsszene.

Als Programmiersprache wurde Java ausgewählt. Das gesamte System wurde vollständig neu aufgebaut, der Einsatz anderer submikroskopischer Verkehrssimulationen verworfen, weil die meisten nur unvollständig beschrieben und nur als geschlossene Programme erhältlich sind und somit der Forschung und der späteren Weiterentwicklung nicht dienen können. Durch einen eigenen Ansatz erhoffen wir uns auch einen größeren Erkenntnisgewinn.

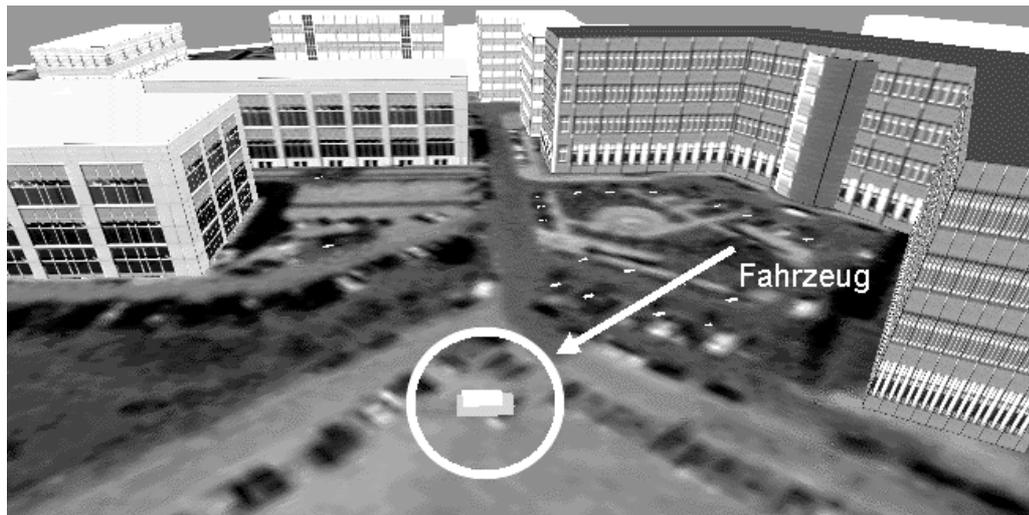


Abbildung 4: Beispiel für die Darstellung der Simulationsumgebung; das Fahrzeug wurde zur besseren Sichtbarkeit nachträglich markiert.

Das System ermöglicht die Spezifikation der Umgebung, der Aktionsmöglichkeiten des Autofahrers und weiterer Parameter in XML. Die Ausgaben werden in XML-Dateien abgelegt oder auf dem Bildschirm dargestellt. Die Simulation läuft mit simulierten Intervallen von 100ms für einfache Szenen in Echtzeit ab.

Unsere Sicht auf die Kognition ist bereits in Abb. 3 gezeigt worden. In den nächsten Abschnitten werden die in ihr enthaltenen Strukturen einzeln tiefergehender besprochen.

4.2 Sensoren

4.2.1 Sehen

Den Augen wird der größte Anteil an der Informationsaufnahme beim Autofahren zugesprochen. Das spiegelt sich wider in der Menge zu diesem Thema erhältlichen Arbeiten. Bei der Erforschung der Aufnahme optischer Informationen werden unterschiedliche Ansätze verfolgt. Neben Modellen, die das Erkennen und Bewerten von Szenen zum Gegenstand haben (siehe z.B. Jürgensohn 1997 oder Dörner 2001) finden sich Untersuchungen zu der Qualität der Informationsaufnahme (Epstein, Rogers 1995) oder zur Bewegung der Augen, sogenannten Sakkaden (siehe z.B. Velichkovsky et al 1995).

Mit dem Submodell des Sehens wird nicht der Anspruch verfolgt, den Prozess der Informationsverarbeitung von der Reizaufnahme bis zur semantischen Kodierung nachzubilden. Die tatsächlichen Abläufe sind noch weitgehend unerforscht. Ihre Komplexität und ihr massiv paralleles Ablaufen sind auf herkömmlichen, sequentiellen Rechnern nicht umsetzbar, wenn die Simulation eine praktikable Ausführungsgeschwindigkeit behalten soll. Interessant sind für uns die Augen stattdessen als eine begrenzte Ressource: Objekte, auf die sie nicht gerichtet sind, können nicht wahrgenommen werden, die Aufnahme einer bestimmten Information dauert eine bestimmte, statistisch erfassbare Zeit, das Erkennen eines Objektes z.B. ~400ms (Dornhoefer 1999). Der Fahrer ist daher in komplexen Situationen gezwungen, die Ressource „Sicht“ angemessen zu verwalten.

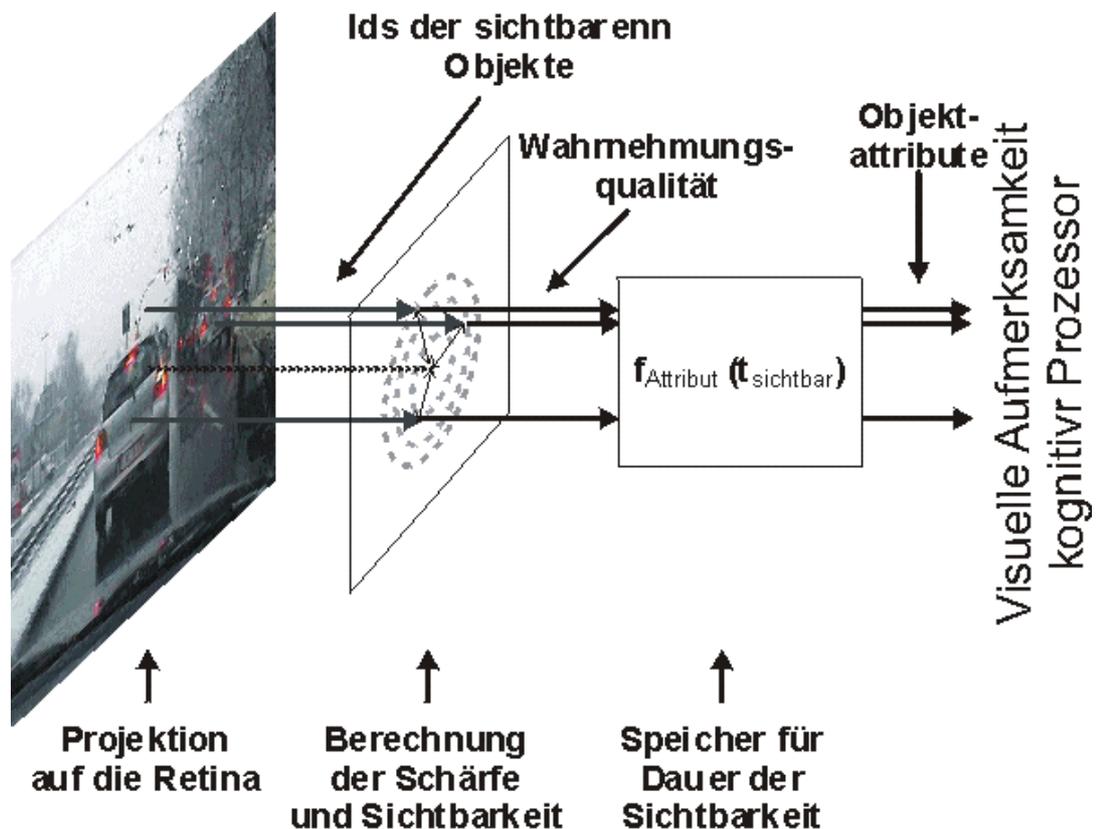


Abbildung 5: Abriss des Modells der Wahrnehmung.

Bei der Simulation des Sehens werden die in der Simulationsbeschreibung kodierte Objekte wie Fahrzeuge, Gebäude und Straßen auf eine simulierte Retina projiziert. In jedem Zeitschritt werden in Abhängigkeit vom Abstand zu der Retinamitte die Qualität ihrer Wahrnehmung berechnet und über mehrere Zeitschritte akkumuliert, wie in Abbildung 5 zu sehen. Die so gefilterten Informationen werden an die Verarbeitungsinstanz weiter gegeben.

4.2.2 Weitere Sinne

Als weitere Sinne werden der Hörsinn und die wahrgenommenen Fliehkräfte berücksichtigt. Den Hörsinn benutzen Fahrer vor allem, um den Zustand des eigenen Fahrzeugs zu bestimmen, insbesondere seiner Geschwindigkeit. In einem starken Zusammenhang damit steht die Wahl des Gangs, beschrieben in Diekamp 1995). Hierbei ist lediglich die Lautstärke des Fahrgeräusches wichtig, so dass das Geräuschmodell sehr einfach gehalten werden kann. In Abhängigkeit von der Drehzahl des Motors und der Geschwindigkeit wird ein Wert für die Lautstärke des Wagens berechnet und direkt an die Fahrerinstanz weiter gegeben. Es wird keine Filterung berücksichtigt.

Messungen ergaben, dass Fahrer „Lieblingsbeschleunigungen“, sowohl längs als auch quer zur Fahrtrichtung besitzen, und man hat versucht, diese zur Bildung von Fahrertypen heranzuziehen (Neumerkel et al 2002). Innerhalb unseres Modells, werden Beschleunigungskräfte nicht als wahrgenommene Größen, sondern als vom Fah-

rer erwünscht modelliert und dienen der Geschwindigkeitswahl beim Planen der weiteren Fahrt.

Weitere menschliche Sinne werden innerhalb des Modells nicht betrachtet, so sind das Schmecken oder das Riechen beim Führen eines Fahrzeugs sicherlich unwichtig. Das gilt ebenfalls für die Aufnahme taktiler Reize, da wir annehmen, dass der Fahrer sein Fahrzeug gut genug kennt und somit das Greifen nach einem Fahrzeugstellregler keine taktile Exploration umfasst. Zwar verhindert das Vernachlässigen taktiler Reize die Behandlung von Rückstellkräften der Fahrzeugregler, die teilweise als Informationsmedium in modernen Fahrerassistenzsystemen eingesetzt werden. Wir hoffen jedoch aufgrund der anfänglichen Fragestellung – der Modellierung normaler Straßenfahrten unter Vernachlässigung extremer Situationen und erweiterter Fahrzeugkonzepte – diese Vereinfachung durchführen zu können, ohne dass die Genauigkeit des Modells stark reduziert wird. Vernachlässigt wird auch die Interozeption, d.h. die Wahrnehmung von Bedürfnissen wie Hunger, Durst etc.

4.3 Zentrale Verarbeitungsinstanz

Eines der größten Probleme bei der Formulierung der Kognition als ein Datenflussmodell ist das Herleiten der dafür notwendigen Informationsarten und Strukturen. Das menschliche Gehirn besteht aus 10^{12} Neuronen (Brause 1991), die über 10^{15} Verbindungen besitzen (Churchland, Sajnowski 1997). Diese Struktur speichert Informationen, weil Neuronen oder Mengen dieser ein bestimmtes elektrisches Potential haben. Sie verarbeitet Informationen jedoch auch durch die Weitergabe der Potentiale an weitere Neurone oder Neuronengruppen über die Verbindungen zwischen diesen. Die neuronalen Strukturen des Gehirns sind weitgehend unerforscht. Somit ist es schwer festzustellen, welche Informationen wo ihren Sitz haben und wie sie verarbeitet werden. Durch die variable Anzahl von Verbindungen scheinen neuronale Informationspräsentationen nicht nur multidimensional, sondern auch in der Anzahl ihrer Dimensionen variabel. So müssen bei der Modellierung sowohl Annahmen zu der Datenverarbeitungskaskade als auch zur Darstellung der innerhalb einer betrachteten Struktur verarbeiteten Informationen getroffen werden, wobei darauf zu achten ist, dass die jeweilige Modellierung das Notwendige umfasst, gleichzeitig jedoch verständlich bleibt.

Die Umsetzung der Verarbeitungsinstanz besteht aus den folgenden, stark miteinander kooperierenden Komponenten: einer interner Weltrepräsentation (IWR), einem Aufmerksamkeitsscheduler, einer hierarchischer Planinstanz und einer Handlungsinstanz. Die Komponenten werden im Folgenden besprochen.

4.3.1 Interne Weltrepräsentation

Die wahrgenommenen Objekte und ihre dynamischen Eigenschaften werden zunächst in ein internes Abbild der Umwelt eingebaut, wo sie für eine Zeit erhalten bleiben, auch wenn das Objekt aus der Sicht verschwindet.

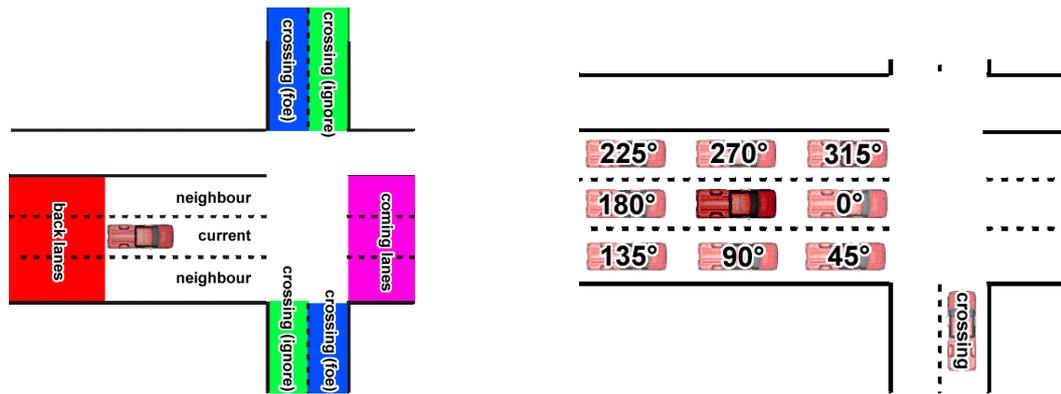


Abbildung 6: Objektklassen in der internen Weltrepräsentation (links: Spurklassen, rechts: Fahrzeugklassen).

Die interne Weltrepräsentation ist als ein Satz von Platzhaltern modelliert. Jeder Platzhalter entspricht einer Objektklasse, die für die Weiterfahrt von Bedeutung ist, so z.B. „aktuelle Spur“, „kreuzende, feindliche Spur“, „Fahrzeug(e) voraus“, „Fahrzeug(e) hinten“ etc. Abbildung 6 zeigt die Platzhalter schematisch, die die interne Weltrepräsentation bilden. Während der Simulation wird ein vom simulierten Fahrer erkanntes Objekt in den passenden Platzhalter eingesetzt, wobei ein Platzhalter mehrfach belegt werden kann, ein Objekt sich aber stets in höchstens einem Platzhalter befindet. Die Grundannahme dieses Modells der IWR ist, dass jedes Objekt in der Umgebung des Fahrers einen bestimmten Einfluss auf ihn hat. So sind die sich vor dem Fahrer befindenden Fahrzeuge für die Wahl der Geschwindigkeit wichtiger als die Fahrzeuge neben oder hinter ihm. Durch eine Kategorisierung der Einflüsse gewinnt man zunächst einen Geschwindigkeitsvorteil bei der Simulation, weil die Bedeutung der gesehenen Objekte nicht wiederholt berechnet werden muss. Zusätzlich scheint diese Sicht auf den Zustand, in dem sich der Fahrer befindet, die Betrachtung dieses Zustandes zu vereinfachen, so dass hier weitere, interessante Untersuchungen zum Thema „Verkehrssituation“ möglich werden.

4.3.2 Planinstanz

Die Planinstanz arbeitet auf den in der internen Weltrepräsentation gespeicherten Werten und einer beim Simulationsstart durch einen Satz von zu überfahrenden Kreuzungen vorgegebenen Route. Die Planinstanz ist an das Drei-Ebenen-Modell von Michon (siehe Ranney 1994) angelehnt, das aus einer Navigations-, einer Bahnführungs- und einer Stabilisierungsebene besteht. Auf der Navigationsebene findet das Verfolgen der Route bis zum Zielort der Fahrt statt, die Bahnführungsebene beinhaltet die Spurwahl, sowohl durch die Navigationsebene als auch durch den Zustand auf der Straße bedingt. Auf der Stabilisierungsebene ist die Regelung der lateralen Spurlage und der erwünschten Geschwindigkeit zu finden. Die vom Fahrer ausgeführten Aktionen sind hierarchisch in diesen Ebenen angeordnet und dienen dem Erreichen des Ziels der jeweiligen Ebene. Normalerweise müssen Aktionen auf tieferen Ebenen an die höher angesiedelten Wünsche angepasst werden.

Die vorgegebene Route bildet vereinfachend die oberste Schicht des Drei-Ebenen-Modells der Fahrzeugführung ab, wobei der bei der Navigation zur Orientierung notwendige, kognitive Aufwand nicht explizit betrachtet wird.

Jedes in der IWR gespeicherte Objekt kann bestimmte Aktionen auslösen. Soll der simulierte Fahrer z.B. an einer Kreuzung abbiegen, so werden die Aufgaben „Brem-

sen“, „Lenkrad drehen“ etc. in ein dynamisches Plankonstrukt eingetragen, das die mittlere, taktische Ebene des Modells nach Michon repräsentiert. Dabei wird angenommen, dass auch diese Aktionen hierarchisch angeordnet sind und nur dann ausgeführt werden können, wenn bestimmte Prämissen erfüllt sind. Diese Annahmen führen zu einer Kaskade von Aktionen, so führt zum Beispiel die Instatiiierung der Aktion „Fahre um Kurve“ zum Setzen der Aktion „Bremse“, weil der Fahrer die Kurve nur mit einer sich aus den von ihm bevorzugten Querschleunigungen ergebenden Geschwindigkeit überfahren will.

Die Stabilisierungsebene wurde nicht als das Ausführen expliziter Aktionen, sondern als ein Regelvorgang implementiert. Die Anpassung an die gewünschte Spurlage erfolgt hier kontinuierlich, mögliche vom Fahrer dabei gemachten Fehler werden bisher nicht berücksichtigt.

4.3.3 Aufmerksamkeitsscheduler

Für Objekte in der IWR wird die Wichtigkeit für die Weiterfahrt bestimmt, die von der Entfernung zu dem jeweiligen Objekt und der Anzahl mit dem Objekt verknüpfter Aktionspunkte der Bahnführungsebene abhängt. Diese Wichtigkeit steuert den Wahrnehmungsprozess, indem die nächste Sakkade auf das aktuell wichtigste Objekt ausgeführt wird.

4.3.4 Handlungsinstanz

Die in der Planungsinstanz gespeicherten Aktionen werden ausgeführt, indem die passende Aktion aus einem Aktionsspeicher herausgelesen und mit vorher abgeschätzten Parametern gestartet wird. Die Aktionen liegen in ihrem Speicher benannt vor, so gibt es jeweils eine Aktion für „Lenkrad drehen“ oder „Beschleunigen“. Dieses Konzept entspricht weitgehend den Annahmen zu Handlungsschemata, welches vom Menschen gelernte Aktionen als schematische Muster im Gehirn vorliegend ansieht, die für die Ausführung passend parametrisiert werden. Nach Elsner (2000) beinhalten Handlungsschemata Informationen über die Vor- und Nachbedingungen der Aktion, die während der Handlung normalerweise aufgenommenen Reize und das motorische Programm der Bewegung.

4.4 Körper

Um die Tot- und Rückstellzeiten der Fahrzeugkontrollen in das Modell aufzunehmen, betätigt der simulierte Fahrer Stellgrößen seines simulierten Fahrzeugs. Die Handlungen werden parametrisiert ausgeführt, d.h. der Fahrer betätigt die Bremse, das Gaspedal oder das Lenkrad mit einer bestimmten Stärke, die sein „Bewusstsein“ aus der aktuellen Situation ableitet. Obwohl die Stellung des betroffenen Reglers berechnet wird, sind die ausführenden Extremitäten an kein Skelett angebunden. Die Implementierung eines Skeletts samt der notwendigen Inversen Kinematik – der Rückführung des Bewegens von Füßen oder Händen auf die Beine bzw. Arme – und das Studium des Verlaufs von Bewegungen erschien uns im Rahmen dieses Projektes nicht durchführbar. Auf in ähnlichen Simulationen benutzte, kommerziell erwerbbar Bibliotheken, wie „Jack“ (siehe EDS 2003), wurde verzichtet.

4.5 Fahrzeugmodell

Im deutschsprachigen Raum finden sich mehrere Publikationen, die detailliert auf die Modellierung der Dynamik von Fahrzeugen eingehen. In Diekamp (1995) wird ein Modell der Längsbewegung vorgestellt, das speziell auf die Problematik des Wechsels von Gängen eingeht. Ammon (1997) beschränkt sich auf die Fahrzeugdynamik, ohne die Fahrerinteraktion zu betrachten, beschreibt jedoch neben der Längs- auch die Querdynamik und die Möglichkeit, die Modellkomplexität zu variieren. Die dritte hier zu nennende Quelle ist das Buch „Kraftfahrzeugführung“ (Jürgensohn 2001), das verstärkt Modelle zur Bewertung von Fahrerassistenzsystemen bespricht.

Da wir uns zumindest zunächst auf Fahrten unter einfachen Bedingungen konzentrieren, können viele komplexe und schwer ermittelbare Parameter eines Fahrzeugs ohne große Qualitätseinbußen vernachlässigt werden. Erweiterungen werden erst bei der Berücksichtigung von schnellen Kurvenfahrten oder zur Bildung von Aussagen über die Auswirkungen vom Zittern, Rütteln etc. des Fahrzeugs, benötigt.

5. Weitere Schritte

Viele der Arbeiten zur Fahrererkennung beruhen auf Experimenten, die unter Laborbedingungen, z.B. in Simulatoren oder auf bestimmten Teststrecken und/oder unter bestimmten Vorgaben durchgeführt worden sind. Es gilt zu prüfen, ob sich solche Ergebnisse in den echten Verkehr übertragen lassen. Es ist anzunehmen, dass eine erneute Kalibrierung der gefundenen Parameter notwendig sein wird, wenn der ursprünglich experimentelle Kontext, in dem die Daten erfasst worden sind, verlassen wird.

Obwohl das Modell ablauffähig ist, muss es daher anhand von Daten, die im Kontext der Fahrzeugführung erhoben worden sind, verifiziert und kalibriert werden. Zunächst werden dafür Videoaufnahmen einer ca. 45min dauernden Fahrt über das WISTA-Gelände in Berlin Adlershof benutzt. Es gilt, die Fahrt möglichst genau nachzubilden – unter Beachtung des Verkehrs auf Hauptstraßen, in die eingefahren wird, der unterschiedlichen Straßentypen und der vorhandenen Hindernisse. Weitere Validierungen mit anderen Daten sollen folgen. Zu diesen Daten gehören Datensammlungen über Fahrzeugfolgeverhalten, die zum Teil über die Internetseiten der Clearingstelle Verkehr (siehe Schulz, Parnitzke & Wagner 2003) erhältlich sind und mit dem ViewCar des DLR (Vollrath 2004) erfasste Daten über das Verhalten eines Autofahrers.

Erst nach einer Validierung des Modells und der gesamten Simulation können speziellere Fragestellungen untersucht werden, wobei – wie bei der Aufnahme taktile Reize besprochen – Erweiterungen des Modells notwendig werden.

Neben dem Fertigstellen des Konzeptes mit der Umsetzung aller als notwendig erkannter Strukturen, soll die Benutzerinteraktion erweitert werden. So soll dem Benutzer die Möglichkeit gegeben werden, eigene Operationen auf Daten innerhalb der Kognition zu spezifizieren, z.B. die Informationsaufnahme mit weiteren Filtern zu versehen. Solche Ansätze könnten z.B. dazu dienen, den Einfluss von Alkohol zu untersuchen.

6. Literatur

- Ammon, D. (1997). *Modellbildung und Systementwicklung in der Fahrzeugdynamik*. Teubner, (Leitfäden der angewandten Mathematik und Mechanik), ISBN 3-519-02378-4 Gb.
- Brause, R. (1991) *Neuronale Netze: eine Einführung in die Neuroinformatik*. Teubner, Stuttgart, ISBN 3-519-02247-8
- Brockfeld, E; Kühne, R. & Wagner, P. (2002). Towards Benchmarking Microscopic Traffic Flow Models, In: *Networks for Mobility, Interntaional Symposium*, September 18-20, 2002, in Stuttgart, Proceedings, Vol I, p. 321-331
- Churchland, P. S.; Sajnowski, T. J. (1997) *Grundlagen zur Neuroinformatik und Neurobiologie*. Vieweg-Verlag, ISBN 3-528-05428-X
- Diekamp, R.(1995). *Entwicklung eines fahrzeugorientierten Verkehrssimulationsprogramms*. Forschungsgesellschaft Kraftfahrwesen Aachen mbH (fka), ISBN 3-925 194-31-2
- Dornhoefer, S.M. (1999). *Blickparameter als Indikatoren der Fahrumgebungserfassung*. Diplomarbeit
- Dörner, D. (2001). *Bauplan für eine Seele*. Rowohl Taschenbuch Verlag AG, ISBN 3-499-61193-7
- EDS. (2003). Internetseiten der „Electronic Data Systems“ (<http://www.eds.com/products/plm/efactory/jack/>)
- Elsner, B. (2000). *Der Erwerb kognitiver Handlungsrepräsentationen*. wvb Wissenschaftlicher Verlag Berlin, ISBN 3932089464
- Epstein, W.; Rogers, S. (1995). *Perception of Space and Motion*. Academic Press Inc.; ISBN 0-12-240530-7
- Fastenmeier, W. (1995). *Autofahrer und Verkehrssituation*. Mensch-Fahrzeug-Umwelt Band 33, Dt. PsychologenVerlag, ISBN 3-925559-80-9
- Hertkorn, G. (2002). „Synthetische Bevölkerung“; In: K. J. Beckmann, G. Rindsfüser, H. Mühlhans, J. Ansorge, U. Brüggemann, N. Eissfeldt, J. Gräfe, G. Hertkorn. *SimVV Mobilität verstehen und lenken – zu einer integrierten quantitativen Gesamtsicht und Mikrosimulation von Verkehr*. Abschlussbericht des SimVV-Projektes, Ministerium für Schule, Wissenschaft und Forschung des Landes Nordrhein-Westfalen
- Jürgensohn, T. (1997). *Hybride Fahrermodelle*. Pro Unniversitate Verlag, ISBN 3-932490-22-3
- Jürgensohn, T. (2001). *Kraftfahrzeugführung*. Springer, ISBN 3-540-42012-6
- Krauß, S. (1998) *Microscopic Modelling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics*. DLR (Hauptabteilung Mobilität und Systemtechnik), ISSN 1434-8454

- Neumerkel, D; Rammelt, P; Reichardt, D & Stolzmann, W. (2002). Fahrermodelle - Ein Schlüssel für unfallfreies Fahren? In: *KI-16*, S. 34-36, Heft 3
- Neisser, U. (1974). *Cognitive Psychology*. Prentice-Hall Inc., Englewood Cliffs, New Jersey, ISBN 0-13-139667-6
- Ranney, T.A. (1994). Models of Driving Behavior: A Review of their Evolution. In: *Accident Analysis and Prevention*, Vol. 26, No. 6, pp. 733-750; Elsevier Science Ltd, 1994
- Schulz, A.; Parnitzke, A. & Wagner, P. (2003). Homepage der Clearingstelle Verkehr unter: <http://www.clearingstelle-verkehr.de/>
- Velichkovsky, B. M.; Challis, B. H. & Pomplun, M.. (1995). Arbeitsgedächtnis und Arbeit mit dem Gedächtnis: Visuell-räumliche und weitere Komponenten der Verarbeitung. In: *Zeitschrift für Experimentelle Psychologie*, Hogrefe-Verlag, Goettingen, 1995, Band XLI, Heft 4, S. 672-701
- Vollrath, M. (2004) *VIEW CAR - Freude am Fahren mit innovativen Assistenzsystemen*. DLR-Nachrichten 106
- Wiedemann, R. (1974). *Simulation des Straßenverkehrsflusses*. In: Heft 8 der Schriftenreihe des IfV, Institut für Verkehrswesen, Universität Karlsruhe

Akteursbasierte Simulation in Virtuellen Welten

TORSTEN ASSELMEYER-MALUGA, PETER FRANK, ANDREAS HOHEISEL, HELGE ROSÉ
& BERTRAM WALTER

Institut FIRST, Fraunhofer Gesellschaft

Schlüsselwörter: Virtuelle Welt, Verteilte Umwelt-Simulationen, User Interaktion, Modell-Server

1. Einführung

Welche Folgen hat die Luftbelastung, die durch industrielle Produktion entsteht, für einen Bauern, der in der Nähe sein Feld bestellt? Wie wirken sich wiederum seine Aktionen auf die nahe gelegenen Wälder und Gewässer aus? Und wie kann man die Interessen verschiedenster Gruppen miteinander abstimmen, so dass gleichzeitig nachhaltige Entwicklungsstrategien entstehen. Wie können die Teilnehmer miteinander lernen, Problemlösungsstrategien entwickeln und verbessern?

Das von uns entwickelte M3(M3=Mensch-Modell-Messung)-Konzept, d.h. die Einbeziehung von Menschen in das Simulationsgeschehen, soll Antworten auf solche Fragen geben. Die M3-Technologie, d.h. die Vernetzung von Umweltmodellen durch die Repräsentanten in einer virtuellen Welt, ermöglicht einerseits die Implementierung einer Vielzahl von verschiedenen, untereinander gekoppelten Simulationsmodellen, welche jeweils bestimmte Bereiche der Umwelt und Wirtschaft simulieren. Beispielfähig implementiert sind das Wachstum von Pflanzen (Landwirtschaftsmodell, Forstmodell), Grund- und Oberflächenwasser (hydrologisches Modell), der Boden und dessen Einfluss auf das Pflanzenwachstum (Bodenmodell), Wetter und die Ausbreitung von Schadstoffen (Wetter- und Luftbelastungsmodell), Verkehr (Verkehrsmodell) und der Einfluss der Umwelt auf den Menschen (Impaktmodelle). Die Ökonomie soll endogen durch die Implementierung eines Wirtschaftssystems in das M3-System, d.h. in die Implementierung des M3-Konzeptes, abgebildet werden. Andererseits beinhaltet die M3-Technologie eine direkte Vernetzung der Teilnehmer durch eine interaktiven web-basierten Simulation. Dort bestimmen die Teilnehmer durch eigene Aktionen am PC die Abläufe und Entwicklungen in der virtuellen M3-Welt. Diese Aktionen haben natürlich auch Auswirkungen auf die virtuellen Repräsentanten der Teilnehmer selbst. So kann eine falsch aufgestellte Chemiefabrik zu einer Erkrankung führen.

Damit bieten sich Einsatzmöglichkeiten in der Untersuchung, Simulation, Erkenntnisvermittlung und im Management von komplexen Alltagssystemen in Wissenschaft, Wirtschaft, Politik, Bildung und Unterhaltung. In der Wissenschaft wird M3 als Integrationsinstrument eingesetzt, in der Wirtschaft als Ausbildungs- und Marketinginstrument. Der Bildung kann es zur Vermittlung komplexer Inhalte dienen, der Unterhaltungsindustrie als Basis einer neuen Generation realitätsbasierter Simulationsspiele.

2. Akteursbasierte Simulation

Die Simulation komplexer Alltagssituationen entwickelt sich derzeit zu einem entscheidenden Instrument der IT-unterstützten Problemlösung in Wissenschaft, Wirtschaft und Gesellschaft. Derzeit wächst das Interesse, interaktive Multiplayer Games in wissenschaftlichen Projekten anzuwenden. Beispiele dafür sind Multiplayer Business Games (Hardmann, van Rossum & van Bolhuis, 1995) und Internetforen für partizipatorisches Guppenlernen und Modellverbesserung (Hare, Gilbert, Medugno, Asakawa, Heeb & Pahl-Wostl, 2001). Diese Varianten setzen sowohl auf ein Mensch-Modell als auch auf Mensch-Mensch Interaktionen. Um dies zu realisieren, benötigt man eine Softwareinfrastruktur, die es ermöglicht ein Netzwerk von Umweltmodellen zu betreiben, die ein virtuelles Abbild der Umwelt realisieren, dass durch eine große Zahl agierender Teilnehmer verändert werden kann. Diese *Akteure* nehmen durch ihre Entscheidungen und Handlungen aktiv an der Simulation teil und können Problemlösungsstrategien entwickeln und testen. Andererseits erleben sie sofort die Konsequenzen ihrer Handlungen als Reaktion der Umweltsimulation und können so in einem evolutionären Lernprozess ihre Strategien verbessern. Dass dies keine Fiktion ist, zeigt eine kürzlich publizierte Untersuchung über Multiplayer On-linegames, die plausibel belegt, dass derartige Spiele wesentliche Aspekte des Verhalten aufweisen, wie es in realen ökonomischen System beobachtet werden kann (Castronova, 2001).

Als eine **Man-Model-Measurement (M3-)**Simulation (Rosé, 2001) oder auch M3-Simulation bezeichnen wir eine Simulation bestehend aus einem Netzwerk aus *Modellen*, die eine Welt simulieren, einer aktuellen Datenbank von *Messdaten* und einer großen Anzahl von realen *Menschen* als agierende und erlebende Teilnehmer.

3. Das M3-System

Das M3-System selbst besteht aus vier Hauptkomponenten: den Multi User Virtual Environment (MUVE), dem Modelserver, der Datenbank und dem Userinterface (Abb. 1).

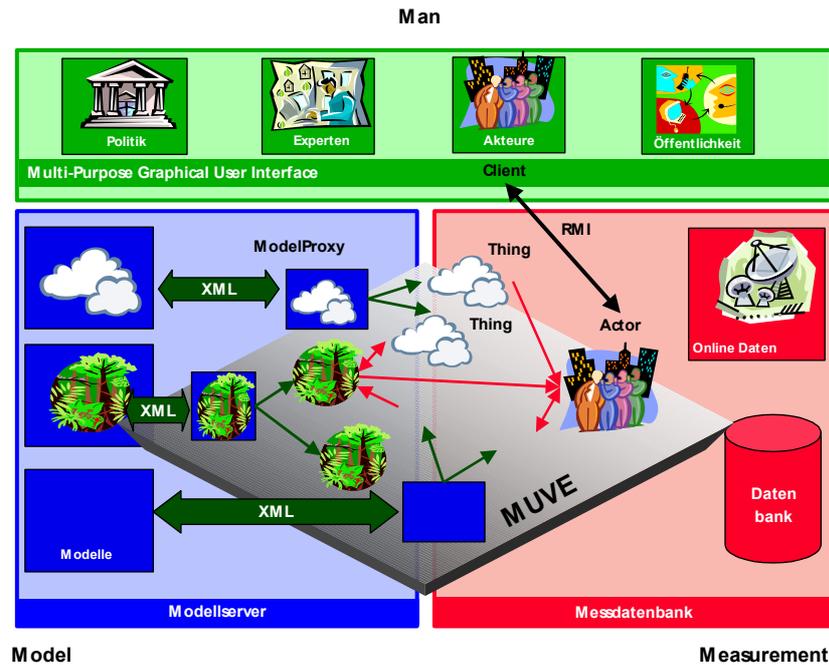


Abbildung 1: Struktur des M3-Systems

3.1 Multi User Virtual Environment

Das MUVE ist die logische Repräsentation des Wirklichkeitsbereiches, der untersucht werden soll und simuliert dessen Struktur durch ein Netzwerk von *Things* (Abb. 1). Die *Things* sind kommunizierende Softwarekomponenten, die die realen Objekte und ihre Wechselwirkungen simulieren. Ein entscheidender Vorteil ist die dadurch mögliche *objekt-orientierte* Modellierung des Realsystems. Jedes *Thing* besitzt hier einen *Zustand*, *Prozesse* der Zustandsänderung und *Relationen*, die seine Wechselwirkungen mit anderen Objekten beschreiben. Das in Java implementierte System realisiert eine Hierarchie von Basisklassen, die die wichtigsten Objekteigenschaften, wie z.B. relational, veränderlich, lokalisiert, sichtbar, beweglich, mechanisch, etc., zur Verfügung stellt. Von diesen elementaren *Things* können dann leicht konkrete Dinge, wie Kornfelder, Häuser, Autos, Pflanzen, etc., abgeleitet werden. Dabei starten wir mit der einfachsten Klasse *Thing*, die durch Implementierung von Interfaces und durch Ableitungen erweitert. Dieses Konzept (Implementierung eines Interfaces und dann Ableitung) ermöglicht es, das Problem der Uneindeutigkeit von Mehrfachableitungen zu umgehen. Die Klasse *Actor* repräsentiert den Spielteilnehmer. Sie stellt die mechanischen und biologischen Eigenschaften, wie z.B. Energieverbrauch, Nährstoffhaushalt, Gesundheitsindex, zur Verfügung. Außerdem implementiert sie die Netzwerkfunktionalität, die die Verbindung zwischen der Actor-Instanz im MUVE und des 3D-Userinterface des Teilnehmers am heimischen PC realisiert. Der derzeitige Prototyp des MUVE simuliert ein vier Quadratkilometer großes Gebiet um den Berliner Alexanderplatz mit einigen charakteristischen Gebäuden, Straßen und computergenerierten Autos die Abgase produzieren.

Externe Umweltmodelle werden in das MUVE durch spezialisierte *Things* integriert. Diese *ModelProxies* werden dynamisch instanziiert und mit den korrespondierenden Zustandsvariablen der *Things* verknüpft. Jedes externe Modell wird durch seinen

eigenen ModelProxy repräsentiert, der die Modelldaten an die Things weiterleitet. Durch dieses Konzept werden die Modelle untereinander durch eine Kommunikationsstruktur verknüpft, die sich aus den Relationen der Things ableitet. Modell und Proxy kommunizieren dabei durch ein XML-basiertes Protokoll (siehe <http://mmm.first.fhg.de/papers/modellkopplung.html> oder http://mmm.first.fhg.de/papers/hoheisel_m3_iemss2002.pdf), dass in unserem Falle eine hohe Flexibilität und leichte Erweiterbarkeit garantiert.

Während der Laufzeit kann durch eine Trennung der Verbindungen des alten ModelProxy und das Knüpfen der Verbindung zu einem anderen Proxy das neue Modell eingefügt werden. Dies ermöglicht die schnelle Evaluation und den Test alternativer Modellimplementierungen ohne Neustart des Gesamtsystems oder Softwareanpassung der Things. Dabei unterstützt der ModellServer den Transfer der internen Zustände zwischen den Modellen und ermöglicht die leichte Integration in das M3-System.

3.2 Der Modellserver

Der Modellserver des M3-Systems realisiert ein Netzwerk von Umweltmodellen, die durch generische Daten- und Steuerschnittstellen verbunden sind. Das allgemeine Problem der Integration heterogener Implementierungen wird damit angegangen. Verfügbare Simulationsmodelle liegen meist in unterschiedlichen Programmiersprachen wie FORTRAN, C oder C++ vor und verwenden diverse Datenformate. Viele dieser Modelle werden als Stand-Alone Anwendungen auf unterschiedlichen Hardwareplattformen genutzt. Bei der Entwicklung dieser Modelle wurde auch meist nicht die Kopplung mit anderen Modellen vorgesehen. Im M3-System nutzen wir daher das Konzept von *Wrappern*, die die externen Simulationsmodelle kapseln und Standardschnittstellen zur Verfügung stellen. Im Gegensatz dazu gibt es noch *ModellProxy*, die der Repräsentant des Modells in der virtuellen Welt sind. Als Beispiel betrachten wir das Landwirtschaftsmodell SWIM. Dieses Simulationsmodell ist in FORTRAN geschrieben und von einem entsprechenden Wrapper umgeben, der sowohl die Programmsteuerung als auch den Datenaustausch mit dem M3-System regelt. Der direkte Repräsentant von SWIM in der virtuellen Welt ist ein Feld, welches im Kontakt zum ModellProxy, als abstrakten Repräsentant, steht. Die Entscheidung fiel dabei auf XML (Extensible Markup Language) als Datenaustauschsprache.

Die in vielen Fällen verwendete enge Kopplung von Modellen z.B. durch Shared Memory erfordert im Allgemeinen einen hohen Integrationsaufwand und erhebliche Änderungen an den verwendeten Modellimplementierungen. Im M3-System werden daher die Modelle durch die Verwendung asynchroner, d.h. auf verschiedenen Zeitskalen stattfindender, Kommunikationen nur lose gekoppelt. Diese Kopplung ist netzwerkorientiert, sehr flexibel und ermöglicht die einfache Wiederverwendung von Modellen in unterschiedlichen Kontexten. Außerdem erfolgt die Ankopplung dynamisch in der Laufzeit, so dass keine starren Kopplungsstrukturen von der Compilierung festgelegt werden müssen. Dies ermöglicht es uns, Modelle während der Simulation auszutauschen ohne die Simulationswelt neu starten zu müssen.

Die Steuerung dieses Netzwerkes aus Simulationsmodellen ist im M3-System durch RMI (Remote Method Invocation) realisiert. Die Modellwrapper stellen einen entsprechenden RMI-Service zur Verfügung, der über das Netzwerk die Modelle starten, stoppen und steuern kann. Dabei werden unter Umständen auch für das Starten

benötigte Informationen mitgeliefert. Wird ein Modell gestoppt, so behalten die noch laufenden Modelle die letzten Werte bis neue geliefert werden können. Die Wrapper nutzen dabei modellseitig die normale Standardein- und -ausgabe des externen Simulationsprogramms zur Kommunikation. Dies ist ein entscheidender Vorteil da jede Programmiersprache eine solche Schnittstelle zur Verfügung stellt.

Die Testversion des M3-Systems implementiert bereits folgende Modelle:

SWIM (Soil and Water Integrated Model) simuliert den Wasserhaushalt, die Vegetation, Erosion und die Nährstoffdynamik ganzer Regionen. Es wurde am Potsdam Institut für Klimafolgenforschung (PIK) entwickelt und in FORTRAN implementiert (Krysanova, Roosaare, Meiner & Vasilyev, 1989).

4C simuliert das Wachstum von Waldbeständen einschließlich der Hydrologie und wurde ebenfalls am PIK entwickelt und in FORTRAN implementiert (Bugmann, Grote, Lasch, Lindner & Suckow, 1997).

WGEN ist ein Wettergenerator von Richardson und Wright. Auf der Basis von statistischen Daten produziert dieses Programm charakteristisches Wetter für eine Region (Richardson & Wright, 1984). Implementierung: FORTRAN.

REWIMET ist ein hydrostatisches drei Schichten Luftschadstoffmodell. Es wurde mit einem Lagrangian Modell zur Simulation von PM10 Rußpartikeln gekoppelt und wird bei FIRST betrieben (Unger, Gerharz, Mieth & Wottrich, 1998). Implementierung: C.

Impact ist ein Modell zur Simulation des Einflusses der Umweltschadstoffe auf die Gesundheit (Beger, Bieninda & Gester, 2001). Implementation: Java.

Traffic ist ein mesoskopisches Verkehrsmodell und wurde in Java implementiert (Schmidt, Schäfer & Nökel, 1998).

3.3 Der Datenbankserver

Der Datenbankserver verwaltet alle Informationen zum Start und Konfiguration der Modelle, der Things und des MUVE. Gleichzeitig protokolliert er alle relevanten Informationen der Simulationsläufe. Dieser netzbasierte Datenbankservice wird durch MySQL realisiert und kann durch eine JDBC-Schnittstelle vom MUVE angesprochen werden. Außerdem ist ein Zugriff und die Verwaltung der Datenbank durch einen WEB-Browser möglich.

3.4 Das 3D-Userinterface

Die von MUVE und Modellserver generierte, virtuelle Welt wird von den Akteuren bevölkert. Dazu nutzten die Teilnehmer ein spezielles 3D-Userinterface, welches ebenfalls in Java geschrieben ist. Für jede Nutzergruppe (Experten, Akteure, Entscheidungsträger, etc.) kann das Interface unterschiedliche *Views* der simulierten Welt liefern, die die für diese Nutzergruppe relevanten Informationen darstellen. Die Implementierung des 3D-Interface' der Akteure ist bereits weit fortgeschritten und stellt die simulierte Welt in einer intuitiven dreidimensionalen Form dar (Abb. 2). Eine genaue Beschreibung der Bedienung etc. findet sich unter <http://mmm.first.fhg.de>.

Zur Visualisierung wird die von uns erweiterte Open Source 3D-Engine Opale Soya verwendet, die ein hardwaregestütztes Rendering über die OpenGL-Schnittstelle erlaubt. Zur Nutzung vorhandener 3D-Objekte wurde ein 3ds-Converter implementiert, der das weitverbreitete 3D Studio Max unterstützt. Die Figuren der Akteure wurden in Poser erstellt. Die anderen Komponenten des 3D-Interface wurden in Java Swing implementiert, wobei das Listener-Konzept (Observer-Pattern) zum Einsatz kam. Die Kommunikation mit dem MUVE erfolgt über RMI-Verbindungen. Damit ruft der Akteur durch eine Aktion eine Methode im MUVE auf, die beim nächsten Zeitschritt ausgeführt wird.

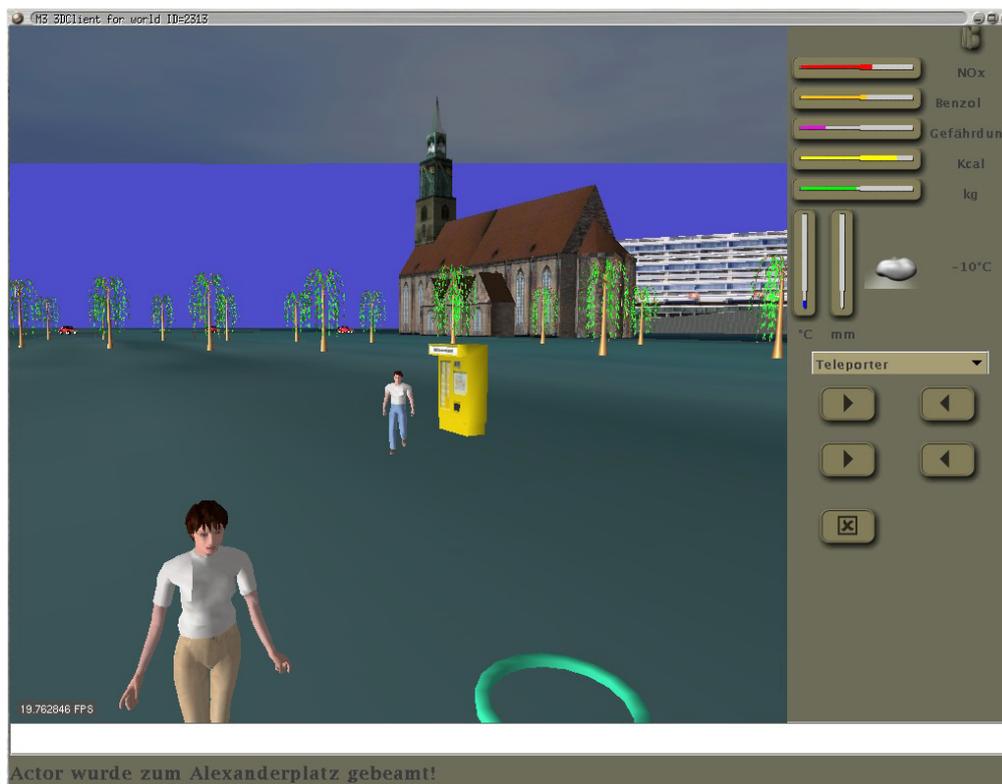


Abbildung 2: Das 3D-Userinterface für die Akteure

Um eine einheitliche Umgebung für alle simultanen Teilnehmer zu realisieren, werden die Positionen aller beweglichen Objekte (Akteure, Autos) im MUVE zentral berechnet. Die neuen Positionen berücksichtigen dabei Kollisionen und Umgebungseinflüsse (z.B. Reibung, Schwerkraft) und werden vom MUVE an die 3D-Interfaces der Akteure gesendet, die dann alle sichtbaren Objekte darstellen.

4. Ausblick

Das M3-Projekt ist ein sehr anspruchsvolles Unterfangen. Zweifellos kann der hohe Aufwand, der zu seiner Verwirklichung notwendig ist, nicht von einzelnen Teams von Informatikern, Umweltwissenschaftlern oder Modellierern erbracht werden. Vielmehr ist eine weitreichende Kooperation aller dieser Entwicklungsgruppen eine unabdingbare Voraussetzung für die erfolgreiche Umsetzung der Vision. Das M3-System selbst, bietet dafür eine geeignete Kommunikationsplattform. Aufgrund seines interaktiven Charakters kann es als seine eigene Entwicklungsumgebung dienen, die eine schnelle Kommunikation und Kooperation sichert. Das M3-System ist nicht

bloß eine „Software“ – es ist eine simulierte Welt, bevölkert von realen Menschen und hat daher die Fähigkeit sich selbst weiter zu entwickeln. Diese Art der *evolutionären Softwareentwicklung* macht den Erfolg von Open Software Projekten wie Linux oder GNU aus (Raymond, 2001).

Alle Nutzer können durch neue Implementierungen zur Entwicklung des Systems beitragen. Zur Weiterentwicklung des Systems werden mehrere „Testwelten“ gestartet, die jeder durch eigene Beiträge beliebig verändern kann. Änderungen, die sich hier bewähren und von einer Jury als funktional, stabil und nützlich bewertet werden, werden dann in die offizielle M3-Welt übernommen. Diese Art der Entwicklung ist in der Tat ein evolutionärer Prozess und eine Studie an der Harvard Business School belegt, dass diese Art der Softwareentwicklung die größten Erfolge verspricht (McCormack, 2001).

Der derzeitige Prototyp des M3-Systems zeigt die prinzipielle Machbarkeit des M3-Konzeptes. Solche Simulationssysteme können dabei durchaus in der Lage sein, realistische Aspekte von wirtschaftlichen Gegebenheiten nachzubilden. Es bleibt allerdings zu untersuchen, ob auch komplexe sozio-ökonomische Systeme hierdurch beschreibbar werden und wie dabei das diffizile Problem der Kopplung verschiedener Modelle befriedigend zu lösen ist. In jedem Fall stellt der Einsatz des M3-Systems in einer wohldefinierten und limitierten Problemsituation ein neues und vielversprechendes Instrument im Verstehen, Lernen und Lösen all jener komplexen Situationen dar, die unser alltägliches Leben für uns bereit hält.

5. Literatur

Beger, E.; Bieninda, D. & Gester, S. (2001). *Modul Impactmodelle im M3-System des Projektes GLOBALSIM*. Interner Technischer Bericht, IBB Ingenieurbüro Beger, 2001.

Bugmann, H.; Grote, R.; Lasch, P.; Lindner, M. & Suckow, F. (1997). A new forest gap model to study the effects of environmental change on forest structure and functioning. In G.M.J. Mohren and K. Kramer, editors, *Global Change Impacts on Tree Physiology and Forest Ecosystems*, pages 255--261. Kluwer Academic Publishers, 1997.

Castronova, E. (2001). *Virtual worlds: A first-hand account of market and society on the cyberian frontier*. CESifo working paper No. 618, submitted to netnomics, 2001.

Hardman, L.; van Rossum, G. & van Bolhuis, A. (1995). *An interactive multimedia business game*. *Journal of Intelligent Systems*, 5, 151--177, 1995.

Hare, M.; Gilbert, N.; Medugno, D.; Asakawa, T.; Heeb, J. & Pahl-Wostl, C. (2001). The development of an internet forum for long-term participatory group learning about problems and solutions to sustainable urban water supply management. In L.M. Hilty and P.W. Gilgen, editors, *Sustainability in the Information Society*, pages 743--750, Marburg, Metropolis Verlag, 2001.

Krysanova, V.; Roosaare, J.; Meiner, A. & Vasilyev, A. (1989). *Simulation modeling of the coastal waters pollution from agricultural watershed*. *Ecological Modeling*, 49, 7--29, 1989.

MacCormack, A. (2001). *Product-development practices that work: How internet companies build software*. MIT Sloan Management Review, 42, 75--84, 2001.

Raymond, E.S. (2001). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. OReilly & Associates, 2001.

Richardson, C.W. & Wright, D.A. (1984). *Wgen: A model for generating daily weather variables*. Technical report, U. S. Department of Agriculture, Agricultural Research Service, 1984.

Rosé, H. (2001). M3-Simulation - Multidimensional Modeling of Sustainability Strategies in Virtual Worlds. In K. Bellmann and C. Landauer, editors, *Proceedings of the International Conference on Virtual Worlds and Simulation*, Phoenix, Arizona, 2001.

Schmidt, M.; Schäfer, R.-P. & Nökel, K. (1998). SIMTRAP: *Simulation of traffic-induced air pollution*. TRANSACTIONS of SCS, 15, 122--132, 1998.

Unger, S.; Gerharz, I.; Mieth, P. & Wottrich, S. (1998). *HITERM - high-performance computing for technological risk management*. TRANSACTIONS of SCS, 15, 109--114, 1998.

Serviceteil

Modellierung und Simulation für Mensch-Maschine-Systeme

Schlüsselwörter: Modellierungswerkzeuge, Bücher, Zeitschriften, weitere Web Ressourcen, Konferenzreihen, Lehrveranstaltungen, Institute und Arbeitsgruppen

Modellierungswerkzeuge

ACT-R

<http://act-r.psy.cmu.edu/>

APEX

<http://www.andrew.cmu.edu/user/bj07/apex/index.html>
<http://human-factors.arc.nasa.gov/apex/index.html>

EPIC

<http://www.eecs.umich.edu/~kieras/epic.html>

COGENT

<http://cogent.psyc.bbk.ac.uk/>

Cognet/iGEN

<http://www.cognitiveagent.com/>

MicroSAINT

http://www.maad.com/index.pl/micro_saint

SOAR

<http://sitemaker.umich.edu/soar>

Bücher

John R. Anderson & Christian Lebiere (1998). *The Atomic Components of Thought.* Mahwah, NJ: Lawrence Erlbaum.

Dieses Buch ist das derzeit gültige Referenzwerk die Modellierung mit der kognitiven Architektur ACT-R. Gegenstand ist ACT-R in der Version 4. Es ist anhand von kognitiven Funktionen gegliedert. Für die Modellierung in angewandten Mensch-Maschine-System-Kontexten von besonderer Bedeutung ist das Kapitel *Perception and Action* von Byrne und Anderson, in dem das ACT-R/PM-System vorgestellt wird.

Richard P. Cooper (2002). *Modelling High-Level Cognitive Processes.* Mahwah, NJ: Lawrence Erlbaum.

In diesem Buch wird die Modellierung kognitiver Vorgänge anhand des Systems COGENT beschrieben. COGENT liefert einen Satz von Bausteinen, die in kognitiven Modellen nützlich ein können, liefert aber keine Einschränkungen bezüglich derer Kombination und Parametrisierung. Das Buch hat drei Teile: Im ersten Teil werden Grundlagen und Hintergründe thematisiert, im zweiten Teil werden eine Reihe von Modellierungsprojekten vorgestellt, die unterschiedliche Bereiche abdecken. Im dritten Teil werden weitergehende Probleme der Fallstudien des zweiten Teils erörtert.

Gunnar Johannsen (1993). *Mensch-Maschine-Systeme.* Berlin: Springer

Ein inzwischen etwas in die Jahre gekommenes Lehrbuch zur Einführung in Mensch-Maschine-System. Das Buch befasst sich mit den ergonomischen, arbeitspsychologischen, systemtechnischen, regelungstechnischen und softwaretechnischen Grundlagen des Zusammenwirkens von Mensch und Maschine. Es richtet sich an Studenten der Ingenieurwissenschaften, der Informatik und der Arbeitspsychologie und erlaubt eine selbständige Einarbeitung in die Grundlagen und Anwendungen der Mensch-Maschine-Systeme. Orientiert sich in der Darstellung jedoch weitgehend an den Mensch-als-Regler-Ansätzen.

Alan Newell (1990). *Unified Theories of Cognition.* Cambridge, MA: Harvard University Press.

Ein Klassiker: ein aus der Sicht eines der Wegbereiter der Kognitionswissenschaft geschriebenes Plädoyer für die architekturgebundene kognitive Modellierung. Liefert auf ca. 550 Seiten empirische, formale wissenschaftstheoretische und forschungsstrategische Argumente für die Idee einer vereinheitlichten Theorie der Kognition. Die Beispiele orientieren sich stark an der kognitiven Architektur SOAR, die dahinter stehenden Konzepte sind auf die meisten kognitiven Architekturen übertragbar.

Ute Schmid & Martin C. Kindsmüller (1996). *Kognitive Modellierung: Eine Einführung die logischen und algorithmischen Grundlagen.* Heidelberg: Spektrum Verlag.

Lehrbuch für eine einsemestrige Einführungsveranstaltung. Zielgruppe sind Interessenten (etwa aus der Psychologie und Linguistik), die Interesse an der Modellierung kognitiver Prozesse mit Methoden der Informatik haben. Grundlagen der Mengenlehre und Logik sowie Algorithmen und Programmierung werden im Kontext kognitiver Theorien (Wissensrepräsentation, Schlussfolgern, Problemlö-

sen, Lernen) eingeführt. Der Schwerpunkt liegt auf der Vermittlung von Basiswissen aus den Teilgebieten der Mengentheorie, Logik und Informatik, die für das Verständnis der Prinzipien der kognitiven Modellierung essentiell sind.

Klaus-Peter Timpe, Thomas Jürgensohn & Harald Kolrep (2000). *Mensch-Maschine-Systemtechnik. Konzepte, Modellierung, Gestaltung, Evaluation*. Düsseldorf: Symposion Publishing

Eine Einführung in die nutzerorientierte Gestaltung von Mensch-Maschine-Systemen. Das fachübergreifend angelegte Buch richtet sich in erster Linie an Ingenieure, Informatiker, Psychologen und Arbeitswissenschaftler die in den Bereichen der Prozess- und Fahrzeugführung, der Luftfahrt und der Produktentwicklung mit dem Aufgabenbereichen Automatisierungstechnik, Systembewertung und Sicherheitstechnik befasst sind. Auch als Einstieg in die systemtechnische Betrachtungsweise von Mensch-Maschine-Systemen für Studierende ingenieurwissenschaftlicher, informationstechnischer oder arbeits- und kognitionspsychologischer Studiengänge geeignet.

Frank E. Ritters Beitrag in diesem Sonderband enthält in den Abschnitten 3.1 bis 3.4. eine Reihe ausführlich kommentierter Empfehlungen zu Lehrbüchern, Standardwerken und wichtigen Zeitschriftenveröffentlichungen aus dem Bereich Modellierung und Simulation.

Frank E. Ritter (2004). *Choosing and getting started with a cognitive architecture to test and use human-machine interfaces* *MMI-Interaktiv*, 7, Juni 2004. S 17-37

http://useworld.net/servlet/handlearticle?obj_id=1444&cat_id=40&vie=true

Zeitschriften

Cognitive Systems Research

Das Journal zielt auf eine verbesserte Analyse und Gestaltung kognitiver und intelligenter Systeme durch auf die Integration von Konzepten, Theorien und Techniken aus unterschiedlichen Forschungsdisziplinen und Forschungsansätze ab. Ergebnisse nicht interdisziplinär Forschung aus der Psychologie oder der künstlichen Intelligenz werden ebenfalls berücksichtigt, wenn die Arbeiten einen weiter gefassten Fokus aufweisen. In diesem Journal wurden bereits mehrfach Berichte zur Modellierung menschlichen Verhaltens in Mensch-Maschine Systemen veröffentlicht.

Ergonomics

Ergonomics ist neben *Applied Ergonomics* und *Human Factors* eines der high impact journals im Ergonomiebereich. Behandelt werden sämtliche Aspekte der Interaktionen von Menschen am Arbeitsplatz. In diesem Rahmen behandeln die Journalbeiträge inhaltlich u.a. psychologische Aspekte der Systemgestaltung mit der Zielstellung, Leistung und Sicherheit und Gesundheit von Mensch bei der Interaktion mit Maschinen zu verbessern. Forschungsberichte zur Modellierung im Bereich der Mensch-Maschine Interaktion werden veröffentlicht, finden sich bisher aber noch eher selten.

User Modeling and User-Adapted Interaction. The Journal of Personalization Research

Das Journal User Modeling and User-Adapted Interaction (UMUAI) bietet ein interdisziplinäres Forum für die Veröffentlichung neuer Forschungsergebnisse über interaktive Computersysteme, die sich an ihre jeweiligen Benutzer anpassen oder anpassen lassen, und über die Rolle von Benutzermodellen im Prozess der Anpassung. UMUAI erscheint vierteljährig seit 1991.
<http://umuai.informatik.uni-essen.de/>

Weitere Web Ressourcen

useworld.net: Methoden::Modellierung & Simulation

http://useworld.net/servlet/listcat?cat_id=101

Konferenzreihen

ACT-R Workshop

Jährliches Treffen der Anwender der kognitiven Architektur ACT-R im Anschluss an die ACT-R Summer-School zumeist in Pittsburgh. Neben anderem werden immer einige Mensch-Maschine-System-relevante Modelle und Architekturweiterungen vorgestellt.
<http://act-r.psy.cmu.edu/workshops/>

BRIMS – Behavior Representation in Modeling and Simulation (CGF-BR Computer Generated Forces and Behavioral Representation)

Diese jährliche Veranstaltung der Simulation Interoperability Standards Organization (SISO) stellt Kognitionswissenschaftlern ein Forum zum Austausch über die Anwendung neuester Methoden zur Verhaltens-Repräsentation zur Verfügung.
<http://www.sisostds.org/cgf-br/index.htm>

CHI: Conference on Human Factors in Computing Systems

Jährliche internationale und interdisziplinäre Konferenz der ACM Special Interest Group on Computer-Human Interaction. Es gibt oft einige Beiträge, die sich mit Modellierung und Simulation in Mensch-Maschine-Systemen beschäftigen. Der Schwerpunkt liegt aber bei interaktiven Softwaresystemen.
<http://www.acm.org/sigchi/conferences/>

Cognitive Science Society Annual Meetings und EuroCogSci

In dieser internationalen Fachgesellschaft sind Kognitionswissenschaftler aus unterschiedlichen Disziplinen organisiert: Künstliche Intelligenz, Linguistik, Anthropologie, Psychologie, Neurowissenschaft, Philosophie und Bildungswesen. Beim jährlichen Treffen, das seit 2003 auch einen europäischen Ableger, die EuroCogSci, hat, werden u.a. auch kognitionswissenschaftlich orientierte Modelle und Modellierungsmethoden präsentiert.
<http://www.cognitivesciencesociety.org/cogsci.html>

DHM – SAE Digital Human Modeling for Design and Engineering

Internationale jährliche Konferenz der Society of Automotive Engineers (SAE) mit dem Schwerpunkt auf anthropometrischen Mensch-Modellen. Neben dem Automobil- werden auch andere Anwendungsbereiche adressiert.
<http://www.sae.org/events/dhm/>

ICCM – International Conference on Cognitive Modelling (ECCM – European Conference on Cognitive Modelling) (EWCM – European Workshop on Cognitive Modelling)

Jährliche internationale Konferenz mit dem Schwerpunkt “*computational cognitive modeling*“. Forum für Kognitionswissenschaftler, die ausführbare kognitive Modelle entwickeln und gegen empirische Daten testen. Ein Schwerpunkt liegt bei den kognitiven Architekturen. Es gibt zunehmend mehr Beiträge zur Anwendung in Mensch-Maschine-Systemen.

2004 (Pittsburgh): <http://simon.lrdc.pitt.edu/~iccm/>

2003 (Bamberg): <http://iccm2003.ppp.uni-bamberg.de/>

2001 (Fairfax): <http://www.lrdc.pitt.edu/schunn/iccm2001/overview/overview.html>

2000 (Groningen): <http://tcw2.ppsw.rug.nl/iccm/>

1998 (Nottingham): <http://www.psychology.nottingham.ac.uk/staff/ritter/eccm98/>

1996 (Berlin): <http://ki.cs.tu-berlin.de/EuroCog/eurocog.html>

SOAR Workshop

Jährlicher Workshop der Benutzer der kognitiven Architektur SOAR. Es werden neue Anwendungen, Architektur Erweiterungen und Werkzeuge vorgestellt.
http://sitemaker.umich.edu/soar/soar_workshops

UM – User Modeling

Die zweijährliche Konferenz von User Modeling Inc. ist *das* Forum für die Präsentation und Diskussion neuester Entwicklungen in der akademischen Forschung über Benutzermodelle und adaptive Systeme und deren industriellen Anwendung. Sie ist interdisziplinär und zieht Beiträge aus den Bereichen Künstliche Intelligenz, Linguistik, Psychologie und Human-Computer Interaction an.
<http://bistrica.usask.ca/UM/conferences.htm>

Leider gibt es keine deutschsprachigen Konferenzreihen, die das Thema Modellierung und Simulation von Human Factors in Mensch-Maschine-Systemen zum Schwerpunkt haben. Auf den folgenden Tagungsreihen gibt es jedoch regelmäßig einige entsprechende Beiträge:

Berliner Werkstatt Mensch-Maschine-Systeme

Das Zentrum Mensch-Maschine Systeme der TU Berlin veranstaltet in zweijährigem Turnus die „Berliner Werkstatt Mensch-Maschine-Systeme“. Auf diesen Werkstätten werden aktuelle Themen aus dem Bereich der Mensch-Maschine-Systeme vorgestellt und diskutiert.

<http://www.zmms.tu-berlin.de/events/BWMMS/>

Fachausschusssitzung Anthropotechnik

Der Fachausschuss Anthropotechnik der Deutschen Gesellschaft für Luft- und Raumfahrt e.V. führt jährlich Ende Oktober eine zweitägige Sitzung durch, bei

der ein aktuelles Thema unter ergonomischen Gesichtspunkten diskutiert wird. In diesem Rahmen wird eine Beteiligung sowohl der Industrie als auch der in Universitäten und Forschungseinrichtungen angesiedelten Forschung angestrebt, womit ein Wissenstransfer und die Umsetzung wissenschaftlicher Erkenntnisse in den Produkten erzielt werden soll.

<http://www.dglr.de/gliederung/fachbereiche/techno/t5/t54/index.htm>

Fachtagung USEWARE

Die VDI/VDE -GMA richtet zusammen mit dem Zentrum Mensch-Maschine-Interaktion der TU Kaiserslautern zweijährlich versetzt zur Berliner Werkstatt Mensch-Maschine-Systeme die Useware-Tagung aus. Sie widmet sich den aktuellen Trends im Bereich der nutzergerechten Technikgestaltung, den neuesten Methoden, Konzepten und Forschungsergebnissen. Die Tagungsreihe ist praxisnah und stellt eine Brücke zwischen Wissenschaft und Anwendung in unterschiedlichsten Bereichen, von der Produktions- und Verfahrenstechnik, über die Fahrzeugtechnik bis zu Konsumanwendungen dar.

<http://www.uni-kl.de/pak/useware2002/>

<http://www.uni-kl.de/pak/useware2004/>

Mensch & Computer

Jährliche Konferenz des Fachbereiches Mensch-Computer-Interaktion in der Gesellschaft für Informatik, die verschiedene Fachgebiete und Praxisfelder in einen fruchtbaren Diskurs zu bringen versucht.

<http://www.mensch-und-computer.de/>

Symposium Simulationstechnik (ASIM Jahrestagung)

Am Vortag werden Anwendertreffen für einzelne Simulationswerkzeuge angeboten, die dem Erfahrungsaustausch dienen und den Kontakt unter den Nutzern fördern. Die Jahrestagung wird von einer Ausstellung begleitet. Durch das Setzen fachlicher Schwerpunkte an einzelnen Tagen lassen die ASIM-Jahrestagungen auch die Teilnahme an einem oder zwei Tagen zu. Um Beiträge aus der Industrie zu vereinfachen, werden Praxisforen ohne Veröffentlichungszwang organisiert.

http://www.asim-gi.org/veranstaltungen/veranstaltungen_frame.html

Lehrveranstaltungen



*Lehrstuhl und Institut für Arbeitswissenschaft,
RWTH Aachen*

Simulation
ereignis-
diskreter
Systeme

*VL mit
Übungen*

Dozent: Introduction to Discrete Event Systems, Languages and Automata, Statecharts, Petri Nets: Foundations and Analysis, Timed Models, Stochastic Timed Automata, Markov Chains, Queueing Models, Bayesian Networks, Dynamic Bayesian Networks, Variable Length Markov Chains, Event Scheduling Scheme and Output Analysis.
Schlick
Siehe auch: <http://www.iaw.rwth-aachen.de/download/lehre/vorlesungen/2003-ws-ses/ses-ws-2003-skript.pdf>

Homepage: <http://www.iaw.rwth-aachen.de/lehre/vorlesungen/2003-ws-ses/index.html>

*Lehrstuhl für technische Informatik,
RWTH Aachen*

Anthropotechnik in der Fahrzeug- und Prozessführung

VL mit Übung

Blockveranstaltung

Dozenten: Die Vorlesung ist Teil des Hauptdiploms für Studenten des Maschinenbaus und wird vor allem von Studierenden der Vertiefungsrichtung Luft- und Raumfahrttechnik gehört. Sie behandelt folgende Themen: Einführung in die Anthropotechnik; Sensorische, motorische und kognitive Verhaltensmuster von Piloten; Mathematische Modellierung der manuellen Regelung; Informationsverarbeitung; Entscheidungsverhalten; Cockpit Design und Konfiguration; Sprachsteuerung; Zuverlässigkeit menschlicher und technischer Systeme; Benutzergerechte Automatisierung in der Luftfahrt; Flug-Management Systeme (FMS); Bord-Diagnosesysteme (ECAM); Stand und Entwicklungstendenzen der Automatisierungstechnik in der Luftfahrt; Evaluierung des Cockpits; Aufgabenteilung Pilot-Automatik; Messung der Arbeitslast; Experimentelle Analyse; Rapid Prototyping; Simulation von Flugmissionen; Versuchsplanung; Versuchsauswertung.

Homepage: <http://www.iaw.rwth-aachen.de/lehre/vorlesungen/2003-ws-ses/index.html>

*Lehrstuhl für technische Informatik,
RWTH Aachen*

Mensch-
Maschine
Systeme I

*VL mit
Übung*

2 SWS

Dozent: Systematik und Entwicklung von MMS: Systematik der Mensch-Maschine Systeme, Entwicklung von MMS, Perspektiven des Fachgebiets; Systemkomponente Mensch, Verhaltensmodelle, Informationsaufnahme, Informationsverarbeitung, Informationsausgabe, Handlungszuverlässigkeit und Fehlverhalten; Technik der Mensch-Maschine Kommunikation: Schnittstellentechnik, Fenstersysteme (GUIs: Graphical User Interfaces), Multimediale Kommunikation, Virtuelle Umgebungen, Sprachkommunikation, Kommunikation über Gesten, Gebärden und Mimik, Multimodale Kommunikation; Mensch-Maschine Systemtechnik, Dialogsysteme, Dynamische Systeme, Assistierende Systeme, Zuverlässigkeit von MMS; Gestaltung und Bewertung von MMS: Gestaltungsziele und Randbedingungen, Gestaltungsrichtlinien, Bewertungsverfahren, Durchführung empirischer Untersuchungen, Formale Bewertungsverfahren.

Homepage: <http://www.campus.rwth-aa-chen.de/rwth/all/event.asp?gguid=0x027EBA52FE3BD611BDB90002A5871170&hideall=true>

*Lehrstuhl für technische Informatik,
RWTH Aachen*

Mensch-
Maschine
Systeme II

*VL mit
Übung*

2 SWS

Dozent: Im Hauptstudium wird die zweisemestrige Vorlesung "Mensch-Maschine Systeme" angeboten (Wahlpflichtfach für Studierende der Studienrichtung IK). Die Vorlesung ist interdisziplinär ausgerichtet und vermittelt den Studierenden in MMS II folgende Inhalte: Teil 1: Lernen aus Daten: Einführung; Datenvorverarbeitung und Merkmalsextraktion: Hauptachsentransformation (PCA), Lineare Diskriminanzanalyse (LDA); Statistische Verfahren: Entscheidungsgrenzen und Diskriminanzfunktionen, Schätzung stochastischer Modelle; Konnektionistische Verfahren: Grundlagen, Einschichtige Netze, Mehrschichtige Netze, Lernen als Optimierungsproblem, Support Vector Machines, Radiale Basisfunktionen, Clustering-Verfahren Teil 2: Wissensrepräsentation und Wissensverarbeitung: Einführung; Aussagenlogik: Einige Grundlagen, Automatische Verarbeitung aussagenlogischer Formeln; Prädikatenlogik erster Stufe (PL1): Einige Grundlagen, Automatische Verarbeitung prädikatenlogischer Formeln, Prädikate mit Zeitbezug, Logik-Programmierung (PROLOG); Expertensysteme: Architektur, Inferenz mit Regeln, Suchverfahren in Regelmengen, Wissensakquisition, Implementierungsaspekte; Verarbeitung unsicheren Wissens: Bayes-Theorem, Bayesian Belief Networks (BBNs), Sicherheitsfaktoren, Dempster-Shafer-Theorie; Fuzzy-Systeme: Repräsentation unscharfen Wissens, MIN-MAX-Inferenz mit unscharfen Regeln, Fuzzy-Regelsysteme, Neuro-Fuzzy Systeme.

Kraiss

Homepage: <http://www.campus.rwth-aa->

[chen.de/rwth/all/event.asp?gguid=0x3B55704A0BFBEA4F93C1DD90F1FBF403&tguid=0x28B907D5A90A0E45A396C345BE4C621A](http://www.campus.rwth-aa-chen.de/rwth/all/event.asp?gguid=0x3B55704A0BFBEA4F93C1DD90F1FBF403&tguid=0x28B907D5A90A0E45A396C345BE4C621A)



*Methoden der
Künstlichen Intel-
ligenz (KI), Institut
für Softwaretech-
nik und Theoreti-
sche Informatik,
TU Berlin*

*Zentrum Mensch-
Maschine Systeme
(ZMMS); TU Ber-
lin*

Künstliche
Intelligenz
für Geistes-
& Sozialwis-
senschaften

Dozenten: Einführung in die kognitive Modellierung. Mengentheoretische, logische & algorithmische Grundlagen zur Formalisierung kognitiver Strukturen und Prozesse. Die Veranstaltung wendet sich an Studierende (etwa der Psychologie, Philosophie oder Linguistik), die Interesse an der Modellierung kognitiver Prozesse mit Methoden der Informatik haben. Grundlagen der Mengentheorie und Logik sowie Grundkonzepte der Informatik, Algorithmen und Programmierung werden im Kontext kognitiver Theorien (Wissensrepräsentation, Schlussfolgern, Problemlösen, Lernen) eingeführt.

VL

2 SWS

Themen: Wissensrepräsentation und Logik; Mengenlehre, Aussagen- und Prädikatenlogik, Semantische Netze, Modellierung in PROLOG; Problemlösen und Produktionssysteme Algorithmen, Automaten und formale Sprachen, Problemrepräsentation, heuristische Suchverfahren, Produktionssysteme; Anwendungsbereiche kognitionswissenschaftlicher Methoden:; Bildverstehen, Sprache, Lernen, Expertise. Wir werden vor allem Gewicht auf die grundlegende Einführung von oben genannten Formalismen legen. Die Darstellung und Diskussion des philosophischen Hintergrunds kognitionswissenschaftlicher Forschung, sowie konkreter kognitiver Modelle wird jedoch so gut wie möglich in die Veranstaltung integriert. Einige Themen werden wir nur streifen, einige Themen vertiefen. Bei der Auswahl der Themenschwerpunkte wie auch bei der Gestaltung der interaktiven Anteile der Lehrveranstaltung soll Raum bleiben, um auf die Vorstellungen und das Vorwissen der Studierenden einzugehen.

Homepage: <http://ki.cs.tu-berlin.de/lehre.html>



*Zentrum Mensch-
Maschine Systeme
(ZMMS)
TU Berlin*

Modell-
bildung und
Simulation
in Mensch-
Maschine-
Systemen I
(MuSiMMS
I)

Dozenten: Die Veranstaltung behandelt die Nachbildung des dynamischen Verhaltens von Bedienern in Mensch-Maschine-System-Umgebungen. Beginnend mit einer Einführung in die Modellbildung werden deskriptive Methoden zur quantitativen Modellbildung des Menschen in unterschiedlichen dynamischen Umgebungen vorgestellt. Diese Methoden werden an aktuellen Daten eingesetzt und vergleichend bewertet.

Urbas, Leuchter

*Integrierte
Veranstal-
tung*

4 SWS

Homepage: <http://www.zmms.tu-berlin.de/modys/studium/lehre/MuSiMMS/index.html>



*Zentrum Mensch-
Maschine Systeme
(ZMMS)
TU Berlin*

Modell-
bildung und
Simulation
in Mensch-
Maschine-
Systemen II
(MuSiMMS
II)

*Integrierte
Veranstal-
tung*

4 SWS

Dozenten: Im Mittelpunkt der Veranstaltung stehen Theorien der menschlichen Informationsverarbeitung sowie ihre Umsetzung und Anwendung in Simulationsmodellen. Ausgehend von kognitionswissenschaftlichen Grundlagen werden die Studierenden zu aktuellen Modellierungsansätzen herangeführt. Beginnend mit psychologischen Theorien der Handlungstheorie werden Beispiele der quantitativen Modellbildung des Menschen in unterschiedlichen dynamischen Umgebungen vorgestellt. Im Vordergrund stehen dabei Modelle von Piloten, Autofahrern und Wartenbedienern. In der integrierten LV werden die Teilnehmer an Methoden der Arbeitswissenschaft (hierarchischen Aufgabenanalyse), der Human-Computer-Interaction (GOMS) und der Kognitionswissenschaft (ACT-R/PM) herangeführt, die dann in einem Modellierungsprojekt eingesetzt werden. In einem parallelen Seminaranteil werden Themen zur Modellierung von kognitiven Leistungen in Kognitionspsychologie, Arbeitswissenschaft und Human-Computer-Interaction vertieft.

Homepage: http://www.zmms.tu-berlin.de/modys/studium/lehre/MuSiMMS_SS04/index.html

*Institut für Mess-
und Automatisie-
rungstechnik (I-
MAT); Universität
Kassel*

Mensch-
Maschine-
Systeme 1
(MMS 1)

*VL mit
Übung*

2 + 1 SWS

Dozent: Mensch-Maschine-Systeme: ein interdisziplinäres Fachgebiet für die industrielle Prozess-, Produktions- und Fahrzeugführung; Prozesse menschlicher Informationsverarbeitung; Analyse und Gestaltung von Mensch-Maschine-Systemen (Einführung) ; Regelungs- und Überwachungstätigkeiten des Menschen in Mensch-Maschine-Systemen; Experimentelle Untersuchungen; Kritische Regelungsaufgabe; Modelle für den Menschen als Regler und Überwacher ; Mensch-Maschine-Kommunikation (Grundlagen, graphische Benutzerschnittstellen); Fehlermanagement in Mensch-Maschine-Systemen; Belastung und Beanspruchung des Menschen

Homepage: <http://www.imat.maschinenbau.uni-kassel.de/lehre/lehr0304.html#MMS1>



*Fakultät für Luft-
und Raumfahrt-
technik, Universi-
tät der Bundes-
wehr München*

Flug-
mechanik /
Anthropo-
technik

LV

Dozent: Darstellung der Wirkungsschleifen der Flugmechanik/Flugführung mit besonderer Herausstellung der Rolle des Piloten zur Durchführung der Flugführungs- und Systemüberwachungsaufgaben. Aufgabenanalyse. Modelle für das Verhalten des Menschen bei der Wahrnehmung und der Informationsverarbeitung (Kognition) sowie als Effektor. Berücksichtigung psychologischer Faktoren. Modelle des menschlichen Fehlverhaltens, verursacht durch eigene Unzulänglichkeiten und/oder durch äußere Einflüsse. Bestimmung von automationsrelevanten Funktionen und personalen Aufgaben. Schnittstellenorientierte Analyse- und Bewertungsverfahren.

Homepage: <http://www.unibw-muenchen.de/campus/LRT/LRT13/deutsch/index.html>



*Lehrstuhl für
Mensch-Maschine-
Kommunikation;
TU München*

Mensch-
Maschine-
Kommuni-
kation 1
(MMK1)

*VL mit
Übungen*

2 + 1 SWS

Dozent: Informations- und Kommunikationssysteme, Dienste, Darstellung von Information; Sinnesorgane und -modalitäten zur Mensch-Maschine-Kommunikation; Analysen von Sprachsignalen, Sprachsynthese, Spracherkennung, Sprechererkennung, Anwendungsfelder; Beschreibung und Analyse von Bildern, Bildverbesserung, Bildrestaurierung, Bildcodierung, Anwendungen; Ergonomie.

Rigoll

Homepage: <http://www.mmk.ei.tum.de/lehre/mmk1/>



*Lehrstuhl für
Mensch-Maschine-
Kommunikation;
TU München*

Mensch-
Maschine-
Kommuni-
kation 2
(MMK2)

Dozent: Funktionskomponenten informationsverarbeitender Systeme, Leistungsfähigkeit, Funktionalität; Interaktionsmodelle; Wissensdarstellung, Lernalgorithmen; Bild- und Sprachinterpretation; natürlichsprachlicher Dialog, Dialog mit Bildern, Multimediadialog; Architekturen für die Bild- und Sprachinterpretation.

*VL mit
Übungen*

2 + 1 SWS

Homepage: <http://www.mmk.ei.tum.de/lehre/mmk2/>

Institute und Arbeitsgruppen



Prof. Dr.-Ing.
Holger Luczak

*RWTH
Aachen*

*Lehrstuhl und
Institut für
Arbeitswis-
senschaft*

*Projekt per-
sonen-
zentrierte
Simulation*

Lehrstuhl und
Institut
für Arbeitswissen-
schaft der RWTH
Aachen
Bergdriesch 27 D-
52062 Aachen
Tel.: (02 41)
80 99 440
Fax: (02 41)
80 92 131
E-Mail:
info@iaw.rwth-
aachen.de

In den Ingenieurwissenschaften ist die computergestützte dynamische Simulation eine innovative Methode zur Planung, Steuerung und Optimierung von technischen Systemen. Unter dem Simulationsbegriff wird in Anlehnung an die VDI-Richtlinie 3633 das Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierfähigen Modell verstanden, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind. Die Mehrheit der existierenden Ansätze zur Simulation von Arbeitsprozessen - zum Beispiel in den Bereichen Fertigung und Montage - beschränkt sich hauptsächlich auf einen Objektfluss und damit auf die Modellierung des Menschen als eine Ressource des technischen Systems. Sollen Arbeitsprozesse im vorgelagerten Teil der Wertschöpfungskette modelliert und simuliert werden, führen die bestehenden Ansätze zu keinem brauchbaren Ergebnis. Im Projekt „Personenzentrierte Simulation“ soll daher ein Simulationsmodell entwickelt werden, welches dem Mitarbeiter eine aktive Rolle zuweist und - beispielsweise - auch Qualifikationen oder persönliche Präferenzen des Mitarbeiters berücksichtigt und damit die oben genannten Unzulänglichkeiten beseitigt.

Homepage: <http://www.iaw.rwth-aachen.de/>

Prof. Dr.-Ing.
Karl-Friedrich
Kraiss

*RWTH
Aachen*

Lehrstuhl für
Technische
Informatik

Lehrstuhl für Technische Informatik
RWTH Aachen
Ahornstraße 55
52074 Aachen
Tel.: +49 241 80-26101
Fax.: +49 241 8888-308

Am Lehrstuhl für Technische Informatik an der RWTH-Aachen wird das Werkzeug TREVIS (Tool for Rapid Evaluation of Interactive Systems) entwickelt, welches Aussagen über die Gebrauchstauglichkeit von interaktiven System trifft. Das Werkzeug kombiniert formale mit experimentellen Evaluierungsmethoden, die sowohl isoliert als auch kombiniert eingesetzt werden können. Dadurch ermöglicht TREVIS eine durchgängige Evaluierung interaktiver Geräte über alle Lebensphasen hinweg. Zur formalen Evaluierung kommt der GOMS-Ansatz (Card, Moran & Newell) zum Einsatz, der eine formale Beschreibung des normativen Benutzerverhaltens in Form von Benutzermodellen ermöglicht. Die Analyse dieser Benutzermodelle liefert neben statistischen Werten Aussagen über die Gebrauchstauglichkeit. So werden z.B. die Ausführungszeit und die Lernzeit zum Erreichen eines Ziels einer Aufgabe abgeschätzt.

Homepage: <http://www.techinfo.rwth-aachen.de/Forschung/MMI/Trevis/index.html>



Dr. Wolfgang
Schoppek

<i>Universität Bayreuth</i>	<i>Lehrstuhl für Psy- chologie Universi- tät Bayreuth</i>	“The main focus of my research is the control of dynamic systems - in the lab as well as in real life. The most important methodology in my research is cognitive modeling. Currently, I'm developing
Lehrstuhl für Psychologie	95440 Bayreuth Tel: +49 921 55-4140 e-mail: wol- fgang.schoppek@ uni-bayreuth.de	ACT-R models of diverse topics such as learning states vs. struc- ture of a system, alphabet arithmetic, and the processes going on in the automated cockpit.” (W. Schoppek)

Homepage: <http://www.uni-bayreuth.de/departments/psychologie/schoppek/>



Dr. Peter Wagner

*Deutsches
Zentrum für
Luft- und
Raumfahrt
(DLR)*

DLR Berlin-
Adlershof
Rutherfordstraße 2
12489 Berlin-
Adlershof

*Institut für
Verkehrs-
forschung*

Vermittlung:
+49 (0)30 67055-0

Abteilung
Verkehrs-
informatik

Die Aufgabe der Verkehrsinformatik im Institut für Verkehrsforschung (IVF) ist es, aus erhobenen Daten Wissen für die Simulation, Steuerung und Planung von Verkehrssystemen zu generieren. Die Herausforderung liegt in der Entwicklung hochwertiger Modelle zur Prognose, Visualisierung und Nachbildung komplexer Vorgänge im Verkehr. In einer groß angelegten Modellierungsoffensive wird zur Zeit weltweit versucht, die Beschreibung und die Simulation von Verkehrssystemen auf einer mikroskopischen Basis durchzuführen. Mittelfristig wird damit ein Instrument geschaffen, das zwei Zielen dient:

(1) es soll die Planung von Verkehrssystemen verbessern helfen und (2) die Grundlagen für ein effektives Verkehrsmanagement legen.

In der Planung wird der dynamische Aspekt benötigt, weil die Aktivitätenplanung von Personen modelliert wird, die natürlich tageszeitabhängig ist. Der dynamische Aspekt im Verkehrsmanagement ist die Unterstützung bei der Rekonstruktion des aktuellen Zustandes. Langfristiges Ziel ist die Kopplung solcher Modelle mit ökonomischen Modellen, um zu einer integrierten Sicht für das kombinierte System Wirtschaft und Verkehr zu gelangen. Dieses Vorhaben kann nur in Forschungsnetzwerken mit externen Partnern realisiert werden. Gleichzeitig soll die – im internationalen Vergleich – starke Position Deutschlands auf dem Gebiet der Entwicklung von Verkehrsmodellen ausgebaut werden.

Angestrebt wird, den Trägern öffentlicher Belange, Forschungseinrichtungen und privaten Dienstleistern den Zugang zu Verkehrsdaten zu erleichtern sowie qualitativ hochwertige Informationen bereitzustellen. Mit der Einrichtung einer Clearingstelle für Verkehrsdaten und Verkehrsmodelle sollen Daten- und Modellgrundlagen für die Verkehrsforschung und -planung verbessert und aufeinander abgestimmt werden.

Homepage: <http://ivf.dlr.de/vf/institut/abteilungen/verkehrsinformatik>



Institut
Rechnerarchitektur
und Softwaretechnik

Dr. Steffen Unger

*Fraunhofer
Gesellschaft*

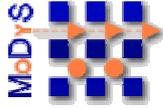
*Institut
Rechner-
architektur
und Software
technik –
FIRST*

Kekuléstraße 7 Design und Implementierung generischer Simulationssysteme für
12489 Berlin Umwelt, Verkehr und nachhaltige Entwicklung; Kopplung von
Telefon: Modellen für GRID Computing; Entwicklung von Komponenten
+49 (0) 30 / zur Integration von Nutzeroberflächen (3-D-Grafik, GIS, WWW-
63 92-18 16 Interfaces, etc.) und Datenbanken; Unterstützung von Nutzern bei
E-Mail: stef- operativem Management, strategischer Planung und Training; Er-
fen.unger@first.fr stellung verifizierter Vorhersagen für Luftqualität und Verkehrspla-
aunhofer.de ge in Ballungszentren.

Arbeitsgebiet
SAS - Sys-
temanalyse
und Simula-
tion

Homepage:

http://www.fraunhofer.de/german/profile/institute/first/first_f_contact_01.html#4



Dr.-Ing.
Leon Urbas

TU Berlin

*Zentrum
Mensch-
Maschine
Systeme*

MoDyS
Research
Group*

**Methods of
User Modelling
in dynamic
Human-
Machine Sys-
tems*

ZMMS Technische Uni- versität Berlin J 2-2 Jebensstr.1 D-10623 Berlin Germany

The MoDyS research group evaluates and develops methods for modeling operator/user-behavior in dynamic human-machine-systems. Especially quantitative models which describe interaction processes are matter of particular scientific interest. Areas of application include operator support systems, training systems and educational software as well as tools for the human centered design of interaction in dynamic human-machine-systems.

To accomplish this task MoDyS brings together ideas, methods and expertise from different research areas:

- fon: ++49 30 31472593
fax: ++49 30 31472581
- cognitive psychology
 - artificial intelligence
 - computational intelligence
 - human-machine-system studies
 - software engineering
 - control theory
 - first-principles based dynamic modeling of technical systems
 - control and information systems engineering

Homepage: <http://www.zmms.tu-berlin.de/modys/>



Prof. Dr.-Ing.
Uwe Völckers

*Deutsches
Zentrum für
Luft- und
Raumfahrt
(DLR)*

Institut für
Flugführung

Deutsches Zentrum für Luft- und Raumfahrt e.V. in der Helmholtz-Gemeinschaft

Institut für Flugführung

Lilienthalplatz 7
38108 Braunschweig
Deutschland

Tel.: +49-531-295
2501

Fax: +49-531-295
2550

E-Mail: Flugfuehrung@dlr.de

Das Institut für Flugführung betreibt langfristig angelegte ingenieurwissenschaftliche Forschung im Vorfeld industrieller Entwicklungen. Es ist auf dem Gebiet der Flugführung und des Luftverkehrsmanagements mit Anbindung an andere Verkehrsträger schwerpunktmäßig in den Bereichen operationelle Verfahren, Technologieentwicklung und mensch-zentrierte Automatisierung tätig.

Das Institut entwickelt und bewertet Konzepte - Lösungen - Verfahren für das Luftverkehrsmanagement, den Flughafen und den Flughafennahbereich. Ziel ist die Verbesserung der Effizienz, Nutzerfreundlichkeit und Sicherheit.

Homepage: <http://www.dlr.de/fl/>



Prof. Dr. Nicola
Henze (*Sprecherin
der Fachgruppen-
leitung*)

*Gesellschaft
für Informa-
tik*

Mail an die Fach-
gruppenleitung:
lg-abis@kbs.uni-
hannover.de

Fachgruppe

*FB KI –
Künstliche
Intelligenz*

und

*FB MCI –
Mensch-
Computer-
Interaktion*

*Adaptivität
und Be-
nutzer-
modellierung
in interakti-
ven Soft-
ware-
systemen
(ABIS)*

Die individuelle Anpassung von interaktiven Softwaresystemen an den jeweiligen Benutzer ist in den letzten Jahren auf großes Interesse sowohl der grundlagen- als auch der anwendungs-orientierten Forschung gestoßen. Industrieunternehmen erkennen zunehmend die Wichtigkeit dieses Themas. In großen Softwarehäusern werden weltweit schon seit geraumer Zeit Systemkomponenten entwickelt, die Benutzern individuelle und aktive Unterstützung anbieten - oft unter der Metapher eines Assistenten. Die Bemühungen gehen dabei immer in eine globale Richtung: das (Anwendungs-) System auf der Grundlage von modelliertem und erworbenem Wissen über die Benutzer und ihre Nutzung des Systems adaptiv zu gestalten. [...] Erfolgreiche Lösungsansätze solcher Systeme werden zunehmend disziplinübergreifend entwickelt: sie nutzen Erfahrungen, Methoden und Entwicklungen aus den Fachgebieten „Software-Ergonomie“, „Mensch-Computer Kommunikation“ und „Künstliche Intelligenz“ aus. Diese generelle Entwicklung interdisziplinärer Zusammenarbeit aufgreifend, soll mit der Fachgruppe Adaptivität und Benutzermodellierung ein Arbeits- und Diskussionsforum geschaffen werden, das es den Interessenten aus den beteiligten Fachdisziplinen erlaubt, ihre Zusammenarbeit und den Erfahrungsaustausch zu intensivieren. Vertreterinnen und Vertreter aus der Praxis und der Forschung sollen zusammenkommen, um ihre aktuellen Ergebnisse und anstehenden Probleme in einem kompetenten Forum zur Diskussion zu stellen. Parallele Bestrebungen in anderen Staaten haben bereits zur Bildung von Fachgruppen auf nationaler und internationaler Ebene geführt. Die Fachgruppe hat den Anspruch, das Bewusstsein für die Bedeutung interdisziplinärer Arbeit auf diesem Gebiet zu fördern, um damit auch zu verdeutlichen, dass der Benutzer mit seinen Problemen beim Umgang mit Werkzeugen oder bei der Lösung von Aufgaben im Mittelpunkt des Interesses und der Entwicklungen stehen muss.

Homepage: <http://www.kbs.uni-hannover.de/~henze/abis/fg2.3.3.html>



Prof. Dr. Ing.
Reiner Onken

Universität
der Bundes-
wehr Mün-
chen

Institut für
System-
dynamik und
Flug-
mechanik

Geschäftszimmer: Wissenschaftliche Arbeitsschwerpunkte:

Madeleine Gabler

089/60043583

Gebäude 41/200

Raum 2212

*Madelei-
ne.Gabler@UniBw-
Muenchen.DE*

- Wissensbasierte, kognitive Systeme zur Unterstützung bei Aus-
bildung und Training

- Systemseitig autonome, zielbezogene Situationserkennung und -
analyse bezüglich des Verhaltens und der Ressourcen des Fahrzeug-
führers, des Fahrzeuges und der Umgebung

- Wissensakquisition

Verhaltensmodellierung von Fahrzeugführern (Motive/Ziele, Ab-
sichten/Aufgaben, Handlungen, Fehlverhalten)

Modellierung von Verhalten bezüglich Fertigkeiten, regelbasiertem
und wissensbasiertem Verhalten

- Aufbau statischer Datenbanken zur Modellierung von Fahrzeug-
führern

Systemseitig autonome Erzeugung von Vorschlägen zur Lösung
von aufkommenden Konflikten und Nutzung von sich anbietenden,
zielunterstützenden Gelegenheiten

- Dialogmanagement: Systeminitiativen zum Abgleich des Situati-
onsverständnisses und der Nivellierung der Beanspruchung des
Fahrzeugführers

- Absicherung kognitiver Systeme gegen Software-Fehler

Projekte:

- Cockpit Assistenz System (CASSY)
- Crew Assistant Military Aircraft (CAMA)
- Automatic cruise control for trucks and cars

Cognitive tutoring system (COTUTSY)

Homepage: <http://www.unibw-muenchen.de/campus/LRT/LRT13/deutsch/index.html>



Prof. Dr.
C. Möbus

*Universität
Oldenburg*

*Department
für Informa-
tik*

*Abteilung für
Lehr- und
Lernsysteme*

Sekretariat: Auf der Grundlage einer psychologischen Konzeption zu Problem- lösen und Wissenserwerb sowie empirischer Datenerhebungen und -analysen werden kognitive Modelle des Erwerbs und der Optimierung von Wissen entwickelt. Hiermit werden zum einen grundla- genorientierte kognitionswissenschaftliche Fragestellungen, z.B. bezüglich der Akzeptanz von Informationen und Hilfen, untersucht und zum anderen das anwendungsorientierte Design von Hilfesys- temen unterstützt. Empirische Feldstudien in Flugsimulatoren die- nen als Basis zur ingenieurmäßigen Modellierung kognitiver Pro- zesse, die zu Fehlbedienungen bei der Mensch-Computer Interakti- on insbesondere im Umgang mit Automatisierungssystemen im Flugzeugcockpit führen können. Im Rahmen einer Simulations- plattform werden derartige Modelle zur Analyse menschlichen Fehlverhaltens im Entwicklungsprozess hochkomplexer eingebette- ter Systeme in Fahrzeugen eingesetzt. Eine nahtlose Anbindung dieser Methode an den industriellen Entwicklungsprozess wird durch die Integration bestehender formaler Designmodelle in die Simulation garantiert.

Homepage: <http://ils.informatik.uni-oldenburg.de/index.php>



Wayne Gray
(Chair)

Human Factors and Ergonomics Society

Human Performance Modeling Technical Group

[in
Gründung; proposed to the HFES Council of Technical Groups (CoTG)]

Department of Cognitive Science
Rensselaer Polytechnic Institute
Carnegie 108
110 8th Street
Troy, NY 12180
Email: grayw@rpi.edu
Phone: 518-276-3315

The Human Performance Modeling TG will be concerned with the development and application of predictive and reliable, quantitative models of human performance. In distinction to other approaches to behavioral and cognitive modeling, human performance modeling considers the human, engaged in some goal-directed behavior, in the context of a designed task environment. The scope of the models of interest to the TG encompasses the scope of the systems of interest to the Human Factors and Ergonomics Society. Hence, we would equally promote models of isolated aspects of human performance, models of the cognitive control of memory, attention, perception, and action, as well as models that are integrative in the sense that they receive task-related information from the environment and produce thoughtful human-like action. The TG will have an interest in promoting and disseminating (a) the basic science foundation of such models, (b) engineering research needed to apply human performance models to human factors applications, (c) new formalisms for human performance modeling, and (d) techniques for evaluating the predictive success of such models. We see the TG as a forum for testing modeling approaches that are emerging from the basic research community against the hard realities of human factors problems. Contrariwise, we see the identification of challenges faced by the human factors community in human performance modeling as providing significant feedback to more basic researchers on the problems to be overcome and the opportunities for improvement to the research base.

Homepage: <http://www.cogsci.rpi.edu/cogworks/hpm-tg/>



Prof. Dr. Felix
Breitenecker
(Sprecher des Lei-
tungsgremiums)

*Gesellschaft
für Informa-
tik*

*Fachbereich
4: Informa-
tionstechnik
und Techni-
sche Nutzung
der Informa-
tik*

Arbeits-
gemeinschaft
Simulation
(ASIM)
(Fachaus-
schuss 4.5
Simulation)

Email: ASIM - Arbeitsgemeinschaft Simulation - ist eine Arbeitsgemein-
schafft im deutschsprachigen Raum zur Förderung und Weiterent-
wicklung von Modellbildung und Simulation in Grundlagen und
Anwendung sowie zur Verbesserung der Kommunikation zwischen
Theorie und Praxis. ASIM ist in Fachgruppen strukturiert, die sich
mit spezifischen Gebieten der Simulationstechnik beschäftigen.
ASIM ist auch Gründungsmitglied von EUROSIM, des Dachver-
bands der europäischen Simulationsgesellschaften, und ist in des-
sen Executive Board vertreten. Enge Zusammenarbeit besteht da-
durch mit anderen und internationalen Simulationsvereinigungen
(SCS, IMACS). ASIM arbeitet in der Administration und als
Buchherausgeber mit der ARGESIM (ARGE Simulation News,
Wien) zusammen.

Homepage: <http://www.asim-gi.org/>