

Useware Engineering mit kognitiven Architekturen

Useware Engineering With Cognitive Architectures

S. Leuchter & L. Urbas, Berlin

Kurzfassung

agimap erlaubt die ingenieurmäßige Erschließung der wahrnehmungsnahen Bereiche der kognitiven Simulation beim Useware Engineering mit kognitiven Architekturen. In einem grafischen Editor wird das XML-Modell der zu bedienenden Softwareoberfläche (oder der computervermittelten Schnittstelle des Mensch-Maschine-Systems) erstellt. Das Modell wird in ACT-R/PM Code mit XSLT transformiert, der von einem um agimap erweiterten ACT-R-Interpreter eingebunden wird. Durch dieses Entwicklungswerkzeug zur kognitiven Simulation ist ein effizienter Usabilityvergleich von Prototypen in frühen Phasen der Systementwicklung möglich.

1. Einleitung

Rechnergestützte Modellierung und Simulation sind nicht mehr wegzudenkende Werkzeuge bei der Gestaltung und Bewertung technischer Systeme. Um diese Methodik auf die Betrachtung des gesamten Mensch-Maschine-Systems auszuweiten, müssen Aufgabe und Leistung der Menschen, die das technische System zur Zielerreichung verantwortlich steuern und führen, die Bediener, ebenfalls im Rechner abgebildet werden.

Anstrengungen zur Bedienermodellierung mit dem Ziel Leistungspotential abzuschätzen und Bedienfehlermöglichkeiten in Mensch-Maschine-Systemen zu identifizieren, sollten auf kognitive Strukturen ausgerichtet sein. Insbesondere die Modellierung von Wahrnehmung, exekutiver Kontrolle und Beschränkungen des Arbeitsgedächtnisses sind dafür nützlich. Aufgrund der im Vergleich zu anderen Modellierungsformalismen wie Künstlichen Neuronalen Netzen besseren Kommunizierbarkeit der Modelle und der einfacheren Behandlung von Wissen empfiehlt sich dafür das Rahmenwerk der regelbasierten Systeme, speziell Produktionssysteme, in dem die Kognitionswissenschaft unterschiedliche Theorien zu Elementen der

Kognition und deren Zusammenspiel gemacht hat. Einige dieser Theorien bringen aus einer Zusammenarbeit von Künstlicher Intelligenzforschung und kognitiver Psychologie Implementierungsumgebungen für die Formalisierung und Simulation kognitiver Modelle mit. Diese kognitiven Architekturen versprechen im Rahmen ihres Geltungsbereiches (der vom Anspruch her universal ist) geprüfte Mechanismen und eine Community, die ihre Arbeit im Rahmen der zugehörigen Theorien durchführt und die Architektur damit weiterentwickelt.

Zur Auswertung der Literatur über architekturgebundene kognitive Modellierung wurde ein Klassifizierungsrahmen für Anforderungen aus Dynamik und Komplexität von Aufgaben in dynamischen Mensch-Maschine-Systemen entwickelt [1]. Arbeiten zu existierenden kognitiven Architekturen wurden daran gemessen und ACT-R/PM als besonders vielversprechender Kandidat ausgewählt. Es zeigt sich aber, dass auch diese Architektur für die ingenieurmäßige Modellierung von Bedienerleistungen in dynamischen Mensch-Maschine-Systemen weiterentwickelt werden muss [1]. Zur Realisierung von Bedienermodellen für dynamische Mensch-Maschine-Systeme wurde neben anderen die ACT-R/PM-Architektur-erweiterung „agimap“ entwickelt.

2. Useware Engineering mit kognitiven Modellen

Die Simulation kognitiver Modelle ist eine Usabilitymethode, die die Bewertung von Prototypen bereits in frühen Phasen der Entwicklung unterstützt. Gegebenfalls ist sogar eine Simulation mit Modellen möglich, die aus vorliegenden technischen Spezifikationen automatisch abgeleitet werden [2]. Die Simulation ist bereits länger eine Methode des Usabilitymethodenkanons. Insbesondere die Familie der GOMS Methoden (KLM, NGOMSL, CPM-GOMS) wird benutzt, um die Bedieneffizienz von Prototypen zu vergleichen. Hier werden Bediensequenzen jedoch lediglich anhand von Ausführungszeiten und Fehlerwahrscheinlichkeiten atomarer Operationen simuliert [3]. Der Einsatz von kognitiven Simulationen ermöglicht folgende Erweiterungen: Die online-Simulation erlaubt auch die Benutzung von sehr komplexen Systemen zu untersuchen, indem Regeln modelliert werden, durch deren Anwendung die Bedienung auch adaptiver Systeme simuliert wird. Zusätzlich können Eigenschaften von Gedächtnisstrukturen, die für die Erledigung von Aufgaben benötigt werden, zur Simulation herangezogen werden. Dadurch wird es möglich, Usabilityvergleiche auf weitere Fehlerklassen und Effizienzmaße auszuweiten: z.B. Verwechseln, Vergessen, schnellerer Abruf bei Wiederholungen, langsamerer Abruf bei Unterscheidungsentscheidungen.

Der Einsatz von kognitiven Architekturen als Basis für die kognitiven Simulationen verspricht einige Vorteile gegenüber einer „ad hoc“-Modellierung in einer allgemeinen Programmiersprache:

- Fundierung auf kognitionswissenschaftlich begründete Strukturen und Mechanismen, die validierte Konstrukte beinhalten, wodurch eine höhere Prädiktionskraft erwartet werden kann, also ein geringerer Aufwand für die Erhebung von Parametern besteht.
- Hinter kognitiven Architekturen steht jeweils eine Community, die neue Erkenntnisse auf der Basis der Architektur gewinnt und Ergebnisse im Rahmen der Architektur zur Verfügung stellt.
- Die kognitiven Architekturen, die für dieses Programm von Interesse sind, sind in einer Simulationssoftware implementiert.

Insbesondere im Bereich der Gedächtnisleistungen und der Wahrnehmung gibt es in aktuellen Architekturen viele funktionierende Bausteine.

Eine verbreitete kognitive Architektur ist ACT-R/PM [4]. Sie basiert auf der *physical symbol theory* [5] und implementiert sie auf der Basis von Produktionen und einer Repräsentation in einem semantischen Netz. Parallel zu dieser symbolischen Interpretation gibt es eine sub-symbolische Ebene, auf der die repräsentierten Fakten über eine variable Aktivierung und die Produktionen über variable Stärken verfügen, die zusammen in das Konfliktresolutionskalkül zur Abarbeitung des Regelsystems einfließen. Die aus der Kognitionspsychologie stammende Architektur wird aktuell u.a. in Richtung einer Anwendung im HCI- und MMS-Bereich von einer internationalen, interdisziplinären und aktiven Community genutzt und weiterentwickelt. Anwendungsdomänen betreffen z.B. die Kraftfahrzeugführung [6], die Flugführung [7], die Flugsicherung [8] und die Anlagensteuerung [9].

Um die Interaktion des regelverarbeitenden Systems (als Simulation eines Benutzers) mit einer technischen Umwelt zu modellieren, existiert das Subsystem PM in ACT-R [10]. Hier sind Strategien und Eigenschaften der Wahrnehmung und der motorischen Handlung auf feingranularer Ebene vorhanden, die in kognitiven Modellen genutzt werden können. Eine Schnittstelle zwischen dem PM-Subsystem und der technischen Umwelt ist das AGI (ACT-R GUI Interface [11]). Es erlaubt einem ACT-R-Modell die Elemente einer grafischen Benutzungsoberfläche (GUI) zu „sehen“ und GUI-Widgets zu „bedienen“. Dadurch wird eine Bedienbarkeitsbewertung auf GUI-Level möglich. AGI ist jedoch eine Abstraktion innerhalb desselben Prozesses wie die kognitive Simulation, also so wie es angeboten wird, keine Schnittstelle zu externen Prozessen.

3. agimap

Eine Kopplung mit beliebigen externen Prozessen ist jedoch für den Einsatz in Mensch-Maschine-Systemen, für die entweder aufwändige Simulatoren oder sogar softwaretechnische Schnittstellen von Maschinen oder Prozessen existieren – also keine realistische Möglichkeit gegeben ist, die technische Umwelt innerhalb der kognitiven Simulation nachzubilden, absolut erforderlich.

„agimap“ ist ein Beitrag, um die AGI-Schnittstelle von ACT-R/PM effizient und effektiv zur Anbindung von externen Prozessen zu nutzen. Dazu bietet agimap ein Framework an, mit dem externe Prozesse über eine Socketverbindung (TCP/IP) an ACT-R/PM-Modelle angeschlossen werden können. Von der entfernten Anwendung wird erwartet, dass sie im verbreiteten Observer Design-Pattern [12] Änderungen von allen Prozessvariablen überträgt. In einem einfachen Protokoll werden dann Wertänderungen und Eingriffe übertragen.

Eine Wertänderung muss sich im AGI widerspiegeln. Dazu kann für jede Prozessvariable ein agimap-Mapper instanziiert werden, dessen Aufgabe es ist, bei einer Änderung „seines“ Wertes diesen Wert zu transformieren und auf eine sinnvolle Art im AGI darzustellen, wodurch der Wert dem ACT-R-Modell zur Verfügung gestellt wird.

Die Ebene, auf der ACT-R/PM mit AGI interagiert, ist sehr detailliert. Um einen Effekt zu erreichen, wie z.B. einen Button zu klicken oder um eine bestimmte Information an einem simulierten Widget im AGI zu finden, aufzunehmen und ACT-R zur Verfügung zu stellen, sind viele interagierende Produktionen erforderlich. Diese niedrige Granularitätsebene erlaubt dann in Simulationen weitergehende Aussagen über die Bedieneffizienz einer Schnittstelle zu machen, als wenn – wie dies häufig vereinfachend implementiert wurde – ACT-R Werte über Prozessvariablen direkt, also ohne Umweg über das „Ablesen“ von Displayelementen, aus einer technischen Umgebung übermittelt werden würden.

agimap ist eine Erweiterung, die ACT-R um eine höher aggregierte Schnittstelle zu AGI erweitert. Die Erweiterung geschieht in einer Weise, so dass die Aktionen auf feingranularer Ebene nicht ersetzt, sondern hinter einer Schnittstelle gekapselt werden. Dazu stellt agimap einige vorgefertigte Widget-Typen mit Mappern zur Verfügung, die im AGI verwendet werden können. Zu jedem vordefinierten Widget gibt es eine Reihe von passenden Strategien z.B. zur Informationsaufnahme und zu manipulativen Aktionen. Für jede Instanz eines solchen Widgets in AGI werden Produktionen und deklarative Strukturen automatisch erzeugt. Zur Benutzung der Strategien gibt es ein API, das über Zielstrukturen mit ACT-R-Produktionen verbunden ist.

AGI ist für die Bedienung von Bürosoftware-GUIs ausgelegt. Entsprechend beziehen sich die Möglichkeiten zur Informationsaufnahme auf digitale Werte wie Texte, Farben und Ziffern. Anzeigeeinheiten von Mensch-Maschine-Systemen arbeiten aber zur Erleichterung der Interpretation durch die Bediener auch mit analogen Anzeigen (z.B. Tachometern oder Füllstandsanzeigen) . Die von agimap zur Verfügung gestellten analogen Widgets basieren darauf, dass die aus der technischen Simulation kommenden Werte von einem Mapper auf eine passende Repräsentation in AGI transformiert werden. Hier wird mit den textuellen Mitteln, die AGI zur Verfügung stellt (Texte an beliebigen Positionen in unterschiedlichen Farben) eine entsprechende Darstellung erreicht. Dazu wird ein Mapper instanziiert und entsprechend konfiguriert. Beispiele für solche Mapper sind MapText2Color (je nach übertragenem Wert Farbcodierung) und MapText2Pos (je nach übertragenem Wert Positionsänderung des AGI-Widgets). Durch deren Kombination kann z.B. eine analoge grafische Füllstandsanzeige mit farbigen Minimum- und Maximumalarmen durch die Transformation des Füllstandswertes in eine Position (der Zeichenkette ---) und die Transformation der Alarmmeldungen in farbige Zeichensymbole (>>) in eine textuelle umgesetzt werden (s. Bild 1). Die Strategien, mit denen die Informationsaufnahme an der AGI-Repräsentation des GUI-Elementes aus der originalen Benutzungsoberfläche erfolgen, sind so gestaltet, dass der analoge Charakter benutzt wird. So kann beispielsweise ein Fülltrend durch Abfrage und Vergleich der Positionen ermittelt werden.

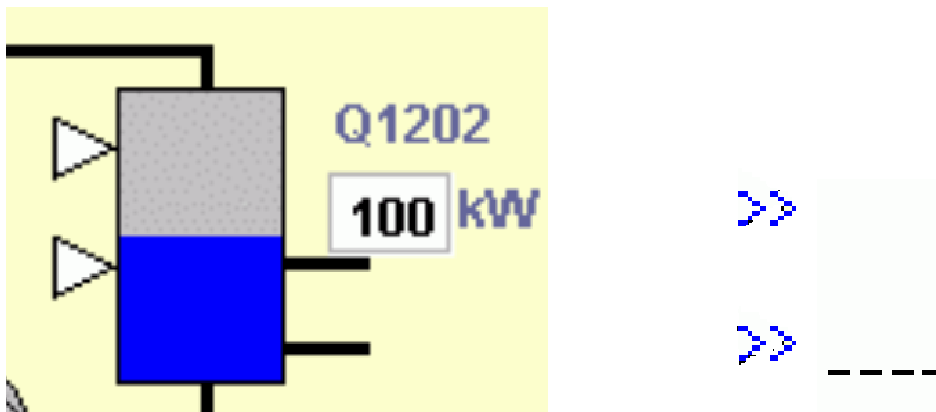


Bild 1: Mapping von grafisch analoger Anzeige im Original-GUI (links) zur textuell analoger Anzeige in AGI über einen Mapper (rechts)

Die Repräsentation des Interface-Modells mit den verwendeten Mapper-Instanzen, ihrer Konfiguration und Daten über die Kopplung der Simulationsprozesse erfolgt in einem grafischen Editor (s. Bild 2). Der Editor benutzt XML zur Speicherung dieser Daten. Dazu wurde eine DTD erstellt, in der die Struktur des XML-Formates definiert ist. Die XML-

Repräsentation kann dann mit einem XSLT-Skript unter Verwendung der DTD nach CommonLISP transformiert werden (modell.agm LISP in Bild 3). Diese automatisch generierte Definition der Kopplung und der verwendeten GUIs wird im ACT-R-Modell der kognitiven Simulation eingebunden (modell ACT-R). Der ACT-R-Interpreter beinhaltet die agimap-Erweiterung (agimap LISP), mit der das erweiterte Modell simuliert werden kann. Das Ergebnis der kognitiven Simulation ist ein Protokoll (Protokoll LOG), das auf Benutzungsprobleme hin analysiert werden kann.

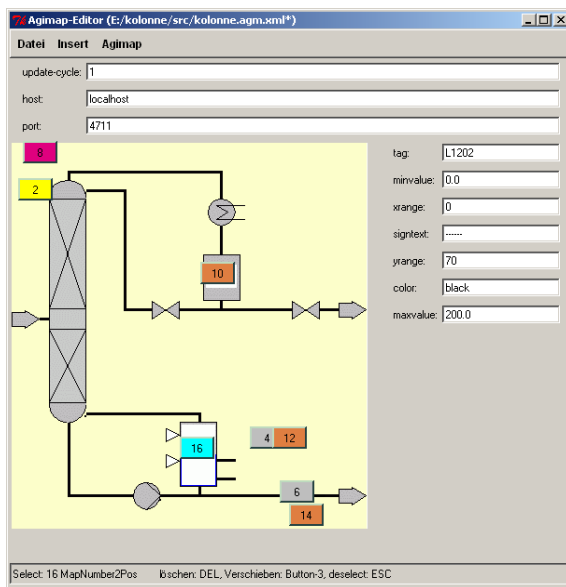


Bild 2: Grafischer Editor zum Erstellen des GUI-Modells und zur Konfiguration der Prozesskopplung.

Die Werkzeugkette von agimap ermöglicht es bei der Modellbildung sowohl von der Implementierung der wahrnehmungsnahen Modellierungsaspekte als auch von der technischen Anbindung an die simulierte Umwelt zu abstrahieren, indem der grafische Editor benutzt wird, dessen Daten direkt in die Simulation eingebunden werden können.

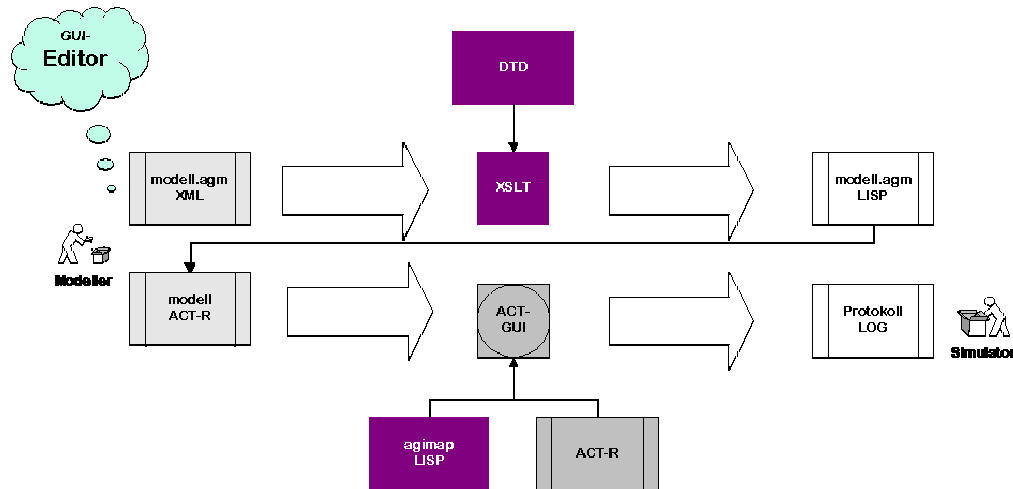


Bild 3: Werkzeugkette von agimap.

Durch die Verwendung von AGI als Basis und die Kopplung über Sockets bietet agimap eine weitgehende Plattformunabhängigkeit. Die Simulation kann auf beliebigen Rechnerplattformen laufen. AGI ist sowohl unter MacOS, Linux als auch unter Windows lauffähig. Der Editor ist in Tcl/Tk, einer ebenfalls auf den meisten GUI-Plattformen verfügbaren Programmiersprache, portabel implementiert. Das Handling der Kopplung bei der Simulation wird durch agimap weitgehend automatisiert.

Durch die Verwendung von AGI wird die Simulation jedoch auf nur ein GUI-Fenster beschränkt, in dem ggf. Unterfenster und der Wechsel zwischen ihnen simuliert werden müsste. Die aktuelle Implementierung von agimap lässt außerdem nur einen Mapper pro Prozessvariable zu.

4. Ausblick

agimap erlaubt die ingenieurmäßige Erschließung der wahrnehmungsnahen Bereiche der kognitiven Simulation mit kognitiven Architekturen. In dem grafischen Editor wird das XML-Modell der zu bedienenden Softwareoberfläche (oder der computervermittelten Schnittstelle des Mensch-Maschine-Systems) erstellt. Das Modell wird in ACT-R/PM Code mit XSLT transformiert, der von einem um agimap erweiterten ACT-R-Interpreter eingebunden wird. Die agimap-Erweiterung erlaubt eine Socket-Kopplung einer externen Simulation der technischen Umwelt, die zu bedienen ist, mit der AGI-Schnittstelle in der kognitiven Simulation. Dadurch kann das modellierte GUI effizient mit ACT-R-Produktionen benutzt werden. Durch dieses Entwicklungswerkzeug zur kognitiven Simulation ist ein effizienter Usabilityvergleich von Prototypen in frühen Phasen der Systementwicklung möglich.

Durch die grafische (Teil-) Programmierung des PM-Anteils des ACT-R/PM-Modells wird die Modellerstellung wesentlich erleichtert. Bislang ist dies das einzige grafische Software-Engineering-Werkzeug im ACT-R-Umfeld und die vordefinierten Interface-Widgets mit den ihnen zugeordneten Benutzungsstrategien der erste Ansatz für Teilmodell-Wiederverwendung in ACT-R. Die automatische Codegenerierung ist nicht nur effizienter als eine Handcodierung; durch die High-Level-Schnittstellen zu ACT-R/PM und die folgende Komplexitätsreduktion ist sie auch weniger fehleranfällig.

Im Moment existieren agimap-Mapper für Interface-Anwendungen in der chemischen Prozesskontrolle. Das Framework ist aber offen gehalten und es können ohne Probleme neue Interfacewidgets mit agimap-Mappern und zugeordneten Strategien definiert werden. So ist ein Einsatz in der Fahrzeugführung einfach möglich, indem z.B. ein Tachometer-Widget bereitgestellt wird. Aufgrund des AGI-Konzeptes ist man jedoch auf Anwendungsfelder mit computerabbildbaren Mensch-Maschine-Systemschnittstellen beschränkt.

Gefördert von der VolkswagenStiftung im Rahmen des Programms „Nachwuchsgruppen an Universitäten“.

5. Literatur

- [1] 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.
- [2] Hamacher, N., Kraiss, K., Marrenbach, J. (2002). Einsatz formaler Methoden zur Evaluierung der Gebrauchsfähigkeit interaktiver Geräte. *it + ti Informationstechnik und Technische Informatik*, 44 49-55.
- [3] Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum.
- [4] Anderson, J. R. & Lebiere, C. (1998). *Atomic Components of Thought*. Hillsdale, N.J.: Erlbaum.
- [5] Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.
- [6] Salvucci, D. D., & Macuga, K. L. (2002). Predicting the effects of cellular-phone dialing on driver performance. *Cognitive Systems Research*, 3, 95-102.

- [7] Gluck, K. A., Ball, J. T., Krusmark, M. A., Rodgers, S. M., Purtee, M. D. (2003). A Computational Process Model of Basic Aircraft Maneuvering. In F. Detje, D. Dörner, & H. Schaub (Eds.), *The Logic of Cognitive Systems. Proceedings of the Fifth International Conference on Cognitive Modeling* (pp. 117-122). Bamberg: Universitäts-Verlag.
- [8] Taatgen, N. A. (2002). A model of individual differences in skill acquisition in the Kanfer-Ackerman Air Traffic Control Task. *Cognitive Systems Research*, 3(1) 103-112.
- [9] Wallach, D. (1996). *Komplexe Regelungsprozesse: Eine kognitionswissenschaftliche Analyse*. Wiesbaden: DUV.
- [10] Byrne, M. D., & Anderson, J. R. (1998). Perception and Action. In John R. Anderson, & Christian Lebiere (Eds.), *The Atomic Components of Thought* (pp. 167-200). Mahwah, NJ: Lawrence Erlbaum.
- [11] anonym (o.J.). *AGI Manual*. Online <http://act-r.psy.cmu.edu/tutorials/AGI.pdf> (letzter Zugriff 15.04.2004).
- [12] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison Wesley.