University of Groningen
Artificial Intelligence
prepublications

author: J. Misker, N.A. Taatgen and J. Aasman
title: Validating a tool for simulating user interaction
status: Revised, Accepted for ICCM-2001

March 2001
prepublication 2001-7

RuG

Validating a Tool for Simulating User Interaction

Jan Misker (jan@misker.nl)
Artificial Intelligence, University of Groningen,
Grote Kruisstraat 2/1, 9712 TS Groningen, The Netherlands
KPN Research,
Sint Paulusstraat 4, 2264 XZ Leidschendam, The Netherlands

Niels A. Taatgen (niels@ai.rug.nl)
Artificial Intelligence, University of Groningen
Grote Kruisstraat 2/1, 9712 TS Groningen, The Netherlands

Jans Aasman (J.Aasman@kpn.com)
KPN Research,
Sint Paulusstraat 4, 2264 XZ Leidschendam, The Netherlands
Industrial Design Engineering, University of Delft,
Jaffalaan 9, 2628 BX, Delft

Abstract

In this paper a tool will be presented that simulates human perception and motor behavior in interaction with graphical user interfaces in the Microsoft Windows environment. The simulated hand and eye tool can be used in combination with any cognitive architecture. In order to validate the simulated hand and eye an experiment has been conducted in which human subjects showed simple, low-level interaction behavior. Eye movements and finger presses were measured and recorded. This allowed for basic validation of the simulated hand and eye. Furthermore, based on the collected data, an adjustment of the EMMA theory is suggested. To demonstrate the full usage if the tool, ACT-R has been used to model basic skill acquisition in instructed interaction behavior.

Introduction

Much research has been done to describe human interaction with computer applications (HCI), as well as interaction with devices. Only recently this research has produced theories with a level of quantitative detail that allows for software implementations of these theories. These advances are also of interest to KPN Research[1], because one of the possible future results could be a tool that automatically tests user interfaces.

Requirements In order to research these possibilities, I did my M.Sc. research project at KPN Research. I was asked to design and implement a software tool that was required to (1) interact with any graphical user interface in the Microsoft Windows environment and (2) be usable in conjunction with common cognitive architectures. Furthermore; the tool should (3) produce realistic timing and accuracy information regarding the interaction.

For KPN Research, the main focus is on user interfaces of devices and appliances, for example, (mobile) telephones. Thus the user interfaces were regarded as interfaces of devices simulated or prototyped as MS Windows applications, not of computer applications. This means that humans would interact with them using their hands and fingers instead of a mouse, which should be reflected in the behavior of the tool.

Because the tool will simulate the hand and eye of humans, it is called the Simulated Hand and Eye, which is shortened to SHE.

Testing the tool To test the usefulness of this approach and give a proof-of-concept, another goal of the project was to build cognitive models in ACT-R that in combination with SHE interact with user interfaces based on explicit instructions.

Existing work In the literature several theories on human interaction with user interfaces can be identified, as well as some implementations of these theories. Of these, work done by Ritter et al. (2000) and Byrne (Byrne, in press) most closely resembles SHE.

Ritter already developed a simhand and simeye, although this tool currently does not interact with MS Windows. The general architecture of SHE has been adopted from his work (Ritter, Baxter, Jones & Young, 2000).

ACT-R/PM (Byrne, 2000; Byrne & Anderson, 1998) is also able to interact with user interfaces, however currently only to interfaces written in two Lisp variants[2]. The latter aims at simulating human computer

[1] KPN Research is the research department of the largest Dutch telecom company.

[2] It should be noted, that ACT-R/PM is designed to be usable with other interfaces as well, but because ACT-R/PM

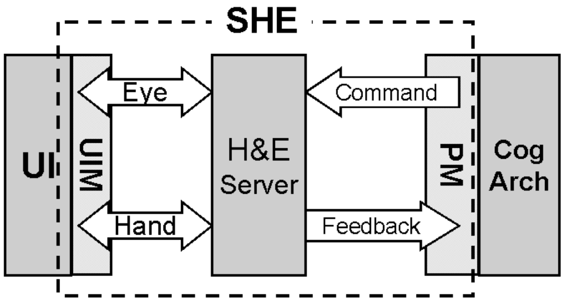

Figure 1: Overview of SHE.

interaction using a mouse, instead of human machine interaction using hands and fingers. Furthermore ACT-R/PM can only be used in conjunction with ACT-R, so users of Soar cannot benefit from this work.

Both of the efforts described above were very valuable in the design of SHE, but the requirements of the project demanded a more generally applicable tool.

Other valuable sources of information for this project were the Model Human Processor (Card, Moran & Newell, 1983), the EPIC architecture (Kieras & Meyer, 1997), the EMMA theory (Salvucci, 2000), the Driver model (Aasman, 1995), work done by Gray et al. (Gray, 2000; Gray & Altmann, in press) and the VisMap project (St. Amant & O. Riedl, in press).

The design of SHE

SHE will be a tool in between the user interface (UI) and the cognitive architecture. Figure 1 gives an overview of the overall design of the tool. This figure is an adapted version from the figure describing the Cognitive Modeling Interface Management System in Ritter et al. (2000). In the following paragraphs I will describe the technical issues and solutions in the design of SHE.

Knowledge Representation Level In the design of SHE, the level of knowledge representation is chosen at a high level. Visual information from the user interface is translated into objects like buttons, windows and textfields[3]. Each of these objects has a fixed set of attributes: id, type, text, position (absolute and relative to the current point of attention) and size.

This relatively high level of knowledge representation was chosen because this was expected to be most useful for modeling interaction behavior. Also, this fits nicely with the level of representation available in the MS Windows environment.

An assumption underlying this level of knowledge representation is that the recognition of the user interface objects is easy and uniform for human subjects. This means that SHE cannot be used to model the problems in recognizing user interface objects, only humans that can easily identify buttons, labels, text and so forth can be modeled. This is not a major restriction, because most individuals that use devices can do this: people know what a button is.

Hand and eye The current implementation of SHE only has one hand with one finger, but SHE is designed to be easily extendible to two hands with five fingers each. SHE has one field of vision, divided into a point of gaze and a point of attention. The hand and eye are controlled through commands, discussed below.

All motor actions are assumed to first go through a preparation phase, which can be interrupted, before entering execution, which can not be interrupted. The duration of the various preparations and executions is extracted from various sources. The EMMA theory is used for eye movements (Salvucci, 2000). For the timing of the hand movements and actions, parameters were taken from EPIC and ACT-R/PM. However, these parameters did not suffice, mainly because ACT-R/PM is targeted to using a mouse for pointing, which is different from using hands and fingers for pointing. For these parameters, the experiment has also contributed, as described later.

Communication SHE communicates with the cognitive architecture through TCP/IP sockets. All modern programming languages have means to communicate using sockets, so using sockets for communication will allow for easy communication with any current and future cognitive architectures. Another advantage is that communication through sockets allows the cognitive architecture to run on a different machine in a network, which is an advantage in unstable situations or when the cognitive architecture is not implemented for the MS Windows platform.

Commands SHE responds to a set of commands that are used to set the point of attention and control the hand. Position arguments can be provided using absolute coordinates as well as relative coordinates. See Table 2 below for the most important commands.

There are also commands that control the simulated time and synchronization.

---

primarily models human interaction with computer applications instead of devices, and time constraints did not permit to further investigate technical details of ACT-R/PM, it is not integrated with SHE.

[3] Actually, the type of the objects is the type it has in the MS Windows environment. As a result of this, a button in an application created with Visual Basic has another type than a button in an application created with Visual C++. It is of course conceivable to make a mapping from MS Windows types to some set of SHE types, but in the current version this is not done.

Table 2: Important SHE commands.

Command	Arguments	Usage
attend	a position or an id	Directs the point of attention (this may induce eye movements)
motorhand	move, moveto, click, press or release; and a position or id	Performs the specified action
motoreye	move, moveto; with a position or id	Moves the eyes directly, instead of using attention shifts

Feedback SHE gives feedback on various processes. This feedback is sent over the socket in a Lisp-like list structure. The most important types of feedback are listed in Table 3.

Simulated time SHE maintains an internal queue of events. Whenever a command is issued, SHE puts this command on the queue as an event. A set of processes reacts on the events on the queue with appropriate behavior.

For example, when a command is issued to move the hand, this is translated to a motor-hand-move-begin event, which when processed puts a motor-hand-move-prep-begin event on the queue. In turn this event will result in a motor-hand-move-prep-end event on the queue, with its time set to the time of the begin-event plus the duration of the preparation. Then a motor-hand-move-exec-begin event is put on the stack, and so forth. Of course events also trigger appropriate behavior (e.g. updating the point of attention or sending mouse-clicks to Microsoft Windows).

The progress of time is controlled by commands from the cognitive architecture. The reason for this, is that in most cases the common cognitive models need an amount of time for calculations that is several orders of magnitude greater than the time the processes they simulate are assumed to last in real time. However, SHE itself is quite fast and can operate user interfaces in real time.

It should be noted that SHE explicitly restricts the cognitive models in their control over the hand and eye. SHE controls the duration and accuracy of visual encoding, movements and other actions.

Programming language The final implementation of SHE was written in Microsoft Visual C++. This enables easy access to user interface elements in MS Windows. Also, C++ is an object-oriented programming language, which implies that it will be easy to reuse parts of the tool in new versions of the tool.

Table 3: Types of responses from SHE.

Type	Contents
visibles	The currently visible objects with their id, size, absolute position and position relative to the point of attention
object	Information on an object to which attention was directed: as above, but including type and text.
poa	The status of the point of attention: free, prep or busy.
hand	The status of the hand: free, prep or busy.

Because cognitive architectures communicate with SHE using sockets, the cognitive architecture can be written in any programming language on any platform, as long as it supports TCP/IP sockets.

Future enhancements

Currently, interaction works satisfactory for interfaces created in Visual Basic, Visual C++ and Java AWT, for other integrated development environments (IDE's) we experienced some limitations. These IDE's include Delphi (partially) and Java Swing. This issue could be resolved by implementing special interfaces with these interfaces or by using the VisMap tool (St. Amant & O. Riedl, in press). SHE is designed to be extensible regarding this aspect.

Another enhancement could be to add other modalities of interaction like audition, speech and tactile feedback.

The experiment

An experiment aimed at gathering human data has been conducted for two reasons. First of all, not all parameters of SHE could be extracted from the literature. So I used this experiment to collect basic eye movement and motor control data for simple interaction tasks. This data will be used to fine-tune the parameter settings of SHE.

A second reason to conduct the experiment was to collect data on tasks that included interactive behavior based on explicit instructions. In this case I was especially interested in the higher level dynamics between low-level interaction behavior and cognition at a higher level.

The experiment was conducted using 15 subjects who were asked to perform a series of tasks. All subjects were students of the Industrial Design Engineering Faculty of the Technical University of Delft, aged 19 through 24. During the experiment, the movements and fixations of both of their eyes were recorded, as well as their finger presses. With one of the subjects no measurements could be performed. In another case the touch-screen software stopped responding, resulting in a partially failed trial, the first two tasks could be measured completely, so only the third task was discarded.

Experimental setup

The equipment used in the experiment was a touch-screen display, a simulation of a programmable thermostat (see Figure 4) and an eye tracker consisting of two small cameras to measure the pupils, a set of infrared transmitters and an infrared detector to measure the position of the head and software to analyze and log the measurements.

The programmable thermostat was simulated and presented to the subjects on the touch-screen display. It was operated by pushing the buttons on the screen. In its standard way of operation, it showed the current day of the week, time, temperature and program mode[4].

The eye cameras and infrared detector were mounted on a helmet. Subjects reported that this did not bother them in their movements, at least for the duration of the experiment, which was 15 to 20 minutes.

This entire setup was located at the faculty of Industrial Design Engineering at the Technical University of Delft.

The tasks

Three different tasks were given to the subjects, each of them repeated 10 times in a row.

The first task required the subjects to push all buttons in the middle row, from left to right. The reason for presenting this very simple task was to gather measurements on low-level interaction behavior.

The second task was to adjust the target temperature of the thermostat. This required the subjects to push a few buttons and to look at the display to wait for feedback. It was expected that subjects would show some learning effects, observable in their timing.

The third task included an instruction on how to adjust the clock of the thermostat. In contrast to the other tasks, in the 10 trials of the third tasks there was a slight variation. Each trial the clock of the thermostat was set to a different day and time, and the subjects were instructed to set the clock to a different day and time on each trial. The procedure to set the clock did not change, of course. This task was expected to provide the most interesting data regarding learning of instructions.

Preliminary results

Data format The following data was collected in the experiment (1) position, time and duration of eye saccades and fixations, sampled at 250 Hz (i.e. 4 ms/sample) and (2) time and duration of button presses,

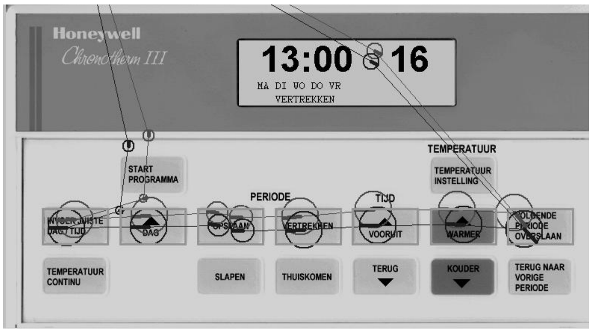

Figure 4: Sample of recorded eye movements.

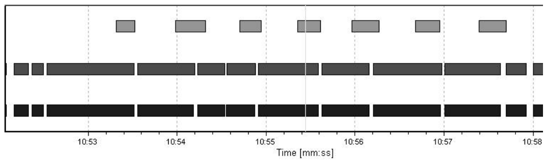

Figure 5: Sample of the combined measurements.

in milliseconds. For an example of the recorded eye movements, see Figure 4. Both sets of recordings are synchronized, thus the eye movements can be linked to the hand movements. This is shown in Figure 5, where the blocks in the upper row represent button presses and the blocks in the lower rows represent saccades of the left and right eye. The data still needs to be fully analyzed. Some preliminary results can, however, be addressed.

Possible EMMA enhancements A major result that might be derived from the collected data is a changed accuracy regarding the landing point of eye-saccades. In the EMMA theory used in SHE, the landing point of a saccade is assumed to follow a Gaussian distribution centered at the target of the saccade. However preliminary analysis of the data suggests a different distribution.

By examining the data, it becomes clear that indeed some saccades do not land at the intended position. Some saccades are very quickly followed by another saccade, which lands at a point nearby. This is predicted by the EMMA theory. However, the saccades that follow the inaccurate saccades are very rarely directed towards the previous origin of the saccade. This seems to indicate that undershoot happens more frequently than overshoot. One possible explanation could be that eyes do not “stop and return”, they “pause and continue”. This (micro)behavior seems reasonable, since it is conceivable that less energy is lost when “pausing” then when “stopping” the eyes, assuming that the eyes can fixate and process input when still moving with a (small) velocity.

[4] The program mode is however not relevant for the experiment and no further reference to it will be made.

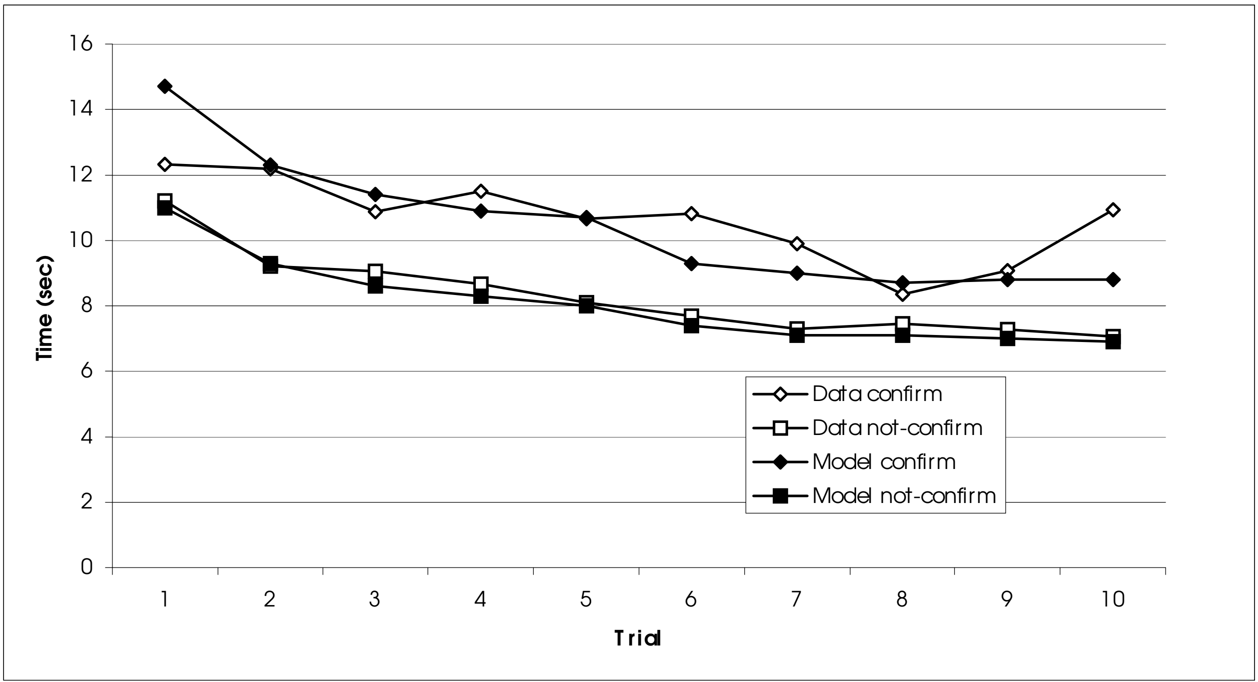

Figure 6: Data and model times for the first task.

In order to support this claim, further analysis is required. This analysis will be done by (1) selecting appropriate triplets of eye fixations in which the eye fixation in the middle has a small duration and the last fixation is spatially close to the middle fixation. Then calculate (2) the distance to the last fixation point and (3) the angle between the lines that (a) connect the first fixation-point and the second (inaccurate) fixation-point and (b) connect the second fixation-point and the final fixation-point. When it is indeed the case that undershoot happens more frequently than overshoot, this will be reflected in a higher frequency of angles in the range (-90°, 90°), possibly distributed following a Gaussian distribution with 0 at its center[5].

Models of interaction

To further test the tool, I have built various models of interaction in ACT-R, most of them to test certain specific details of SHE. The model of the first task fits the data, discussed below. The third task of the experiment is also partially modeled. Although not yet fully finished the models do show that it is relatively easy to make cognitive models that perform non-trivial tasks in interaction with fairly complex interfaces by using SHE.

[5] For this analysis it is necessary to assume that the last fixation is a correction to the short middle fixation. Therefore the middle fixation should have a very small duration, in order to assume that only the amount of visual input is processed that is needed to determine that the saccade was inaccurate.

One basic model All models were built on top of a basic ACT-R model. This basic model incorporates the use of instructions, based on the abstractions as introduced by Taatgen (1999, in press). This basic model contains all chunk-types, feedback-chunks and production rules necessary for interactive behavior.

The production rules operate on a set of standard instruction chunks that can be used to perform various basic interaction subtasks.

A model of the first task

The first task, in which subjects had to press the buttons in the center row, has been modeled successfully. The subjects showed two types of behavior and were divided in two groups accordingly. Subjects in the 'not-confirm' group (n = 10) pushed a button, looked at the next and pushed that one, repeated for all buttons. The subject in the 'confirm' group (n = 4) pushed a button and then looked at the display to look for a change, giving them feedback on whether the button was successfully pushed, after which they looked at the next button and continued.

Both groups were modeled using the same model that differed only in the way they handled confirmation of a button push.

Figure 6 shows the results. The time required for one trial was used as the measurement to fit.

Conclusions

The current implementation of SHE meets most of the requirements. It can be used to interact with most of the standard user interfaces in MS Windows, it can be operated in conjunction with common cognitive architectures and it simulates behavior comparable to human subjects interacting with user interfaces of devices. The latter can be further improved by further analyzing the data from the experiment and by using SHE for modeling more tasks.

Acknowledgements

The research presented here was made possible by support from KPN Research. The experiment was conducted in the laboratory for informational ergonomics at the faculty of Industrial Design Engineering at the University of Delft, with help from Theo Boersema, Kees Jorens and Arend Harteveld.

References

Aasman, J. (1995). *Modelling Driver Behaviour in Soar.* Doctoral Dissertation, Leidschendam, The Netherlands: KPN Research.

Anderson, J. R. & Lebiere, C. (1998). *The atomic components of thought.* Mahwah, NJ: Lawrence Erlbaum Associates.

Byrne, M. D., (in press). ACT-R/PM and menu selection: Applying a cognitive architecture to HCI. *International Journal of Human-Computer Studies.*

Byrne, M. D. (2000). The ACT-R/PM Project. *Simulating Human agents: Papers from the 2000 AAAI Fall Symposium* (pp. 1-3). Menlo Park, CA: AAAI Press.

Byrne, M. D. & Anderson, J. R. (1998). Perception and action. In J. R. Anderson & C. Lebiere (Eds.), *The Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum Associates.

Card, S. K., Moran, P. T. & Newell, A. (1983). *The Psychology of Human-Computer Interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Gray, W. D. (2000). The nature and processing of errors in interactive behavior. *Cognitive Science, 24*(2), 205-248.

Gray, W. D., & Altmann, E. M. (in press). Cognitive modeling and human-computer interaction. In W. Karwowski (Ed.), *International encyclopedia of ergonomics and human factors*. New York: Taylor & Francis, Ltd.

Kieras, D. E. & Meyer, D.E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction, 12.*

Ritter, F. E., Baxter, G. D., Jones, G., & Young, R. M. (2000). Cognitive models as users. *ACM Transactions on Computer-Human Interaction* 7(2).

Salvucci, D. D. (in press). An Integrated Model of Eye Movements and Visual Encoding. *Cognitive Systems Research.*

Salvucci, D. D. (2000). A model of eye movements and visual attention. In *Proceedings of the Third International Conference on Cognitive Modeling* (pp. 252-259). Veenendaal, The Netherlands: Universal Press.

St. Amant, R. & O. Riedl, M. (in press). A perception/ action substrate for cognitive modeling in HCI. *International Journal of Human-Computer Studies.*

Taatgen, N. A. (1999). *Learning without limits: from problem solving toward a unified theory of learning.* Ph.D. Thesis, Department of Psychology, University of Groningen, The Netherlands.

Taatgen, N. A. (in press). A Model of Individual Differences in Learning Air Traffic Control. *Proceedings of the fourth International Conference on Cognitive Modeling.*

University of Groningen

Artificial Intelligence

prepublications

AI-2001-7	**J. Misker, N.A. Taatgen & J. Aasman** Validating a tool for simulating user interaction
AI-2001-6	**N.A. Taatgen** Extending the past-tense debate: a model of the German plural
AI-2001-5	**G.R. Renadel de Lavalette & H.P. van Ditmarsch** Exact models for knowledge games
AI-2001-4	**H.P. van Ditmarsch** The semantics of concurrent knowledge actions (extended abstract)
AI-2001-3	**B.P. Kooi & H.P. van Ditmarsch** Speltheorie en logica
AI-2001-2	**B. Dunin-Keplicz & L.C. Verbrugge** Collective Intentions
AI-2001-1	**W. van der Hoek & L.C. Verbrugge** Epistemic logic: problems and puzzles
TCW-2000-13	**H.P van Ditmarsch** Knowledge games (article)
TCW-2000-12	**N.A. Taatgen & D. Wallach** Whether Skill Acquisition is Rule or Instance based is determined by the Structure of the Task
TCW-2000-11	**H.P. van Ditmarsch** Knowledge games (thesis)
TCW-2000-10	**N.A. Taatgen** Dispelling the magic: towards memory without capacity
TCW-2000-9	**N.A. Taatgen & J.R. Anderson** Why do children learn to say "Broke"?
TCW-2000-8	**H.P. van Ditmarsch** Axioms for cardgames
TCW-2000-7	**F. Dignum, B. Dunin-Keplitcz & L.C. Verbrugge** Agent architecture and theory for team formation by dialogue
TCW-2000-6	**H.P. van Ditmarsch** Dynamic knowledge logic
TCW-2000-5	**F. Stulp & L.C. Verbrugge** A knowledge-based algorithm for the internet protocol TCP
TCW-2000-4	**P. Hendriks** Het taalorgaan en andere linuïstische curiositeiten: de taalkunde in het nieuwe millennium
TCW-2000-3	**W. Teepe** “Wij kiezen voor u” Online Voting Advice
TCW-2000-2	**N.A. Taatgen** samenvatting van “Learning without limits”
TCW-2000-1	**N.A. Taatgen** Expertsysteem strandt op leergedrag expert

TCW-1999-15 **P.S. Hasper & L.C. Verbrugge**
Filosofie in tweeduizend en vierenvijftig lemmata

TCW-1999-14 **B. Dunin-Keplitcz & L.C. Verbrugge**
Collective motivational attitudes in cooperative problem solving

TCW-1999-13 **H.P. van Ditmarsch**
The semantics of actions in knowledge games

TCW-1999-12 **H. Prakken & G.A.W. Vreeswijk**
Logics for Defeasible Argumentation

TCW-1999-11 **G.A.W. Vreeswijk**
Necessary and Sufficient Conditions for a Version Space Representation Theorem

TCW-1999-10 **H.P. van Ditmarsch**
The logic of knowledge games: showing a card

TCW-1999-9 **L.C. Verbrugge**
Review of Epistemic Logic for AI and Computer Science

TCW-1999-8 **P. Hendriks**
The problem with logic in the logical problem of language acquisition

TCW-1999-7 **P. Hendriks**
Review of Projecting the Adjective

TCW-1999-6 **B. Dunin-Keplicz, F. Dignum & L.C. Verbrugge**
Dialogue in team formation: a formal approach

TCW-1999-5 **B. Dunin-Keplicz & L.C. Verbrugge**
The role of dialogue in collective problem solving: Part I

TCW-1999-4 **N.A. Taatgen**
A model of learning task-specific knowledge for a new task

TCW-1999-3 **L. Jongman & N.A. Taatgen**
An ACT-R model of individual differences in changes in adaptivity due to mental fatigue

TCW-1999-2 **N.A. Taatgen**
Implicit versus explicit: an ACT-R learning perspective Commentary on Dienes & Perner

TCW-1999-1 **N.A. Taatgen**
Cognitief Modelleren: Een nieuwe kijk op individuele verschillen

TCW-1998-5	**P. Hendriks** Waarom Plato s probleem niet van toepassing is op de verwerving van taal
TCW-1998-4	**N.A. Taatgen** Review of The Atomic Components of Thought
TCW-1998-3	**P. Hendriks & H. de Hoop** Optimality Theoretic Semantics
TCW-1998-2	**H.P. van Ditmarsch** User interfaces in natural deduction programs
TCW-1998-1	**C. Lebiere, D. Wallach & N.A. Taatgen** Implicit and explicit learning in ACT-R
TCW-1997-2	**N.A. Taatgen** Explicit learning in ACT-R
TCW-1997-1	**L.C. Verbrugge & B. Dunin-Keplicz** A reconfiguration algorithm for the maintenance of collective commitment

University of Groningen
Artificial Intelligence / Cognitive Science and Engineering (AI/TCW)
Grote Kruisstraat 2/1
9712 TS Groningen
The Netherlands
tel.: +31 50 363 77
fax. +31 50 363 84
http://www.ai.rug.nl/prepublications