Cognitive Modeling of Strategies in Dynamic Tasks

Alberto De Obeso Orendain (a.de-obeso-orendain@sussex.ac.uk)
Representation and Cognition Group, School of Informatics
University of Sussex, Falmer, Brighton, BN1 9QJ, UK

Sharon Wood (s.wood@sussex.ac.uk)
Representation and Cognition Group, School of Informatics
University of Sussex, Falmer, Brighton, BN1 9QJ, UK

Abstract

Computer simulations, or microworlds, have been used for studying various topics including problem solving. This work investigates strategies for complex, dynamic problem solving in a fire-fighting microworld. Using data from a study by Cañas, Antolí, Fajardo & Salmerón (2005), an ACT-R cognitive model is developed with the aim of providing insight into the development and selection of strategies participants use. One particular behavior observed in participants when trained repetitively on the same scenario, the creation of a fire-break barrier to prevent the fire spreading, is discussed. It was found that selection of a particular strategy depends on the fine-tuning of ACT-R production rule utilities as a consequence of environmental rewards, highlighting the role of reward size and timing. The model is able to capture various aspects of the data by promoting a free competition of small blocks of behavior based on rational analysis. A key finding is that good performance is linked to effective combination of strategic control with attention to changing task demands reflecting time and care taken in informing and effecting action.

Keywords: Cognitive Modeling; ACT-R; problem solving; strategy; microworlds.

Introduction

Microworlds are computer simulations that represent a middle point between naturalistic scenarios and laboratory tasks (Brehmer and Dörner, 1993). Although microworlds are relatively simple, they embody the essential characteristics of real-world dynamic decision-making environments (Gonzalez, Vanyukov and Martin, 2005). Microworlds allow for an economic and standardized presentation of scenarios, data registration and computing of results (Frensch and Funke, 1995; Brehmer and Dörner, 1993). These tasks have been used for studying various domains including problem solving (Frensch and Funke, 1995; Brehmer and Dörner 1993; Taatgen 2005).

Microworlds have three characteristics. Firstly, complexity, owing to the number of elements and number (and nature) of their interrelationship (Frensch and Funke, 1995). Second, lack of transparency; the problem solver does not have access to all relevant task information, making interaction with the world necessary for knowledge requirements. Last, the problem state changes both independently and as a consequence of the participant's actions. Microworlds consequently place a variety of cognitive demands on the problem solver. According to Anderson et al. (2004) dynamic tasks require considerable goal-directed processing within demanding perceptual displays and execution of motor commands under severe constraints. They require continuous processing of feedback in order to select appropriate actions within an ever-changing situation (Brehmer and Dörner, 1993). This paper focuses on the demands posed by these dynamic task characteristics, in particular the way performance feedback from a dynamic environment is processed, and how this allows the consolidation of strategies.

Frensch and Funke (1995) suggest that it is important to understand the process of Complex Problem Solving (CPS), rather than the product; this process is an interaction between the problem solver, the task and the environment. A cognitive model is able to reveal the internal processes for selecting actions together with their interaction with the environment, increasing our understanding of these processes. Cognitive modeling has been used in dynamic environments such as air traffic control (Taatgen, 2005). The work presented here uses the FireChief fire-fighting microworld (Omodei & Wearing, 1995).

The FireChief Microworld

FireChief participants combat fires spreading in a landscape using truck and copter units. Trials last 260 seconds. A FireChief scenario is specified by a variety of properties such as landscape distribution of forest, clearings and property, the number and position of initial fires, the direction and strength of the wind, and the initial position of fire-fighting units. Figure 1 shows the central cells of a FireChief trial display converted for model use. Copters (shown as CR) and trucks (TR) can move between landscape grid cells (R, L & H) and can Drop Water (DW) over cells to extinguish fires (F*n* where *n* indicates fire intensity). Copters are three times faster than trucks and cannot be destroyed by fire, but a truck's water tanks have twice the capacity and are able to Control Fire (CF) by creating a fire-break. Commands are issued through a combination of mouse and keyboard operations and their execution takes a fixed amount of time, 4 seconds to DW, 2 seconds to CF, and a variable amount of time to Move a unit depending on distance and type of unit. Wind strength and direction are in the upper right-hand corner of the display.

FireChief is a dynamic decision-making problem solving task environment where a series of interdependent decisions

are required to reach the goal, the environment changes over time, and user actions change the state of the world (Gonzalez et al., 2005). The problem solver is engaged in a strategic situation where he or she has control over a limited number of fire fighting units and has to use them to accomplish one mission: to fight and quell the fire. Task performance is inversely proportional to the number of cells destroyed by fire at the end of the trial.

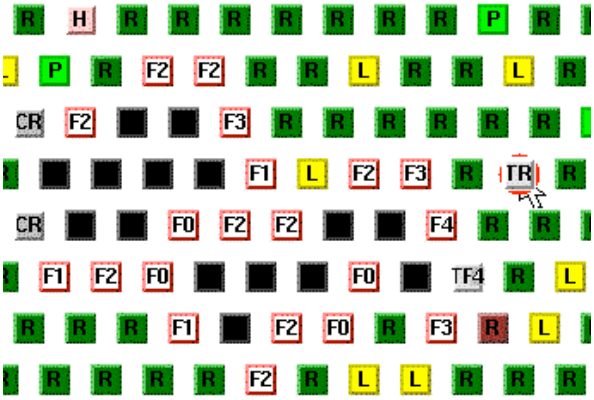

Figure 1: Central cells of the *model version* of a FireChief trial display using 'buttons' (Lisp)

ACT-R Architecture

The CPS model is implemented in ACT-R 6.0 (Anderson et al., 2004). ACT-R is divided into various modules according to the kind of information they process: a visual module for identifying objects in the visual field (in Figure 1 the focus of attention is on the cell in the fourth row of the penultimate column), a manual module for controlling the hands (the mouse pointer is located in the same cell), a declarative module for retrieving information from memory, and a goal and imaginary modules for keeping track of current goals and intentions. Communication between modules is achieved through buffers where the content of any buffer is limited to a single declarative unit of knowledge, a 'chunk'. Thus the system can only respond to a limited amount of information. Behavior in ACT-R occurs through interaction of its specialized modules via the buffers, coordinated by a central production system.

There are two types of knowledge in ACT-R: chunks encode declarative knowledge whereas procedural knowledge is represented by production rules, where each rule corresponds to a cognitive processing step. Each ACT-R production has two elements: the condition, a combination of states from the different buffers, and an action, which can perform transformations over the state of buffers and trigger actions in modules. ACT-R functionality is achieved through many mechanisms, but two are of the utmost importance in this model: utility and reward.

Utility designates the value of executing a rule; it represents the perceived value of a production and is updated by rewards from the environment. Utility of productions is compared during the process of conflict resolution where only the rule with the highest utility is acted upon. From a computational perspective, a participant can be considered as a collection of utility values. By interacting with FireChief, these utility values are tuned throughout a sequence of trials in a unique fashion within constraints imposed by the properties of the FireChief task, the procedural knowledge represented by rules, and rewards from the environment. The combination of ACT-R utility learning mechanisms with the dynamic nature of FireChief means the model can run a number of times under the same task conditions with the same knowledge and yet produce a different pattern of behavior each time. Rewards are the ACT-R mechanism for giving the model feedback from the environment. When a reward is triggered the utilities of all productions that have fired since the last reward are updated. The amount and distribution of rewards have an important impact on model's behavior (Janssen, Gray and Schoelles, 2008).

Human Study Data

The data used for specifying and fitting the CPS model comes from a study by Cañas et al. (2005). Those participants trained on the same, reliably predictable FireChief scenario for 16 trials were found to increasingly preferentially select the fire-fighting strategy that achieved the best outcome. This paper focuses on modeling strategy selection during constant training in order to understand this process and thereby gain insight into strategy formation. The constant scenario is characterized by a strong, constant easterly wind. Participants are limited to 2 copters and 2 trucks. To begin with there are two groups of fire in close proximity which quickly spread eastward (Figure 1 shows their initial distribution). A variety of different strategies can be used to stop the fire, as described in the next section.

Strategy Use

In total, 1728 protocols from 72 participants were analysed to identify four main strategies. In the *Non-Barrier* strategy CF commands are issued with noticeable spatial dispersion and are interleaved with DW commands. In the *Stop* strategy DW commands are used alone and are issued over the most intense fires within sufficient proximity to stop the fire. In the *Follow* strategy only DW commands are used but they do not target the strongest fires nor are they issued in close proximity to each other. The most structured strategy is called *Barrier* and it turns out to be very effective in the constant training scenario; it is used twice as often (50 vs. 27) by the top four performers compared to the four worst. For these reasons it is discussed here in more detail.

The Barrier strategy

The *Barrier* strategy presents a very characteristic way of dealing with the fire: the issuing of an ordered pattern of CF commands in a shape, similar to a barrier, intended to stop the fire spreading. There are many forms in which the barrier is created but a semicircle or straight line is the most frequent. In Figure 2 the barrier has the form of a semicircle where the black squares represent CF commands and the

grey squares represent DW commands. The strategy recruits top-down processes in constructing a fire-break but is sensitive to bottom-up perceptual processes so the final form of the barrier is a function of the shape of the fire that is being controlled.

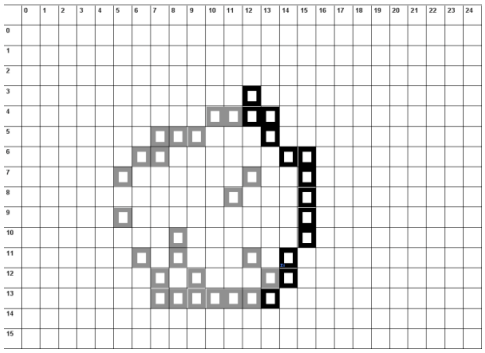

Figure 2: A typical *Barrier* strategy formation.

The Cognitive Model

To allow interaction between ACT-R and the FireChief task a Lisp version of FireChief was developed following the original specification provided in the FireChief manual (Omodei & Wearing, 1993). This is able to control all relevant aspects of the task: the landscape, development of fire, execution of commands, and performance calculations. Before running the model a FireChief scenario is loaded in an experimental window in the form of a matrix of multi-colored labeled buttons. Buttons enable interaction between the model and the experimental window by means of mouse and keyboard commands

The model implements all four main strategies, deciding which to use (based on initial utility comparisons) or switching to another (as utilities change) during the trial if the fire is not under control. An ineffective strategy, poorly rewarded, can be abandoned at any point, therefore. A chosen strategy is held in the imaginal buffer and affects model behavior by defining, for example, whether the model will use a mixture of DW and CF commands, whether or not a barrier will be created, or which ways of attacking the fire are preferred. In the very first trial the rules that select a strategy have an initial random utility determined by the standard ACT-R utility equation that has a random component. After the trial ends the utility of these productions is modified according to the final result. In this way, the actual means of executing a strategy emerges by rewarding certain rules over others (so a strategy is more precisely a *set* of strategies manifesting similar behaviour).

Creating a barrier

The functional block of rules described here belong to the set of strategies for creating barriers (see Figure 3). These rules represent a small subset of all the productions that are available to the model which is able to select and perform any of the four main strategies identified from the human data analysis. A FireChief trial lasts 260 seconds and a typical barrier is created in 60 seconds. Each cell in a barrier requires a Move followed by a CF command and the average number of grid cells needed for a barrier is 15. The average number of commands in a trial is 110.

First the model must specify a starting point for the barrier. This will depend upon the current state of fire and wind conditions. Second, the location of the next section of the barrier must be determined. A design decision was that the form of the barrier should be the result of a competition for locating the next cell of the barrier; top-down and bottom-up processes compete through the ACT-R conflict resolution mechanism. The selection of a target cell follows a process in which the candidate cell is proposed and then various tests (based on perceptual actions) are conducted. Third, a truck is moved to the selected cell before executing a CF command comprising a sequence of steps: locate the target, store location of target in working memory, find a truck, attend the unit, move the cursor to the unit, click the unit, attend target, move mouse to target, click mouse. Of these actions moving a cursor shows the highest time variability in the model (this information is not recorded in the human study protocols) stressing its importance in the total latency of the command and its corresponding importance to overall performance. When the truck has finished moving a CF command can be initiated. Fourth, the status of the barrier is monitored. Eventually, the barrier is considered complete when the fire-break is sufficient to contain the fire. The shape of the resulting barrier is a product of competition between various rules and the reward they receive when executing commands.

In the excerpt shown in Figure 3, the model is following the *Barrier* strategy and has just started a Move command with a truck. The current intention of the model is to create a fire-break barrier using CF commands. In step 1 the model must choose between waiting for the truck that has initiated its movement (and is disabled until it arrives) or using the other truck. In this step the utilities of productions 1-A and 1-B are compared and the one with the highest expected value is fired. In this case the model decides to wait. In step 2, the model searches for a visual-location that satisfies a set of constraints. In this example the model is verifying if the truck has arrived at its destination. The first constraint is spatial: the column and row of the destination cell. The second constraint is graphical: the cell must have a light-grey color (if the destination cell is white it means that the truck is still moving). The result of this search determines step 3. If the truck has not yet arrived, the model returns to step 1. When the model detects that the truck has arrived at its destination a shift of attention is made to that location. At the end of this attention shift the visual buffer is loaded with a chunk representing the content of the cell, namely the type of landscape and whether the cell is on fire (plus its intensity). Step 4 starts by checking whether the visual chunk encoded in the visual buffer is a product of an explicit shift of attention or the product of *buffer stuffing*. Buffer stuffing is an ACT-R mechanism in which a chunk is stored in the visual buffer without an explicit request from a

production rule. This can be a recurrent source of distraction for the visual system but also allows the detection of unforeseen events (for example new fires appearing in the scenario). In this example, if the model is distracted a visual chunk (that does not represent the location details for where the CF is going to be executed) is placed in the visual buffer. If the model proceeds with step 4 it will move the mouse pointer to the cell that distracted its attention instead of the correct cell. If the visual element encoded in the visual buffer is a product of the explicit attention shift executed in step 3, the CF command can be applied there because now the unit is in position. Before issuing a CF command the mouse pointer must be located over the truck, so step 4 initiates a mouse movement towards the attended cell. During this time the target cell may catch fire; in this case the model aborts the execution of the CF command. In step 5, after the mouse movement is complete, the CF command is initiated by pressing a key. In the normal flow of events the CF command would start after the click. Figure 3 shows a different outcome: just after rule 5-A fires the target cell catches fire, rendering the execution of a CF command impossible and consequently an alarm is emitted. Following this, the model is able to detect this alarm and, making use of the contents of the imaginal buffer, can select an appropriate course of action based on its strategy choice.

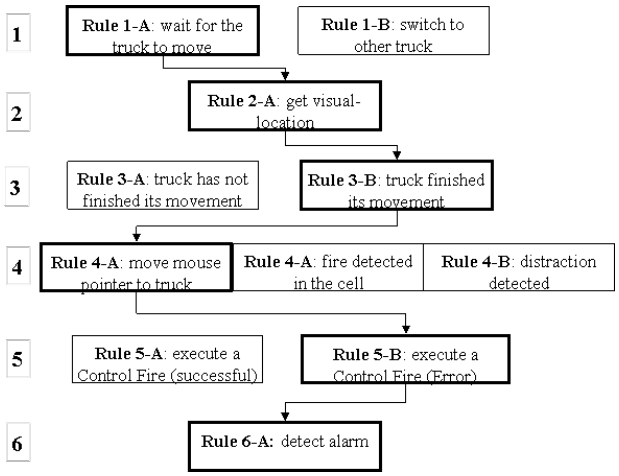

Figure 3: Sequence of Barrier strategy rules

A model run lasts 4160 seconds (16 sessions of 260 seconds). The model was run 40 times, following the same experimental design as in the Cañas et al. (2005) study. The data generated by the model provides a complete protocol of interaction with FireChief, as for each human participant, as well as a detailed trace of the operations being executed inside its various modules.

Data Fitting

During initial development, the simplest natural model was implemented based on a GOMS (Card, Moran, and Newell, 1983) analysis of the task and then fitted to the human study data. This initial model was highly efficient: all units were used all the time so time wasted was negligible. This initial model also followed a rigid strategy specification; however, the data reveals that participants do not use time as efficiently as in the initial model nor do they repeatedly execute the same strategy, making the importance of achieving flexibility in behavior evident. The approach adopted was to provide the model with complete knowledge about all the available strategies (cf. Gray & Boehm-Davis, 2000) but to allow them to compete freely based on their perceived utility.

Various reward schemes were tried, the most successful being the one that focuses on individual commands. In the 'single reward' scheme a reward (based on final performance) is given at the end of the trial. In the 'reward sub-task' scheme the completion of salient tasks is rewarded. For example, in the *Barrier* strategy stages are completion of a barrier, refilling a unit, or extinction of the fire. The problem with both these schemes is that, because several hundreds of rules may fire between rewards, the utility values of the most recent rules are changed only. This affects the model's behavior because the rules responsible for achieving good performance may not receive the proper reward and hence appropriate learning is deterred. In the scheme selected for use here positive rewards are awarded for successfully completing individual commands and negative rewards for executing unsuccessful commands and wasting time. Executing Move and CF commands generates a fixed amount of reward but the reward of a DW command is a function of the intensity of the fire that is extinguished.

In fitting the model there was no attempt to obtain the exact behavior of any individual; rather, data fitting centered on identifying decision points, encoding rules for executing actions and assigning rewards.

Results

Three metrics are used here to compare behaviour: task performance (reflecting appropriate strategy use); command duration (reflecting underlying cognitive and other processing steps); and interactions between commands (reflecting performance-related functional relationships between the Move and the CF and DW commands). There are other metrics not discussed here.

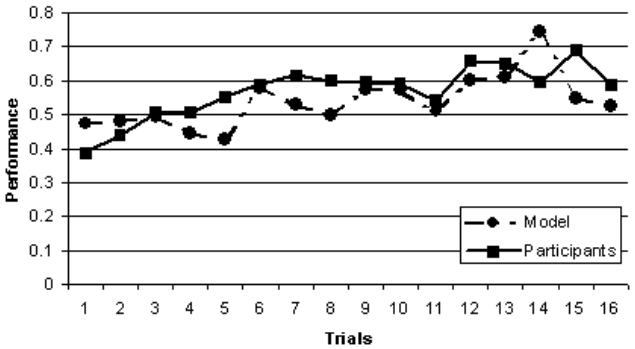

Figure 4: Comparison of performance between model and Cañas et al (2005) study participants

Figure 4 compares performance in the constant training condition for participants and the model. As can be seen, the model is able to replicate performance levels and also capture the incremental improvement in performance (R=.538). A significant performance increment was obtained by comparing the first and last four trials for both participants and the model. ($F(1,33)=4.417$, $p<.05$ and $F(1,33)=5.17$ $p<.05$ respectively).

	Performance		Frequency(%)	
Strategy	**Data**	**Model**	**Data**	**Model**
Barrier	81.59	81.05	0.65	0.66
NonBarrier	72.38	71.74	0.17	0.18
Stop	91.98	71.3	0.02	0.11
Follow	57.69	66.42	0.16	0.06

Table 1: Strategy use during constant training trials

Table 1 shows that *Barrier* is the most frequently used strategy during constant training[1]. Due to the high wind strength in the constant training scenario it is very difficult to stop the fire using DW commands only.

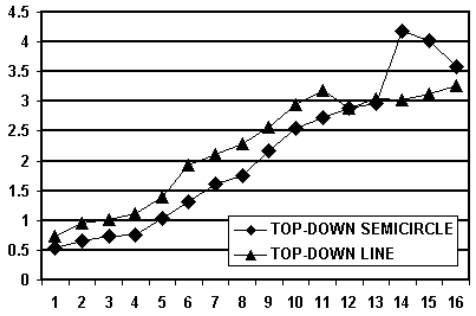

Figure 5: Increase in production utilities during consolidation of the *Barrier* strategy

A typical run of the model involves around 200 decisions, and the execution of each decision requires between 1 to 6 rules. On average the model executes 103 commands and participants execute 110 commands per trial. The model improves performance due to the tuning of its production utilities to the constant training trial scenario. Figure 5 shows how the utility of productions related to the creation of the barrier steadily increases as trials are completed. This continuous increment of utility values implies that FireChief commands are being completed with success with more frequency over trial runs.

[1] The good performance shown in the human data for the *Stop* strategy is based solely on two participants who used it extremely successfully from the outset whilst other less proficient participants rapidly abandoned it in favour of more reliable strategies.

Good vs. Bad Performers

A comparison of the best and worst performers in the constant training condition is presented with the aim of showing how utility values can be used for understanding more about participant behavior. Performance metrics for the top four participants in the constant training group are compared with the worst four participants and the same is done with model data. The best performers use the *Barrier* strategy twice as often as the worst performers (50 vs. 27 times) and performers/model-runs have an average performance per trial of 86.80/87.41 while the worst performers have an average performance of 70.91/71.80 when using this strategy.

All participants and model-runs, take a similar amount of time to issue a CF command that forms part of a fire-break barrier ($F(78,1)=.637$, $p=.427$) and ($F(81,1)=1.792$, $p=.185$), so the performance differences do not lie here. However, there is a functional dependence between moving a unit and issuing a CF command. Before executing a CF command the truck must be moved to the right place. The model embodies the assumption that the decision about where to move the truck is taken when the execution of the movement is initiated. There is a significant difference between the best and worst participants in the time it takes to execute a movement prior to issuing a CF command when forming a barrier ($F(1260,1)=67.980$, $p<.001$). The model captures latency times for the best performers only; worst performers spend much less time on this activity than the model. The best approximation to worst performance provided by the model is to execute only a single perceptual action to ascertain the fire location without checking whether the target fire-break cell is on fire. The model uses the fire-front for selecting where in a particular row the next fire-break cell should be, and poor performers often get this wrong (see next section). Even so, the model remains slower than participants by 800ms. on average. Even if all perceptual and cognitive processing could be removed from the model it cannot reduce the time taken by a sufficient amount to match human latencies. An explanation for this could be connected to the duration of motor commands: a Move command requires two key-presses and two mouse pointer moves. Perhaps poor performers execute these actions with more hastiness. Evidence to support or refute this explanation is subject to ongoing research

Utility profile

With the aim of gaining insight into what differentiates best and worst performers, two profiles were created based on utility values for each group from the model run. To obtain the profiles, the utility of relevant productions for each group is queried at the end of the training phase and averaged. In doing this, the comparison is focused only on the rules relating to the creation of a barrier: the way trucks are used, how they are moved, and how the barrier is created.

The comparison shows that the most striking difference between good and bad performers is that good performers

successfully combine top-down and bottom-up processes to create a barrier, while the worst performers apply only top-down processes successfully, failing to combine them well with bottom-up processes so that cells selected for the fire-break prove less effective. The key differences are that the best performers pay more attention to the fire-front, and also that they wait for the trucks to finish their (short) movements before executing a CF command, thereby completing the sequence of commands successfully. These differences can be identified by looking at the utility values of the productions that compete at the relevant decision points (as in Figure 5).

Discussion

This paper is focused on the adaptive selection of strategies for fire fighting with the aim of demonstrating how cognitive modeling can improve our understanding of problem solving behavior when interacting with dynamic microworlds, with implications for real-world complex problem solving. The model continuously interleaves cognitive with perceptual-motor operations, selects different strategies and implements them according to the reward structure of the task. A particular implementation of a strategy depends on the fine-tuning of ACT-R production rule utilities as a consequence of environmental rewards and thus is a product of both the configuration of the trial (in this case the constant training trial) and the history of interactions between problem solver and task (which is stored in the collection of utility values). As noted by Cañas et al. (2005) the constant training condition allows participants to consolidate strategies (see Figure 5).

The most important learning mechanism for the model is the one that updates utility. The main objective during the fitting of the model was to allow rules to be rewarded (or punished) by their effects in the environment, however the set of available strategies was not altered. In other words, fitting the model was restricted to affecting the competition between strategies.

This work highlights the role of size and location of rewards for strategy selection. As pointed out by Janssen, Gray & Schoelles (2008) the definition of reward has an important influence on model behavior. Due to the large number of rules being fired in each trial, it is necessary to arrive to an appropriate reward frequency to enable appropriate learning. Rewarding productions for their effectiveness in successfully completing individual commands seems a good criterion; however, in doing this it is important to identify where cognitive effort is made. In the case of FireChief relevant cognitive effort for e.g., placing a new section of barrier, is traced to the time a sequence of actions is initiated prior to the final successful movement being executed, and not just when that final CF command is issued (that is, there is a causal link between the CF command and those actions previously taken).

The process by which a barrier is created is only one amongst many others that occur during a model run. A similar analysis based on utility comparisons can be carried out for other strategies by identifying the rules that govern them. Understanding strategy selection as a consequence of previously learned utility also offers a means to understand more about performance differences. Worst performers reflect a different pattern of utility values in rules used for the creation of the fire barrier, owing to impoverished attention to the dynamic problem solving state and apparent lack of care in issuing commands. Overall the work presented demonstrates that complex dynamic tasks can be fruitfully explored through a cognitive modeling approach. By providing a loose strategy definition the model is able to implement complex patterns of behaviour which in turn are able to successfully stop the fire while replicating many other aspects of the human study data.

Acknowledgments

This research is funded by CONACYT.

References

Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review,* 111, (4), 1036-1060.

Brehmer, B., Dörner, D.B. (1993) Experiments with computer-simulated microworlds: Escaping both the narrow straits of the laboratory and the deep blue sea of the field study. *Comp. in Human Behavior*, 9, 171-184.

Cañas, J.J., Antolí, A., Fajardo, I., Salmerón, L. (2005) Cognitive inflexibility and the development and use of strategies for solving complex dynamic problems: effects of different types of training. *Theoretical Issues in Ergonomic Science*, 6 (1) 95-108.

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

Frensch, P.A., Funke, J. (Eds.) (1995) *Complex problem solving: The European Perspective*, Hillsdale, NJ: Lawrence Erlbaum.

Gonzalez, C., Vanyukov, P., Martin, M. (2005) The use of microworlds to study dynamic decision-making. *Computers in Human Behavior,* 21, 273–286

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.

Janssen, C.P., Gray, W.D., Schoelles, M.J. (2008). How a modeler's conception of rewards influences a model's behavior: Investigating ACT-R 6's Utility Learning Mechanism. *Proceedings of the 15th Annual ACT-R Workshop,* Pittsburgh, PA, p42.

Omodei, M.M., Wearing, A.J. (1995). The FireChief microworld generating program: An illustration of computer simulated microworlds as an experimental paradigm for studying complex decision-making behavior. *Behav. Res. Meth. Instr.& Comp.*, 27, 303–316.

Taatgen, N.A. (2005). Modeling parallelization and speed improvement in skill acquisition: from dual tasks to complex dynamic skills. *Cognitive Science*, 29, 421-455.