Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

# Modelling an Empirical Investigation into Memory and Learning in Simple Interactive Search Tasks

**Juliet Richardson, Andrew Howes and Stephen J. Payne**
School of Psychology, University of Wales, Cardiff,
P.O.Box 901, Cardiff, CF1 3YG, U.K.
+44 1222 874007
RichardsonJ@Cardiff.ac.uk

## ABSTRACT

In this paper we address the issue of how initial menu search experiences are encoded and then used to guide subsequent search. We report empirical data from participants searching in a menu structure in which they cannot use spatial strategies and are therefore required to use just the labels to guide themselves. We then describe two cognitive models of menu search: the AYN model which encodes recognition chunks for tried options and gradually acquires positive and negative control knowledge; and an activation-based model which increases the activation of seen and tried options and then uses these activation levels on subsequent trials to guide its search. The data from the activation-based model provides the better fit to the empirical data.

### Keywords

Interactive Search, Cognitive Models, Memory, Learning, Computer Menus

## INTRODUCTION

Searching through menu structures is a common method of interacting with computers: using software packages, browsing the world-wide web and searching databases are just some of the tasks that require menu search (or *interactive search*). The task of interactive search can be specified in basic terms as requiring a person to make selections in order to find a particular goal. They can either select an option[1] to move forward down a branch of the menu structure, or select an operator to move back up the menu structure (either back just one step or back to the initial starting point). The task of interactive search is therefore different from other problem solving tasks in that people initially do not know what the outcome of operators (moves) will be until they are tried.

In this paper we summarise an empirical investigation of interactive search (for a full report see Howes, Richardson and Payne, in preparation), together with two possible cognitive models of interactive search which are then assessed against the empirical data. We are especially interested in understanding how memories encoded during the initial search experience shape behaviour on subsequent searches for the same goal. In particular, how does a user learn the sequence of choices that leads to a particular goal? The delay between the time when a menu option is selected and the time when that option can be evaluated as correct or not (when the goal is achieved) makes this task more difficult than it might first appear. In many instances, incorrect paths will be explored before the correct route to the goal is found. The user must learn to distinguish those options which were tried and found to be incorrect from those which eventually led to the goal.

One of the most obvious guides as to which options to select during initial search in an unfamiliar menu is the semantics of the labels: labels which are closely related to the current goal should be better choices than those which are more distantly related to the goal (Franzke, 1994; Franzke, 1995). For example, given the goal of checking the spelling of a document in Microsoft Word, the menu header "Tools" seems like a better choice then "Insert" or "Font". However, the label semantics are rarely a sufficient guide to the correct route to a goal. In the above spell-check example, both "Tools" and "Format" might seem equally good choices to a novice user.

There have been several previous cognitive models of how people search in menu structures where the semantics are not sufficient, such as, the IDXL model developed by Rieman, Young and Howes (1996) and the model of expert search behaviour developed by Kitajima and Polson (1995). However, these have tended to focus on the initial search process and the question of how to decide which options to select. Whilst such models are candidate models of how experts and novices search during initial exploration of a menu structure they do not address the problem of how memories of that search are encoded and subsequently used: they leave open the question of how people perform a menu search task for the second, third or fourth time, or how performance improves with experience. A start has been made at addressing these questions with the AYN model (Howes, 1994).

One of the first questions that we can ask is how the initial search is encoded. The experience could be encoded just in terms of the menu labels. For example, the spell-check task might be encoded as selecting "Tools" followed by "Spelling". We term this a lexical encoding. In addition or alternatively, people might exploit the spatial structure of the menu tree and encode their search experience in terms of some spatial representation of the menu structure and the spatial location of the goal within that representation. For example, the spell-check task might be encoded as selecting an item towards the right of the menu bar and then selecting the first item under it.

In addition, it is also possible that users rehearse their choices during the initial search process. For example, at

[1] We use the terms "options", "selections", "choices" and "items" interchangeably to refer to the labels at a menu node.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

any one time, users could attempt to rehearse the sequence of choices leading to their current position in the menu structure. Upon reaching the goal, the most recently rehearsed sequence would be the correct route. We would expect rehearsal of this type during search to give rise to a primacy effect (improved performance for the first items in the sequence as they will have been rehearsed for a longer time than the later items).

Alternatively, a recency effect (improved performance for the last items in the sequence) might emerge if users reflect on the actions that they have just performed when they reach the goal (e.g. Howes, 1994).

Therefore, one way to investigate the question of how initial search experiences affect subsequent learning is to look at the order in which the sequence of options leading to the goal are learnt. We use the term "effect of levels" to refer to this differential learning of options at different levels.

## EMPIRICAL INVESTIGATION

In this experiment (described in full in Howes, Richardson and Payne, in preparation), we wished to investigate the order in which the choices leading to the goal were learnt, independent of factors such as semantics which may differentially affect different decisions. For example, if one choice was between two semantically plausible options and a second choice was between one highly plausible option and one that was implausible, we would expect the second choice to be learnt more readily than the first. In order to avoid potentially confusing effects of semantics such as this in our experimental data, we used menu trees constructed entirely from labels which had no semantic relationship to the goals. Thus, at each node in these menu trees the level of semantic guidance to the correct choices was the same and there should be no differential effects.

Such semantically unhelpful menu trees are not entirely unrealistic. As pointed out above, semantics are a far from perfect guide in many real-life menus and users are often faced with selecting between two (or more) equally plausible or implausible options.

In addition, we also wished to investigate how people's performance is affected if all possibility of forming spatial encodings is removed and they are forced to just use the labels. In order to achieve this we used two groups of participants. The first group performed the menu search task with randomised positioning of the labels at each node (each time a participant visited a node, the label positions might or might not be swapped around). This manipulation should prevent these participants from encoding their experience spatially. The second group of participants performed the search task in "normal" menu trees where the positioning of the labels at each node was kept constant over time. This should allow us to see what the effect is on learning and performance when participants are forming only a lexical representation of the menu tree in terms of just the labels as compared with when they can also encode spatial aspects of their experiences.

### Method

Thirty-two undergraduate students from the Psychology Department at the University of Wales, Cardiff took part in this study for course credits. The experiment consisted

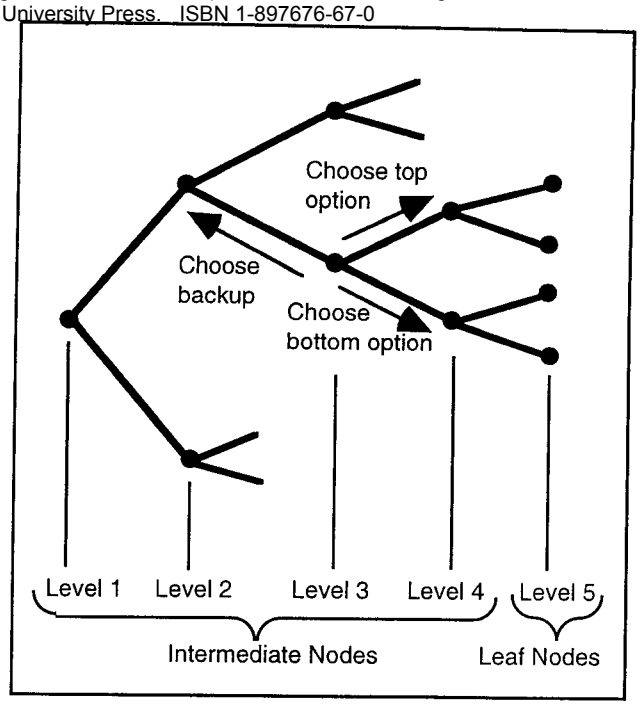



Figure 1: Illustration of the design of the trees used in the experiment.

of seven trials. On each trial the participant was asked to search for the first target in the first menu tree, followed by the second target in the second menu tree, the third target in the third menu tree and the fourth target in the fourth menu tree. In order to produce a balanced design the order of presentation of the four menu trees was manipulated in order to ensure that each menu tree was presented equally often first, second, third and fourth. The experiment was presented on an Apple Macintosh computer using a program written in MacProlog32. This program automatically recorded the choices made by participants and the time taken to make them.

Each menu tree consisted of five levels with binary choices between a top and a bottom label at each node, as illustrated in Figure 1. The target was one of the choices at a leaf node. At each node there were two options that could be selected to move forward down the tree and a backup option to move back up the tree (except at the top-level root node, where backup is not possible, and in the target node, where backup would allow the participant to review the choices leading to that target). Pairs of semantically related words (e.g. "Carbon" and "Charcoal") were used for the option labels at nodes in these menu trees. These label pairs were placed in different random positions for each participant. There were no close semantic relationships between the label pairs both within and between trees as determined by the experimenters' judgement.

One between-participants factor was manipulated in this experiments. Participants were randomly allocated to one of two equally sized groups. For one group, at each node, the two options were positioned randomly in the top and bottom positions on each visit. For the other group, the two options at each node appeared consistently in the same positions throughout the experiment.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

## Results

The main results that we are interested in modelling are the order in which participants learnt the sequence of choices leading to the target (effect of levels), and the improvement in performance over trials. Therefore, we shall only consider those analyses. (The full set of analyses can be found in Howes, Richardson and Payne, in preparation).

The effects of levels were investigated by seeing whether participants selected the correct or the incorrect option at the nodes leading to the target. For each target, there was a correct sequence of five actions that would lead directly to that target. For each of the nodes on this correct path, the percentage of correct options chosen by participants on their first visit to that node on each trial was calculated. The action taken on the first visit to each node on each trial was used because this should reflect the effects of long term memory, rather than any effects of temporary memory for recent local search sequences. This measure should therefore show how participants' memory for the correct actions developed with experience.

The data for the trial 2 levels effect are summarised in Table 1. These data were subjected to an Anova to check for main effects of level and node label positioning and for any interaction between these factors. There were no significant main effects of node positioning: $F(1, 30) = 0.756$, $p = 0.39$, nor of level $F(3, 90)=0.959$, $p = 0.42$. There was not a significant interaction between these variables $F(3, 90) = 2.15$, $p = 0.099$. However, t-tests revealed that there were significant differences between levels 1 and 3 for the consistent condition, but that there were no significant differences between any levels in the randomised condition.

The same analysis was carried out for all trials except the first. The data are summarised in Table 2 and were subjected to an Anova to test for main effects of main effects of trial, node option positioning and level and for interactions between these factors.

There was no significant main effect of level, $F(3, 90) = 0.58$, $p = 0.63$ on the total number of correct actions. However, there was a significant interaction between the positioning of node options and level, $F(3, 90) = 2.72$, $p < 0.05$. There was a significant effect of levels when the positioning of the node options was consistent: the percentage of correct choices made at levels 1 and 2 was higher than at levels 3 and 4. This primacy effect was most pronounced on trials 2 and 3. There was no effect of levels when the label positioning was random: on all trials there were no significant differences between performance at different levels.

Table 1: The mean percentage of correct choices made by participants on the first visit to nodes at levels 1, 2, 3 and 4 on the correct path on trial 2.

| | Node option positioning | | | |
|---|---|---|---|---|
| | Randomised | | Consistent | |
| Level | M | S.D | M | S.D. |
| 1 | 52% | 35% | 72% | 22% |
| 2 | 58% | 26% | 67% | 27% |
| 3 | 58% | 26% | 48% | 27% |
| 4 | 61% | 23% | 61% | 21% |

Table 2: The mean percentage of correct choices made by participants on the first visit to nodes at levels 1, 2, 3 and 4 on the correct path averaged over trials 2 to 7.

| | Node option positioning | | | |
|---|---|---|---|---|
| | Randomised | | Consistent | |
| Level | M | S.D | M | S.D. |
| 1 | 73% | 28% | 79% | 20% |
| 2 | 76% | 22% | 79% | 23% |
| 3 | 78% | 22% | 70% | 26% |
| 4 | 79% | 24% | 72% | 22% |

In addition, there was a significant main effect of trial, $F(5, 150) = 35.26$, $p < 0.05$ on the total number of correct choices made. Correct choices increased over trials 2 to 4 but not thereafter. There was no significant main effect of node option positioning, $F(1, 30) = 0.07$, $p = 0.79$ on the total number of correct choices. There were no other significant interactions.

Performance over trials was calculated in terms of the average number of actions taken to reach the goal on each trial. These data are summarised in Table 3. These data were subjected to an Anova to check for main effects of trial and node label positioning and for any interaction between these factors. There was no significant main effect of positioning of node options, $F(1, 30) = 2.09$, $p = 0.16$. There was a significant main effect of trial, $F(6, 180) = 52.99$, $p < 0.01$. The number of actions taken to reach the goal decreased significantly over the first four trials but not thereafter. There was no significant interaction between positioning and trial, $F(6, 180) = 0.84$, $p = 0.54$.

## Conclusions

When spatial consistency was removed the correct choices at all levels within the menu structure were learnt at the same rate. In comparison, when the menu structure was spatially consistent, participants learnt the choices at the top levels first (primacy effect). This result suggests that participants in the spatially consistent condition might have been carrying out some form of spatial rehearsal whilst performing the initial search. The lack of

Table 3: The mean number of actions taken by participants to reach the goal on each trial.

| | Node option positioning | | | |
|---|---|---|---|---|
| | Randomised | | Consistent | |
| Trial | M | S.D | M | S.D |
| 1 | 60.2 | 20.6 | 52.5 | 20.6 |
| 2 | 39.6 | 31.8 | 24.3 | 12.7 |
| 3 | 26.9 | 24.7 | 20.8 | 14.4 |
| 4 | 18.1 | 13.6 | 12.8 | 9.3 |
| 5 | 14.8 | 17.7 | 11.5 | 11.4 |
| 6 | 11.0 | 13.5 | 8.1 | 4.4 |
| 7 | 14.6 | 26.9 | 7.5 | 3.3 |

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

either a primacy or a recency effect when participants were forced to rely just on the labels to guide their search suggests that no lexical rehearsal took place.

However, even when participants had to rely just on using the labels, they could still learn to perform the task as quickly (in terms of the total number of actions taken to reach the goal) as when the label positions were left constant over time. Therefore, even though participants in the random positioning condition did not appear to be using lexical rehearsal, they were still able to learn the correct choices with practice. Possible accounts of how this might occur are discussed below in the context of two possible cognitive models of the data.

## COGNITIVE MODELLING

The initial goal was to develop a cognitive model of learning in menu trees without spatial consistency (i.e. label-based learning only). Such a model can then be used as a starting point for a model of learning in the spatially consistent menu trees, where participants appeared to be using spatial rehearsal.

The main test of the model will obviously be its degree of fit to the experimental data described above. The model should therefore show a flat effect of levels (i.e. equal rate of learning of the choices on the path leading to the goal), together with improvement in performance over trials. Ideally the model should not only show the same pattern of data as the empirical participants, but also the same values. For example, its performance (in terms of the number of actions taken to reach the goal) should improve over the first four trials only but not thereafter.

Two models of label-based interactive search are: (1) The AYN model (Howes, 1994) which encodes chunks for tried items, uses this knowledge to limit the search space on subsequent trials and learns that the most recently selected item is correct when it finds that it is on the right path. (2) An activation-based model which boosts the activation levels of the representations of tried and seen items and then makes decisions based on the relative activation levels to guide its selections.

## COGNITIVE MODEL 1: AYN

The first model of interactive search that we describe is the AYN model (Howes, 1994). AYN acquires two types of knowledge as it interacts with a menu structure: *recognition* knowledge and *control* knowledge.

The recognition knowledge consists of episodic chunks that are encoded for every combination of goal, menu and action that the model experiences, regardless of whether the action in question leads to the goal or not. AYN also acquires recognition knowledge that the goal has been achieved. This recognition knowledge supports identification of the menu trees that have been previously visited, which selections made and which goals visited.

AYN uses its recognition knowledge to help guide search in the menu structure during both initial exploration and subsequent searches. A set of rules determines how the model applies this knowledge: (1) if the goal has not yet been achieved then avoid recognised selections; (2) if the goal has been achieved and there is a recognised selection then it should be applied; (3) if there are no recognised selections and the goal has been achieved then a backup operator should be applied. These rules help limit the size of the search space.

AYN also acquires both positive and negative control knowledge through its exploration of the menu structure. This knowledge determines which menu selections lead to the goal and which lead to dead-ends. In AYN working memory is bounded to store only the previous action. Thus, when the goal is achieved AYN only learns positive control knowledge for the selection immediately preceding the goal. On the next trial, when AYN reaches the selection known to be right (i.e. the one before the goal), it learns positive control knowledge for the immediately preceding selection that led to it. In this way positive knowledge is passed back up the structure in a *final-first* way until positive knowledge has been learnt for all the selections leading to the goal.

AYN acquires negative control knowledge in a similar way for selections that lead to dead-ends. In fact, the AYN model was altered slightly from the version reported by Howes (1994) in order to get it to learn negative knowledge from backing up, rather than from cancelling and returning to the start state. AYN was altered so that it learnt that a particular move was "bad" either if that move led directly to a dead-end or if that move led to a node where both options were rated as "bad".

The AYN model was run fifty times (for seven trials on each run) over a five-level binary menu tree (i.e. the same structure as that used in the experiments) to generate the data. The data generated from the model should therefore be in a form that is comparable with that obtained empirically.

### Results

The effect of levels on each trial was calculated in terms of percent correct selections made on the first visit to each of the nodes on the correct path. The data for trial 2 only are summarised in Table 4 and Figure 2. The 95% confidence interval was calculated for the empirical mean obtained at each level in the menu (see Grant, 1962, for a discussion of this type of analysis). These confidence intervals are shown in Figure 2. None of the means generated by the AYN model fell inside the confidence interval at any level. At each level, the means generated by the AYN model were higher than those of the experimental participants.

The data for the effect of levels averaged over all trials are summarised in Table 5. The correlation between the percentage of correct choices made at each level on each trial by the AYN model and by the experimental participants was calculated. The correlation was fairly poor, $r = 0.747$, $r^2 = 0.559$.

Table 4: The mean percentage of correct choices made by AYN and the activation-based model on the first visit to nodes on the correct path on trial 2.

| | AYN | | Activation-based model | |
|---|---|---|---|---|
| Level | M | S.D. | M | S.D. |
| 1 | 72% | 45% | 52% | 50% |
| 2 | 76% | 43% | 56% | 50% |
| 3 | 68% | 47% | 54% | 50% |
| 4 | 100% | 0% | 48% | 50% |

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

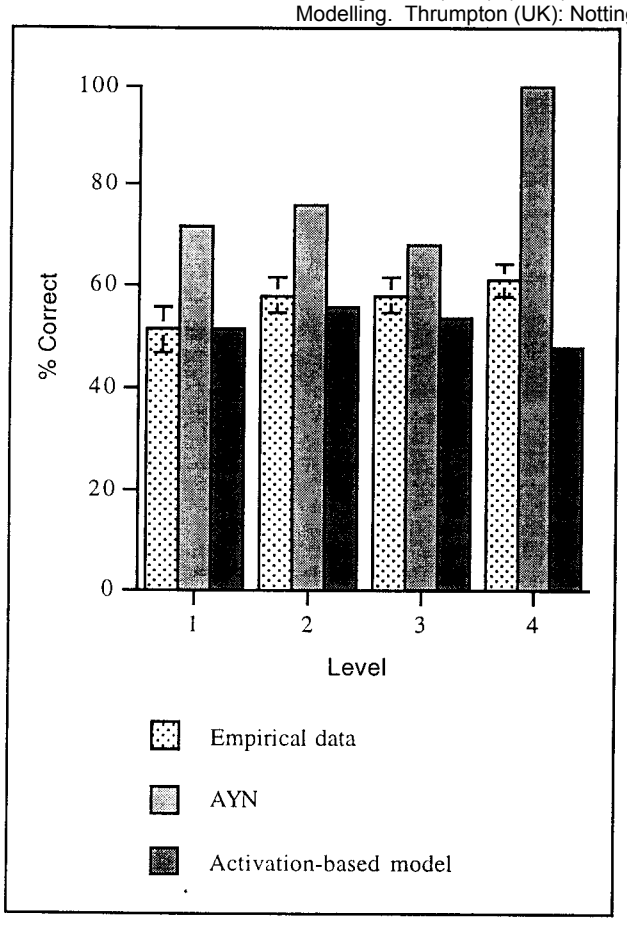



Figure 2: The percentage of correct selections made on the first visits to nodes on the correct path on trial 2 by the empirical participants, the AYN model and the activation-based model

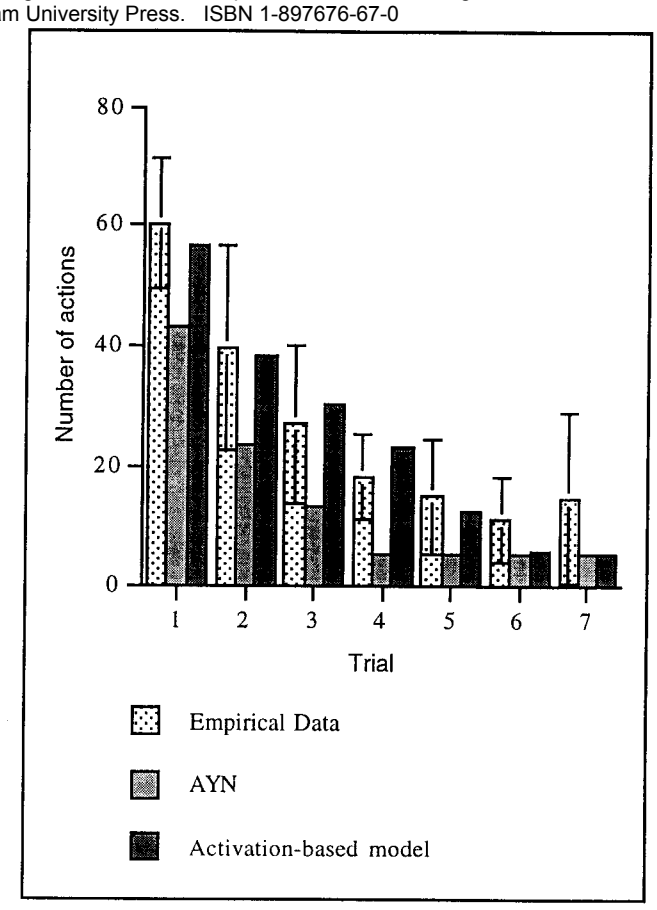



Figure 3: The number of actions taken to reach the goal on each trial by the empirical participants, the AYN model and the activation-based model

The number of actions taken by the AYN model to reach the goal on each trial was calculated. These data are summarised in Table 6 and Figure 3. The 95% confidence interval was calculated for the empirical mean obtained on each trial, as shown in Figure 3. The means generated by the AYN model fell outside the confidence interval on trials 1, 3, 4 and 5. On each of these trials, the means generated by the AYN model were lower than those of the experimental participants. The correlation between the data generated by the AYN model and the empirical data was calculated. The correlation was very good, $r = 0.991$, $r^2 = 0.982$.

Table 5: The mean percentage of correct choices made by AYN and the activation-based model on the first visit to nodes on the correct path averaged over trials 2 to 7.

| | AYN | | Activation-based model | |
|---|---|---|---|---|
| Level | M | S.D. | M | S.D. |
| 1 | 90% | 20% | 83% | 29% |
| 2 | 93% | 14% | 87% | 26% |
| 3 | 95% | 8% | 84% | 31% |
| 4 | 100% | 0% | 84% | 29% |

**COGNITIVE MODEL 2: ACTIVATION-BASED MODEL**

The second cognitive model of interactive search that we consider is a simple activation-based model which makes more refined judgements than the AYN model. This model does not just distinguish tried from untried options, but makes four classifications of options: untried; seen and possibly tried; definitely tried; and very recently tried. Most importantly, this model does not acquire any form of AYN-like control knowledge. Instead it simply uses the relative activation levels to determine which choices are correct and which are incorrect.

Table 6: The mean number of actions taken by AYN and the activation-based model to reach the goal on each trial.

| | AYN | | Activation-based model | |
|---|---|---|---|---|
| Trial | M | S.D | M | S.D |
| 1 | 43.3 | 26.4 | 56.4 | 38.9 |
| 2 | 23.6 | 21.6 | 38.0 | 32.8 |
| 3 | 13.2 | 15.8 | 30.2 | 35.7 |
| 4 | 5.2 | 0.6 | 23.1 | 31.5 |
| 5 | 5 | 0 | 12.5 | 21.2 |
| 6 | 5 | 0 | 5.6 | 2.7 |
| 7 | 5 | 0 | 5.1 | 0.9 |

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

In this activation-based cognitive model, when an option is seen its activation is boosted by 10 units, and when an option is selected its activation level is boosted by a further 40 units (unseen options have an activation level of zero). Every time a move is made (selection of an option or selecting backup) the activation of all other options decays by 1%. Therefore with time the activation levels of previously tried and seen options decrease. There are 110 decay cycles between trials (to simulate the intervening tasks in the experiments).

The model assesses the activation levels of the options that it encounters in order to infer whether options have been seen or tried before. It then uses these inferences to determine which action to select. If the activation level of an option is less than 1 unit, then the model infers that that option has never been seen before (= untried). If the activation level is between 1 and 20 units then it infers that the option has definitely been seen before and could possibly have been tried some time ago as well (= seen-and-possibly-tried). If the activation level is between 20 and 40 then it infers that the option has definitely been tried before (= definitely-tried), and if the activation level is above 40 then it assumes that the option was tried very recently (= very-recently-tried). The model uses its assessments of the activation of the possible options at a node, together with knowledge of whether the goal has already been achieved or not, in order to decide which action to take.

If the goal has not yet been achieved, the model uses a simple search algorithm similar to that of the AYN model. It avoids options that are assessed as being definitely-tried or very-recently-tried and selects those that are assessed as being untried or seen-and-possibly-tried. At a node with two options which are untried or seen-and-possibly-tried, it prefers to select the untried option. At a node with one option that is definitely-tried or very-recently-tried and one that is untried or seen-and-possibly-tried, it selects the untried or seen-and-possibly-tried option. At a node with only definitely-tried or very-recently tried options, or at a node that is a dead-end, it selects backup. In this way the model searches efficiently through the menu structure to reach the goal.

Once the goal has been achieved, the model again uses a search algorithm based on that of the AYN model. It prefers to select an option that has been assessed as being definitely-tried before, but not very-recently-tried. If not it will select an option that is assessed as seen-and-possibly-tried. It does not select untried options or very-recently-tried options. It also backs up from deadends.

The model was run fifty times over a five-level binary menu tree, for seven trials on each run, to generate the data.

### Results

As before, the effect of levels on all trials was calculated in terms of the percentage of correct choices made on the first visit to each of the nodes on the correct path. The data for trial 2 only are summarised in Table 4 and Figure 2. For this model, the means for the percentage of correct choices made at levels 1 and 2 on trial 2 fell within the 95% confidence intervals for the empirical means. The mean percentage correct choices for levels 3 and 4 fell below the confidence interval: the model made fewer correct choices at these levels, on average, that the experimental participants.

The data for the effect of levels averaged over all trials are summarised in Table 5. The correlation between the percentage of correct choices made at each level on each trial by the model and by the experimental participants was calculated. The correlation was good, $r = 0.917$, $r^2 = 0.840$.

The number of actions taken to reach the goal on each trial was calculated. These data are summarised in Table 6 and Figure 3. The means generated by the model fell within the 95% confidence interval for the empirical means on all trials. The correlation between the data generated by the model and the empirical data was calculated. The correlation was very good, $r = 0.964$, $r^2 = 0.930$.

## CONCLUSIONS

The data generated by the AYN model provided a good fit to the shape of the empirical practice data, although its performance was higher, as would be expected given its 100% accurate all-or-nothing recognition. However, for the effect of levels, the correlation of the AYN data to the empirical data was not as good: AYN showed a recency effect in the learning of the choices on the correct path (i.e. better performance for the last item), whereas no such effect was seen in the empirical data. In addition, the overall level of correct selections made by AYN at the nodes on the correct path was much higher than that of the empirical participants.

The activation-based model gave a very good fit to the empirical data for learning based on labels alone. The correlation between its data and the empirical practice effect data was similar to that seen for AYN. However, unlike the AYN model, the curve that it generated did not differ in absolute value from the empirical data. The data from the activation-based model also provided a reasonable fit to the empirical levels effect data. Its correlation to this data was much higher than the AYN model. In addition, although the curve that it generated had a slightly different shape to the empirical data, the absolute values were not different for two of the four means.

The activation-based model is therefore able to learn the correct menu choices at the same speed and in approximately the same pattern as empirical participants without recourse to either rehearsal or the use of AYN-like control knowledge. Learning occurs simply through the gradual increase in activation of the correct choices relative to other choices.

A slight, but important difference between the activation-based model and the empirical data is that the experimental participants showed a slight (but non-significant) recency effect whereas the model showed an almost completely flat levels effect. However, further trials of the model showed that when the delay between trials is reduced, the model begins to show a recency effect similar to that of the experimental participants. It would be interesting to see whether the small recency effect exhibited by the participants alters with the delay between tasks in a similar way. There is some evidence in straightforward recognition tasks that delay affects

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

recency effects in this way (Wright, Santiago, Sands, Kendrick and Cook, 1985).

## GENERAL DISCUSSION

The simple activation-based model provided a better fit to the empirical data for searching a spatially inconsistent menu structure than the AYN model. Importantly, the activation-based model learns the correct path without manifesting a recency effect. This is due to the fact that it doesn't acquire explicit control knowledge. Instead the activation levels of each of the correct choices gradually increases relative to the other choices. On average this rate of relative increase is the same for all of the choices leading to the goal and so a flat effect of levels was observed.

In addition, this result showed that making a simple all-or-none distinction between tried and untried options (as AYN does) led to better performance than was seen empirically. Instead it seems likely that in reality menu users might occasionally be unsure as to whether a particular item has been selected before or merely seen. Errors will therefore arise when users select items that have merely been seen before and not tried. Such uncertainty is akin to a feeling of mere familiarity for a menu item and can be contrasted with definite recollection that an item has been tried before (see Jacoby, 1991, and Mandler, 1980, for an account of the distinction between familiarity and recollection and Payne, Richardson & Howes, in preparation, for an account of the role of familiarity in guiding menu search behaviour). There is currently some debate as to whether familiarity and recollection are indeed separate processes or just end-points on a continuum (see for example, Dodson & Johnson, 1996; Jacoby, 1991). However a simplified version of the single-process model of recognition can be seen as analogous to the decision process underlying the activation-based model. This simplified model assumes that there is a single quantity (activation levels, in our case) underlying different recognition judgements. If the activation level is above a certain criterion, an item will seem merely familiar, whereas if the activation level is above another, higher, criterion the item will be recollected. The activation-based model can therefore be thought of as preferring "recollected" selections over "familiar" ones. It occasionally makes errors by selecting, on the basis of their familiarity, items that had only been seen before and not actually tried. This model therefore had a lower overall level of performance that was not significantly different from that of the empirical participants.

However, this activation-based model only accounts for the data obtained in the situation where participants had to rely on the labels alone and could not exploit the spatial consistencies within the environment. In other words, this model only simulates the possible *lexical* encodings that a person might form whilst navigating through a menu structure, it does not account for any *spatial* representations that might be constructed. As shown in the experiment reported earlier, when spatial consistency was provided, people seemed to perform some form of spatially-based rehearsal. The activation-based model should therefore be extended to model this type of performance, perhaps by adding another "layer" which rehearses spatial location whilst searching through the menu structure.

There are several other possible avenues of development for the activation-based model. One of the first changes to explore might be the effect of different functions for the rate of decay. For example, research shows that the rate of forgetting might be governed by a power law (see Anderson, 1995, for an account). Another possibility is to explore the effects of adding in features such as associations and spreading activation between the activated representations. However, the evolution of this model to date has been driven largely by the goal of modelling empirical data, and any future architectural developments will therefore be made only in response to empirical data that challenge the model.

Finally, both the AYN model and the activation-based model as described here do not use the semantic plausibility of the menu items to guide their search. Other experiments that we have carried out suggest that people do use semantic plausibility, in conjunction with recognition memory, to determine which choices to select in a menu (Payne, Richardson & Howes, in preparation). Both models have, in other versions, been altered to use the semantics of the labels to guide their choices. For both models the effect of this is effectively to limit the search space to just the subset of the menu labels that are judged to be semantically plausible.

## ACKNOWLEDGEMENTS

The research reported in this paper was funded under the ESRC's Cognitive Engineering Initiative, Grant No. L127251029.

## REFERENCES

Anderson, J. R. (1993). *Rules of the mind.* London: Lawrence Erlbaum Associates.

Anderson, J. R. (1995). *Learning and memory: An integrated approach.* Chichester: John Wiley and Sons, Inc.

Dodson, C. S. & Johnson, M. K. (1996). Some problems with the process-dissociation approach to memory. *Journal of Experimental Psychology: General, 125*, 181-194.

Franzke, M. (1994). *Exploration, acquisition and retention of skill with display-based systems.* Ph.D. Thesis, Department of Psychology, University of Colorado, Boulder.

Franzke, M. (1995). Turning research into practice: characteristics of display-based interaction. *Proceedings of the Conference on Human Factors in Computing Systems* (New York, NY.), Association for Computing Machinery, 421-428.

Grant, D. A. (1962). Testing the null hypothesis and the strategy and tactics of investigating theoretical models. *Psychological Review, 69*, 54-61.

Howes, A. (1994). A model of the acquisition of menu knowledge by exploration. In B. Adelson, S. Dumais and J. Olson (Eds.) *Proceedings of Human Factors in Computing Systems CHI'94* (Boston, MA.), ACM Press, 445-451.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

Howes, A. Richardson, J. & Payne, S. J. (in preparation). Strategies and representations for interactive search tasks.

Jacoby, L. L. (1991). A process dissociation framework: Separating automatic from intentional uses of memory. *Journal of Memory and Language, 30*, 513-541.

Kitajima, M. & Polson, P. G. (1995). A comprehension-based model of correct performance and errors in skilled, display-based, human-computer interaction. *International Journal of Human-Computer Studies, 43*, 65-69.

Mandler, G. (1980). Recognizing: The judgement of previous occurrence. *Psychological Review, 87*, 252-271.

Payne, S. J., Richardson, J. & Howes, A. (in preparation). Strategic use of familiarity and plausibility in display-based problem solving.

Rieman, J., Young, R. M. & Howes, A. (1996). A dual-space model of iteratively deepening exploratory learning. *International Journal of Human-Computer Studies, 44*, 743-775.

Wright, A. A., Santiago, H. C., Sands, S. F., Kendrick, D. F. & Cook, R. G. (1985). Memory processing of serial lists by pigeons, monkeys and people. *Science, 229*, 287-289.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

# LICAI+: A Comprehension-Based Model of The Recall of Action Sequences

**Muneo Kitajima**
National Institute of
Bioscience and Human-Technology
1-1 Higashi Tsukuba Ibaraki 305, JAPAN
Tel: +81 (298) 54-6731
E-mail: kitajima@nibh.go.jp

**Rodolfo Soto and Peter G. Polson**
Institute of Cognitive Science
University of Colorado
Boulder, CO 80309-0345, USA
Tel: +1 (303) 492-5622
E-mail: {soto, ppolson}@psych.colorado.edu

## ABSTRACT

This paper presents a model of occasional use of functions of an application by an experienced user of an environment like Windows 95 or the MacOS. We have developed a simulation model, LICAI+, that assumes that users store episodic records of correct steps discovered by exploration or told to them during training. They then use the application display and their goal as retrieval cues in attempts to recall these episodes later. The model predicts, and supporting data show, that tasks that violate the label-following strategy are not only hard to learn by exploration but also difficult to remember even if the correct steps have been previously presented.

### Keywords

cognitive model, learning by exploration, label-following strategy, LICAI+

## INTRODUCTION

Experienced users of an environment like Windows 95 or the MacOS are *occasional* users of many applications (e.g., a graphics package). Furthermore, many functions of a frequently used application like a word processor are only used occasionally (e.g., constructing and editing a table). Thus, a large majority of the *different* tasks undertaken by skilled users are performed infrequently (Santhanam & Wiedenbeck, 1993).

Such patterns of occasional use should constrain the design of usable computer systems. Ideally, such systems should consistently support learning by exploration. At a minimum, they should facilitate memory for action sequences learned by demonstration or by being looked up in a manual. The ease of recalling infrequently performed functions can be a major determinate of usability. This is not a novel claim. For example, the designers of the Xerox Star had very similar insights (Bewley, Roberts, Schroit, & Verplank, 1983; Smith, Irby, Kimball, Verplank, & Harslem, 1982). This paper presents a theoretical model of recall of tasks that have been done once or a few times and data supporting the model.

LICAI+ is a model of recall of occasionally used action sequences. LICAI+ assumes that users store episodic records of correct steps discovered by exploration or told to them during training. They then use the application display and their goal as retrieval cues in attempts to later recall these episodes. The resulting model of the recall process is similar to models of text recall (Wolfe & Kintsch, submitted).

LICAI+ is an extension of LICAI[1] (Kitajima & Polson, 1996; 1997) which is a model of the processes involved in comprehending task instructions and using the resulting goals to guide successful exploration. Both LICAI and LICAI+ are based on Kintsch's (1986; in press) construction-integration theory of text comprehension. LICAI+ adds to LICAI the processes involved in encoding and successfully retrieving encodings of correct actions. LICAI+ assumes that successful performance of occasionally performed tasks involves a mixture of recall of episodes of correct actions and problem solving if recall fails. The model is related to Ross' (1984) and Rickard's (1997) models of skill acquisition.

Following a general description of the LICAI+ model, we present a theoretically motivated analysis of recall of occasionally performed action sequences. Readers interested in a more detailed descriptions of the LICAI model should consult (Kitajima & Polson, 1995; 1996; 1997). In support of the LICAI+ model and our theoretical analysis we compare our simulation results with data reported by Franzke (1994; 1995) and Soto (1997). In conclusion, we describe design implications of our results. We demonstrate that *both* ease of learning by exploration and good recall are supported by similar attributes of an interface.

## DESCRIPTION OF LICAI+

LICAI+ simulates skilled Mac users in an experiment where they are taught novel tasks using a new application, Cricket Graph III. The task instructions are very explicit but do not contain any information about how to perform the task. Then, at some later time ranging from several minutes to a week, they are tested for retention of these skills when given the task descriptions and the displays generated by the application as retrieval cues. Users attempt to perform each task by exploration and/or recalling an action sequence. However, hints are given by the experimenter if users cannot discover correct actions by themselves.

---

[1] LICAI is an acronym of the Linked model of Comprehension-based Action planning and Instruction taking. When LICAI is pronounced [li kai], the pronunciation represents a two-kanji Japanese word, 理解, meaning 'comprehension.'

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

LICAI simulates comprehension of task instructions and hints, the generation of goals, and the use of these goals to discover correct actions by exploration. LICAI+ adds to LICAI processes that encode successful actions and retrieve them after a delay.

### Goal Formation

LICAI's action planning processes contain limited capabilities to discover correct actions by exploration. These processes are controlled by *goals* generated by comprehending task instructions and hints. LICAI assumes that goal-formation is a specialized form of the normal reading process in which task specific strategies generate inferences required to guide goal formation. LICAI's goal-formation process is derived from Kintsch's (1988; in press, Chapter 10) model of word problem solving.

Kintsch's model takes as input a low-level semantic representation of problem text, the *textbase*, and processes it sentence by sentence. The result is a *problem model.* Construction of the problem model makes extensive use of comprehension schemata which elaborate the original text representation with problem domain specific inferences.

LICAI incorporates comprehension schemata that transform relevant parts of the textbase for the task instructions and hints into goals that control the action planning process. Propositions that describe actions on task objects in the textbase are recognized and further elaborated by specialized task domain schemata to generate a more complete description of a task. For example, consider a graphing task in which the user was given the instruction, Plot a variable named 'Observed' as a function of a variable named 'Serial Position.' LICAI transforms this task description into the propositional representations of two sentences. 1) Put 'Observed' on the y-axis, and 2) Put 'Serial Position' on the x-axis. The representations of the last two sentences are then transformed into *task goals* that control the action planning process. Terwilliger and Polson (1997) demonstrated that users actually perform this transformation.

In the studies described in this paper, experimenters gave hints of the form 'perform a specific action on a specified screen object' (e.g., pull-down the **Options** menu). LICAI requires that these text or verbal descriptions of an action on an object have to be transformed into a goal, a *do-it goal*, that specifies a specific object on the screen and/or legal actions on that object. Specialized comprehension schemata carry this transformation. See Kitajima and Polson (1997) for extensive descriptions of comprehension schemata.

### Action Planning

The heart of LICAI is the action planning processes. LICAI assumes that successful action planning involves linking propositional representations of a goal (e.g., create a new graph), the screen object to be acted on (e.g., the **Graph** menu), and an action to be performed on that object (e.g., press and hold). The most critical of the three links is the link between the goal and the correct screen object. This link can be retrieved from memory or generated by an exploration process.

#### *Skilled Users*

Kitajima and Polson (1995) developed a version of the action planning process used by *skilled* users of an application. This model represents an arbitrary sequence of actions required to perform a task as hierarchical goal structure that is retrieved from long-term memory and used to generate the actions. A task is decomposed into a sequence of task goals. *Task goals* refer to actions (e.g., edit) on a task object (e.g., graph title). Each task goal is linked to an ordered sequence of one or more *device goals.* Each device goal specifies a unique object on the screen (e.g., the **Options** menu, the graph title) and the state of the object (e.g., highlighted) after it has been acted on. Thus, skilled users retrieve the critical links between goal and screen object from memory. However, Kitajima and Polson (1995) did not describe how such goal sequences are learned or how they are retrieved from memory.

#### *New Users*

When a *new* user of an application attempts to perform a task for the first time, Kitajima and Polson (1997) assumed that they have a task goal but not the device goals. LICAI can simulate exploration by generating the correct actions for a novel task without the device goals if the task goal can be linked to correct screen objects by LICAI's action planning processes.

A task goal is a proposition with two arguments describing a task action and a task object (e.g., hide legend). If a correct object on the screen has a label representing either one of these concepts (e.g., a menu labeled "hide"), the representation of the object will be linked to the task goal. LICAI will retrieve the correct actions (e.g., move the cursor to the object and press-and-hold) on this object from long-term memory, completing the necessary links to generate actions. We and numerous other researchers have called this linking process *the label-following strategy* (Franzke, 1994; Franzke, 1995; Kitajima & Polson, 1997; Polson & Lewis, 1990;. Rieman, Young, & Howes, 1996). Thus, the critical links can be generated to mediate successful exploration. The label-following strategy is the only method that LICAI has for learning by exploration. If there is no direct link between the task goal and the correct object, users must be given a hint.

### LICAI+'s Encoding and Recall Processes

LICAI already incorporates a model of encoding and recall of goals based on the Kintsch and Welsch (1991) model of text recall. They assumed that the textbase is stored in episodic memory during the comprehension process. The strength in episodic memory of a given element of the textbase is determined by the number of cycles it stays in working memory and the activation levels it achieves during each cycle. LICAI+ generalizes this model to the encoding and recall of successful actions. LICAI+ also incorporates assumptions from the Wolfe and Kintsch

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

(submitted) model of story recall that enables us to compute predicted recall probabilities.

*Encoding Process*

LICAI+ assumes that encoding and storage of a successful action is just a special case of the comprehension process. The model "comprehends" the results of a successful action during training. A comprehension schema creates a representation of the successful action which is stored in memory during the comprehension process.

There are two forms of this encoding. The first includes the task goal, the object acted on, and results of the action if the label-following strategy can discover the correct action. The second case is defined by the failure of the label-following strategy. The experimenter gives a hint which is transformed into a do-it goal by the instruction comprehension processes. A do-it goal specifies an action on a screen object (e.g., Pull-down the **Options** menu). The do-it goal is included in the encoding of the successful action in this second case.

LICAI+'s goal formation, action planning, encoding, and retrieval processes are implemented as special cases of Kintsch's (1988; in press) construction-integration theory of text comprehension. Each process is modeled by one or more iterations of a general construction-integration cycle.

The following is a description of the encoding and recall cycles. See Kitajima and Polson (1997) for detailed descriptions of the remaining processes.

The construction phase of the encoding process generates a network of propositions that contains the following representations:

1) the task goal,
2) the do-it goal (if a hint was given),
3) the acted-on object,
4) its label (if the acted-on object is labeled),
5) salient changes in the display state caused by the action (e.g., menu dropped),
6) the display caused by the action (e.g., a pull-down menu),
7) a special encoding node that links the nodes 1, 2, 3, 4, and 5 with the strengths defined by an analyst.

In addition, the fundamental linking mechanism assumed by the construction-integration theory, the argument overlap mechanism, is applied to connect any two propositions in the network sharing arguments. Figure 1 illustrates a network generated for encoding a step of pulling down the **Legend** menu. This action caused a pull-down menu to appear with menu items, **Hide**, **Show**, **Move**, and **Arrange**.

The integration phase of the encoding process is performed using a spreading activation process. The nodes in the network can be partitioned into sources of activation, targets of activation, and links between sources and targets. In the encoding process, the representations of screen objects, the task goal, and the do-it goal serve as sources of activation. In Figure 1, these nodes are shaded. The encoding node is the target. The results of the integration of the network are stored in episodic memory.

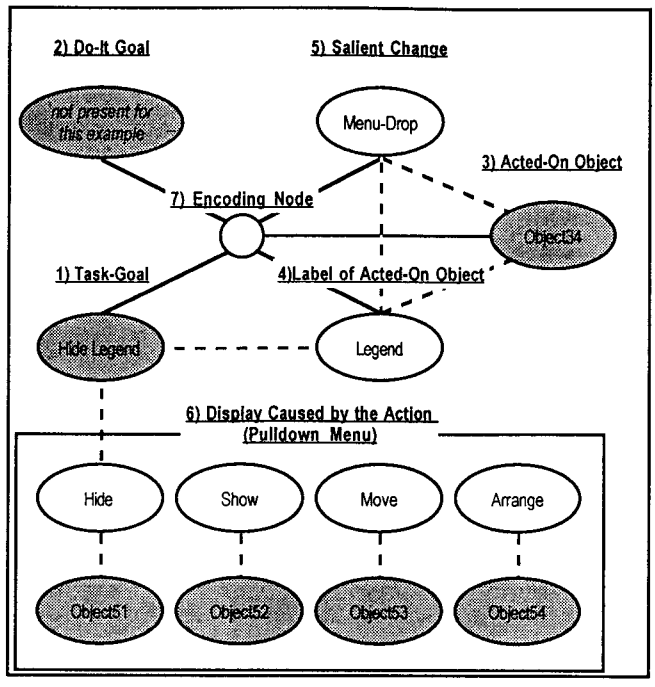



**Figure 1.** A diagram showing the propositional network generated by the construction subprocess in the encoding process. The dotted lines represent the argument overlap links. The solid lines connecting nodes, 1 through 5, with the encoding node, 7, are special links defining the encoding process.

At the end of training, episodic memory contains the nodes representing the textbase for the task instructions and hints, and the nodes participated in encoding processes for the correct steps. The strengths of links between these nodes are determined by the pattern of activation levels achieved in respective integration processes for text comprehension and encoding.

*Recall Process*

The recall process of LICAI+ assumes that users employ the task goal and the current display representation as retrieval cues. The recall process retrieves nodes in episodic memory that are linked to these cues. Nodes from episodic memory are sampled with replacement until the model retrieves an encoding of a step or retrieves a do-it goal (i.e., the action planning representation of a hint).

The predicted sampling distribution for retrieving nodes from episodic memory for a given set of retrieval cues is calculated by using a sampling probability matrix. This matrix is a fully interconnected matrix generated from the original episodic memory network. Following Wolfe and Kintsch (submitted), the sampling probability matrix is generated by two steps: 1) dividing each link strength in the episodic memory network by the maximum link strength, 2) for any two nodes linked by an indirect path, assigning the product of the strength values of the link segments in the path to their link strength.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

Any nodes that are directly linked with the retrieval cues in the sampling probability matrix are retrievable. The probability of retrieving a retrievable node in a single memory sampling trial is proportional to its relative link strengths with the retrieval cues.

Sampling is with replacement, and sampling terminates on retrieval of one of the step encodings or a do-it goal. These assumptions enable us to calculate the recall probability distribution for step encodings and do-it goals (recall targets).

*Action Planning After Recall*

LICAI+ attempts to act using the retrieved step encoding or the hint. If the step encoding or the hint generates the correct action, the model successfully recalls the current step. However, there are no explicit order cues in the encoding of each step, so the model can retrieve steps out of order or retrieve hints that don't apply to the current display. In this case, the retrieval process fails, and the model has to explore the interface again as on the training trial. The exploration will succeed in performing the correct action if the label-following strategy works for this step.

## AN ANALYSIS OF RECALL OF OCCASIONALLY PERFORMED TASKS

The basic claim of LICAI+ is that how a step in a task is learned, by exploration or with hints, determines how that step is encoded and retrieved. Thus, we distinguish between label-following (LF) steps or tasks, and non-label-following (NLF) steps or tasks where the label-following strategy fails for lack of linking shared concepts.

Franzke (1994; 1995) and many others have shown that LF steps are rapidly discovered and "accurately" recalled. However, it is hard to distinguish between rediscovery and recall of a step after one training trial because both recall and discovery processes can have similar latency distributions.

Soto (1997), in an analysis of a large number of different graphing tasks using Cricket Graph III, showed that NLF tasks have some LF steps, usually toward the end of their action sequences. The task 'hide legend' is a good example. The first two steps (pull-down the **Options** menu, and select **Show Graph Items...**) are NLF steps. No menu label matches the task goal. The third step (clear the check box labeled by Legend) is an LF step. The last step (click OK) is a highly over-learned action that closes a dialog box and terminates the action sequence.

Rodriguez (1997) and Soto (1997) found that the first NLF step in the hide legend task is the source of the difficulties that users have with this task. Almost all users required a hint to complete the first step. Franzke (1994; 1995) found a highly significant interaction for number of hints between number of targets (screen objects) for possible actions on the screen and LF versus NLF steps. There are many targets for possible actions on the first step of any task. Thus, we would expect first steps to be especially problematic. Once users are given the hint "pull-down the **Options** menu" in the hide legend task, there are only 7 menu items on that menu.

We have used two versions of the hide legend task in the simulation described in the following sections. The first version was a simulation of performing the hide legend task using Cricket Graph III, Version 1.5.3 described above. We will refer to this as the NLF scenario. The other version of the simulated task used a hypothetical version of Cricket Graph III that added a **Legend** menu to the menu bar. The items on this menu were **Show**, **Hide**, **Move**, and **Arrange**. This version of the hide legend task requires two steps (select **Hide** from the **Legend** menu) using this hypothetical interface. We will refer to this simulation as the LF scenario. Our discussion will focus on recall of the first step for each of the two versions.

## SIMULATION

A Mathematica program was developed implementing processes incorporated in LICAI+ and simulating responses from Cricket Graph III for correct actions in the hide legend task. Training was simulated by assuming that each step was performed correctly with hints given for the first NLF step. The following processes are simulated for the training: the comprehension process that generates goals and comprehends hints, storage in episodic memory during comprehension, retrieval of goals from episodic memory, and action planning, encoding of successful actions, and storage in episodic memory.

Representations of the task instructions, hints, and interface displays were coded and input to the simulation. The simulation also incorporated extensive knowledge about the basic Macintosh interface conventions for each screen object. For example, the **Options** menu item affords pull-down, and the **Options** menu item causes menu-selection, and so on. Other knowledge about actions, including moving and dragging the mouse pointer, and single- and double-clicking the mouse button, etc., was incorporated into the model.

### Simulation of Training

Training on each of the scenarios for the hide legend task was simulated in several encoding conditions as described below. At the end of training, episodic memory included nodes representing the task instructions, the hint (for the NLF scenario), the acted-on object and its label for each step, and the display generated by the application. The link strengths of nodes in episodic memory are proportional to the activation level of these nodes obtained in the encoding cycle.

*Encoding Bias*

In encoding cycles, we manipulated the relative strengths of the links between the rest of the network and the links between the network and the task and do-it goals. The motivation for such manipulations is a fundamental property of the action planning process. The action planning process will *not* work unless the links between the current task, or do-it goal, and the rest of the network

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

are much stronger than the rest of the links in the network. These strong links cause a goal to dominate the integration subprocess. This subprocess selects the object to be acted on and the action to be performed on each step of the task. Manipulating relative strengths of the links between the goal and the rest of the network enables us to explore the hypothesis that the goal may dominate *both* action planning and encoding processes.

Encoding processes have been simulated under three conditions. In task goal biased encoding condition (TG), we generated a network by multiplying by a factor of 4 the strengths of links from the task goal. The strengths of the links from the do-it goal were not changed. In Figure 1, three links from the task goal (hide legend) are strengthend by a factor of 4. In do-it goal biased encoding condition (DIG), the strengths of the links from the do-it goal were multiplied by a factor of 4, and those from the task goal remained unchanged. In the neutral encoding condition (N), no multiplication factor was applied. The NLF scenario was simulated using the TG, DIG, and N conditions. The LF scenario was simulated for the TG and N conditions since hints are not required and there is no do-it goal for the LF scenario.

### Simulation of Recall

The recall cues are the task instruction and the representation of task goals used in the action planning process in training trial, and the initial display for the first step. In each simulation, nodes in the episodic memory that match the representations of the cues were identified, and then the probability distribution of retrieving the recall targets were calculated. The recall targets were two encoding nodes for the LF scenario, and the do-it goal and four encoding nodes for the NLF scenario.

#### *Recall after LF training*

The probabilities of recalling the encoding of the first step for the LF scenario for TG and N bias conditions are given in Table 1. In the LF scenario, the encodings of the first and second steps are linked to the task goal. In the TG condition, the probabilities of recalling the encoding for each of the two steps was nearly equal since the task goal dominated the encoding process, reducing the influence of the application display. Thus, the model retrieved the representation of the first step a little more than 50% of the time. In the remainder, the model retrieved representation of the second step blocking the successful retrieval of the first step.

Correct performance of both steps is mediated by the same task goal, and the encodings are linked strongly to the common task goal in the TG condition. One implication of these results is that the encoding of a multi-step LF task will not reliably be retrieved by the combinations of task goal and display cues on each step. Thus, correct performance will depend on a mixture of successful recall and the label-following strategy. However, by lessening the biasing on the task goal in the N encoding condition, the display cues made a much stronger contribution to the encoding process and

**Table 1.** Probabilities of recalling the do-it goal or the encoding of first step for the LF and NLF scenarios. TG, N, and DIG stand for task goal biased, neutral, and do-it goal biased encoding condition, respectively.

| | *LF Scenario* | | *NLF Scenario* | | |
|---|---|---|---|---|---|
| | *TG* | *N* | *TG* | *N* | *DIG* |
| *Probability of recalling the do-it goal* | N/A | N/A | .027 | .253 | .618 |
| *Probability of recalling first step encoding* | .551 | .736 | .251 | .446 | .177 |
| *Total* | .551 | .736 | .278 | .698 | .795 |
| *Predicted Hints* | N/A | N/A | .722 | .302 | .205 |

significantly increased the probability of correctly recalling the encoding of each step.

#### *Recall after NLF training*

The probabilities of recalling the encoding for the first step and the do-it goal for the NLF scenario in the TG, DIG, and N bias conditions are given in Table 1. For the NLF scenario, the row labeled *Total* gives the probability of correctly performing the first step. LICAI+ cannot perform the first step without recalling the encoding or the do-it goal. The entries for Predicted Hints are, 1–*Total*.

Manipulation in the NLF scenario of the bias has a huge impact on recall performance. In the TG biasing condition, the probability of recalling the do-it goal is small. The task goal dominates the encoding process and the do-it goal has very weak, indirect links to the task goal. The task goal does have links to all four encodings of each step. The probabilities of recalling each step encoding are almost equal, .251, .227, .180, and .315, respectively.

In the N encoding condition, both the recall probabilities for the do-it goal and the first step encoding increased compared with the TG encoding condition. The reason is the same as the LF case. The display cues become more effective in recall process. Included in these cues is the label for the **Options** menu which is directly linked to the do-it goal. Thus, the initial display is a more effective retrieval cue for both the encoding of the first step and the do-it goal.

On the other hand, in the DIG condition, all links involving the concept Option are very strong. This enhances the effectiveness of the representation of the **Options** menu as a retrieval cue and strengthens the representation of the do-it goal in episodic memory, making it easier to retrieve.

## COMPARISONS WITH USER PERFORMANCE

Franzke (1994) and Soto (1997) have done studies relevant to evaluating LICAI+'s recall predictions. For NLF steps, the model predicts that users will require a hint to successfully perform the step if they fail to recall the correct step encoding or hint. We used the best available measure of recall, proportion of subject

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

**Table 2.** Proportion of times at least one hint was required for steps categorized by link type, training (exploration) and recall trial (short or long delay). From Franzke (1994).

| *Link Type* | *Training* | *Short Delay* | *Long Delay* |
|---|---|---|---|
| Exact Match | .07 | .00 | .14 |
| Synonym | .08 | .02 | .18 |
| Inference | .42 | .07 | .29 |
| No Link | .88 | .05 | .60 |

**Table 3.** Observed proportions of tasks requiring at least one hint as a function of task type and training and delay. From Soto (1997).

| | *Session 1* | | *Session 2* | |
|---|---|---|---|---|
| *Task Type* | *Training* | *Short Delay* | *Long Delay* | *Short Delay* |
| LF/C | .01 | .00 | .00 | .00 |
| LF/U | .19 | N/A | .12 | N/A |
| PL/C | .84 | .26 | .46 | .11 |
| PL/U | .58 | N/A | .29 | N/A |

requiring a hint on a task or step. However, this variable does not provide an unambiguous measure for evaluating the recall predictions for LF steps and tasks. Both successful recall and the label-following strategy can generate correct actions within 10 seconds.

For LF steps and tasks, LICAI+ predicts that no hints should be required during training or on recall trials. However, Rieman (1996) and Rieman, Young, and Howes (1996) found that users will explore an interface before taking the initial correct action predicted by the label-following strategy. This initial exploratory behavior can lead to long latencies and hints on LF steps that are outside the scope of LICAI+.

### Description of Available Experimental Data

We first present experimental data from Franzke (1994) and Soto (1997) focusing on the proportion of hints required on training and recall trials.

*Description of Franzke (1994)*

Franzke (1994) had four groups of 20 participants create a graph and then perform 9 editing tasks on the graph using one of four graphics applications, Cricket Graph I or III, or one of two versions of EXCEL. During training, participants did the task by exploration, receiving hints when necessary. Half the participants in each group were tested for retention after a 5 minute delay (short delay), and the remainder were tested after a 7 day delay (long delay).

Franzke classified each step in each task into one of four categories according to the relationship between the task goal for each step given in her instructions and the label of the object to be acted on for that step. Her exact match and synonym categories are examples of LF steps. In her third category an inference is required to link the correct object and the task goal. In the fourth category (no link) there is no meaningful link between the screen object and task goal. The latter two categories are both examples of NLF steps.

The results relevant to LICAI+ from Franzke's (1994) experiment are shown in Table 2. The table shows the proportion of times that at least one hint was required on a step, with the steps categorized by link type, training (exploration) and recall trial (short or long delay).

*Description of Soto (1997)*

Soto (1997) performed a study replicating and extending Franzke's results. Soto's 19 participants were trained on a series of 33 graph editing tasks using Cricket Graph III and were tested for retention after a 2 or a 7 day delay. All participants were experienced Macintosh users who had not used a graphing application. Editing tasks were carried out on three types of graphs: histograms, pie charts, and bar charts. The 11 histogram editing tasks and the first of the 11 bar and pie chart editing tasks were used as warm-up tasks, and these data are not included in the results described below.

Four out of the 10 experimental pie and bar chart editing tasks were unique (U) to that graph type and occurred once during training and testing. An example is "stand out a pie slice." Six of the tasks were common (C) to both graph types and occurred twice during training and recall sessions. An example is 'hide legend.' The delay between the two presentations of the common tasks averaged about 7 minutes. In Soto's data analysis, the second occurrence of a common task was treated as a recall trial with a short delay. His participants had no trouble recognizing the second occurrence even with a change in graph type.

Soto classified his editing tasks into three categories. Label-following (LF) tasks required acting on objects whose labels were semantically related to the goal. Thus, all steps in these tasks were equivalent to Franzke's direct match and synonym step types. Direct-manipulation (DM) tasks required acting on the task object (e.g. pie slice) mentioned in the task goal. These data are not discussed as it is beyond the scope of this version of LICAI+. Poorly-labeled (PL) tasks did not support either label-following or direct-manipulation violating the label-following strategy. Occasionally, a task supported label following as well as direct manipulation (e.g., 'Change the graph title to "Year of Production"'). For this reason, the tasks were classified based on the method used by the subject, rather than on a priori criteria.

Soto's analysis is by task rather than by the step level. The typical PL task has one or two initial NLF steps. Soto's findings and Franzke's (1994) results suggest that the initial NLF step has the largest impact on users' performance. Previously, we summarized Franzke's result showing that there is an interaction for the number of hints needed between LF versus NLF and the number of possible targets for action on a screen. The difficulty of

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

NLF steps increases dramatically as a function of the number of targets.

### Comparison With LICAI+'s Predictions

*Training Performance*

LICAI+ predicts perfect performance for both training and recall trials at all delays for LF steps. If we use the proportion of users requiring hints as our measure, a large majority of Franzke's (1994) results (shown in Table 2) and Soto's (1997) findings (shown in Table 3) support this prediction. The largest deviation that we know of is in the data from LF/U, Soto's condition where 19% of the participants required hints on the training trial.

The model makes equally strong training performance predictions for tasks and steps that do not support the label-following strategy (NLF tasks). LICAI+ predicts that these tasks and steps cannot be learned by exploration without hints or information looked up in a manual or help system. However, this prediction for NLF tasks is not sound. The observed proportions of tasks or steps requiring at least one hint ranges from less than .5 to .9 in different conditions of the Franzke and the Soto data.

However, the pattern of deviations in both the Franzke and the Soto data is instructive and supports the claim that the LF-NLF distinction is a useful design heuristic. LICAI+ makes incorrect predictions for learning by exploration in NLF tasks because of the model's simple exploration process. First, the model cannot perform exploratory activities like pulling down a menu to see if any items on that menu link to the tasks goal. Experienced Macintosh users carefully explore menus (Rieman, 1996) and act upon matching labels uncovered during such explorations.

Second, users seem to be able to use elimination strategies when dealing with a small number of screen objects like the items on a menu. For example, when participants are given the hint to pull down the **Options** menu in the hide legend task, they correctly select **Show Graph Items...** by a process of elimination. The other items on this menu are more specific and clearly have nothing to do with the hide legend task. LICAI+ can perform this step if it is given the knowledge that 'show is the opposite of hide' and that 'the legend is a graph item.'

The above arguments suggest that an interesting test of the model would be to consider NLF tasks in which the first two steps violate the label-following strategy. 'Hide legend' is such a task. Rodriguez (1997) shows that 100% of his subjects required hints to be able to perform this task. Franzke (1994) found that approximately 90% of the participants required hints for steps where there was no link between the task goal and the correct object's label.

*Recall at Short Delays for NLF Tasks*

LICAI+ predicts that successful performance on recall trials is possible only when users retrieve a hint or an encoding of a step from episodic memory. However, the model does not make predictions about the effects of delay. We have assumed that LICAI+'s recall predictions apply to delays of one or more days.

Franzke's (1994) and Soto's (1997) results show that immediate recall of NLF steps is quite good. Franzke (1994) found that about 90% of NLF steps can be recalled after a 5 minute delay (see Table 2). About 75% of Soto's PL tasks were performed correctly, without a hint, after a short delay (See Table 3).

*Recall at Long Delays for NLF Tasks and Steps*

LICAI+ predicts that successful recall performance can vary from .722, to .205 as a function of the encoding bias for NLF tasks and steps. Franzke's and Soto's results at long delays are hard to interpret because of the results from training trials for NLF tasks. Users' learning by exploration is better than that predicted by LICAI+. Thus, contrary to the predictions of the model, users will be able to discover the correct action on a recall trial even if they fail to recall a hint or encoding of the step.

We reanalyzed both Franzke's no link and inference steps at the long delay shown in Table 2 and Soto's recall data from his PL conditions shown in Table 3 at the long delay. We made the assumption that the probability of requiring hints on recall trials, $P_{require\_hint}$ , is just the probability of failing to recall a hint or step encoding, $P_{fail\_recall}$ , times the probability of failing to discover the correct action by exploration, $P_{fail\_exploration}$, assuming that the two events are independent. If we assume that $P_{fail\_exploration}$ estimated by the probability of requiring hints on the training trial, $P_{fail\_recall}$ can be estimated by $P_{fail\_recall} = P_{require\_hint}/P_{fail\_exploration}$.

The estimated values of $P_{fail\_recall}$ for Franzke's no link steps is .68, and .69 for the inference steps. These values are close to the predicted value for the TG condition shown in Table 1.

The estimated values of $P_{fail\_recall}$ for Soto's poorly labeled tasks at a long delay is .50 for the unique tasks and .55 for the common tasks. These results suggest that the task goal has a strong influence on the encoding process but that it is not as strong as the 4:1 bias assumed in computing the predictions for the TG conditions shown in Table 1.

## CONCLUSIONS AND IMPLICATIONS FOR PRACTICE

We have asserted that most users are occasional users of many applications, and they routinely use only a small fraction of the functionality of their frequently used applications. A model of routine cognitive skill is not a good description of users' actual patterns of use. The action sequences for occasionally performed tasks are generated by a mixture of recall of previous episodes of use and of problem solving processes that attempt to reconstruct missing action knowledge. Performance of these tasks is more like the reconstructive processes involved in recalling a story rather than the execution of a rule-based representation of a routine cognitive skill.

LICAI+ is a model of occasional users. This model suggests the partitioning of all steps executed in

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

performing a task into two categories: steps that support the label-following strategy and those that do not. Steps and tasks that support the label-following strategy can be performed by exploration. We know that users have strong preferences for learning by exploration (Carroll, 1990; Rieman, 1996), which the label-following strategy supports.

Experienced users can make effective use of manuals (Rieman, 1996) to perform tasks that are not supported by the label-following strategy. However, users will have continued trouble with steps not supported by label following (NLF steps). These steps once correctly performed with the assistance of hints are difficult to remember over long delays (2 or more days). We estimate that the probability of recall failure is at least .5.

The data from the short delay recall conditions also suggests a possible limitation of empirical usability tests. Test users will have trouble with the initial versions of common tasks that don't support the label-following strategy. Second and third versions of these tasks that are given to test-takers later in a session will be performed correctly, and evaluators may incorrectly infer that there are no problems with the interface for these later versions.

In summary, the theoretical and empirical results presented in this paper and in numerous other studies demonstrate the wide applicability of the label-following strategy. It supports rapid learning of all kinds of applications, not just walk-up-and-use applications like automated teller machines. We have shown in this paper that label following is also a major contributor to the usability of occasionally performed tasks.

## REFERENCES


Bewley, W.L., Roberts, T.L., Schroit, D., & Verplank, W.L. (1983). *Human Factors Testing in the Design of Xerox's 8010 'Star' Office Workstation: Case Study D: The Star, the Lisa, and the Macintosh.*

Carroll, J.M. (1990). *The Nuremberg funnel: Designing minimalist instruction for practical computer skills.* Cambridge, MA: MIT Press.

Franzke, M. (1994). *Exploration, acquisition, and retention of skill with display-based systems.* Unpublished Dissertation, University of Colorado, Boulder, Department of Psychology.

Franzke, M. (1995). Turning research into practice: Characteristics of display-based interaction. *Proceedings of human factors in computing systems CHI '95* (pp. 421–428). New York: ACM.

Kintsch, W. (1988). The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, **95**, 163–182.

Kintsch, W. (in press). *Comprehension: A paradigm for cognition.* Cambridge University Press.

Kintsch, W., & Welsch, D.M. (1991). The construction-integration model: A framework for studying memory for text. In W. E. Hockley and S. Lewan-dowsky (Eds.), *Relating theory and data: Essays on human memory* (pp. 367–385). Hillsdale, NJ: Erlbaum.

Kitajima, M., & Polson, P.G. (1995). A comprehension-based model of correct performance and errors in skilled, display-based human-computer interaction. *International Journal of Human-Computer Studies*, **43**, 65–99.

Kitajima, M., & Polson, P.G. (1996). A comprehension-based model of exploration. *Proceedings of human factors in computing systems CHI '96* (pp. 324–331). New York: ACM.

Kitajima, M., & Polson, P.G. (1997). A comprehension-based model of exploration. *Human-Computer Interaction*, **12**, 345–389.

Polson, P.G., & Lewis, C. (1990). Theory-based design for easily learned interfaces. *Human-Computer Interaction*, **5**, 191–220.

Rickard, T.C. (1997). Bending the power law: A CMPL theory of strategy shifts and the automatization of cognitive skills. *Journal of Experimental Psychology: General*, **126**, 288–311.

Rieman, J. (1996). A field study of exploratory learning strategies. *ACM Transactions on Computer-Human Interaction*, **3**, 189–218.

Rieman, J., Young, R.M., & Howes, A. (1996). A dual space model of iteratively deepening exploratory learning. *International Journal of Human-Computer Studies*, **44**, 743–775.

Rodriguez, M. (1997). A detailed analysis of exploratory behavior of new users of a graphics application. Institute of Cognitive Science Technical Report. University of Colorado, Boulder.

Ross, B. (1984). Remindings and their effects in learning a cognitive skill. *Cognitive Psychology*, **16**, 371–416.

Santhanam, R., & Wiedenbeck, S. (1993). Neither novice nor expert: the discretionary user of software. *International Journal of Man-Machine Studies*, **38**, 201–229.

Smith, D.C., Irby, C., Kimball, R., Verplank, W.L., & Harslem, E. (1982). *Designing the Star User Interface: Case Study D: The Star, the Lisa, and the Macintosh.*

Soto, R. (1997). Properties of Tasks that Determine Success in Learning By Exploration and Recall. Unpublished Master Theses.

Terwilliger, R.B., & Polson, P.G. (1997). Relationships between users' and interfaces' task representations. Proceedings of human factors in computing systems CHI '97. New York: ACM.

Wolfe, M., & Kintsch, W. (submitted). An overview of the construction-integration model.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

# Modeling Human Error within a Cognitive Theoretical Framework

**Daniela K. Busse and Chris W. Johnson**
Dept. of Computing Science, University of Glasgow
17, Lilybank Gardens, Glasgow G12 8RZ
Tel: +44 141 339 8855 ext. 2918
Email: [bussedljohnson]@dcs.gla.ac.uk

**ABSTRACT**
Current cognitive user models enable interface designers to describe, analyze and predict aspects of user cognition. However, none of the major cognitive user models such as ICS, MHP, or CCT tackle the human error aspect of cognition explicitly. The represented operator performance is constrained to be error-free, expert performance. This paper argues that usability and design analysis will greatly benefit from representing a cognition-based error model within a cognitive architecture, such as ICS. The Netscape Internet browser acts as a case study throughout. The resulting approach is shown to aid the analysis of human error. Reasoning about potential error causes as well as the generation of design recommendations can thus be grounded in cognitive theory.

**Keywords**

Human Error, Netscape, Cognitive User Modeling, ICS

## INTRODUCTION

### Integrating Error Models and Cognitive Architectures

Cognitive architectures seek to represent the building blocks of human cognition. They provide the basis for cognitive user models, which strive to represent some aspects of the user's understanding, knowledge, or cognitive processing. These models can then contribute to our understanding of the cognitive limitations of an operator performing a task, for example the effects of cognitive load on user performance (Barnard and May, 1993; Ashcraft, 1994).

Erroneous task performance highlights precisely these limitations of human cognition. It is surprising, therefore, that the major cognitive user models do not explicitly tackle issues associated with erroneous performance based on cognition. They strive to represent error-free performance, assuming expert performance in some perfect context (see for instance Simon, 1988; Grant and Mayes, 1991; Booth, 1991). This idealizes real-life conditions of task performance.

User error can point to problems in human-system interaction that need to be resolved in order to enhance the system's usability. Human error taxonomies aid the prediction and detection of error classes. They can thus be exploited for error prevention and recovery mechanisms (Reason, 1990; Taylor, 1988). Those can then be incorporated into the interface design.

On the other hand, stand-alone human error theories highlight possible sources of erroneous performance without providing a language in which to express these error tendencies when applied to human cognitive task performance. This paper will use a cognitive architecture as a vehicle for expressing not only expert task performance but also the more realistic error-prone thought and action sequences processed by the human operator. By doing this, the error modeling capability implicit in the comprehensive ICS cognitive architecture is made the focus of inquiry into the underlying cognition of user performance. Such explicit modeling of erroneous performance can thus help to communicate user cognition analyses, and to ground design decisions in a cognitive theoretical framework.

As a running example, error modeling will be applied to tasks concerning the use of Netscape Navigator™. This example is appropriate because it represents a mass-market application where errors frequently lead to high levels of frustration during common tasks (Johnson, C., 1997).

### Interacting Cognitive Subsystems (ICS) and Reason's Model of Human Error

We will use Interacting Cognitive Subsystems (ICS) (Barnard and May, 1993) to illustrate the modeling of human error within a cognitive architecture. ICS provides a comprehensive account of human cognition. It has

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

proved powerful in explaining cognitive phenomena such as the stability of users' mental models during dual task interference effects (Duke, et al. 1995). It has been applied to real-life systems and tasks, such as cinematography (May and Barnard, 1995). Alternative cognitive user models, such as Task Analysis for Knowledge based Descriptions (TAKD) (Johnson, P. et al., 1994), User Action Notation (UAN) (Hartson et al., 1990), or Soar (Newell, 1990) might have been used. However, they lack the level of detail in ICS's representation of cognitive processes, or, in the case of Soar, the inherent constraints these have to satisfy (Wilson et al., 1988; Kjaer-Hansen, 1995). ICS was designed to provide a theoretical framework within which to place user cognition. It attempts to "satisfy the need for applicable theory" (Barnard, 1987). ICS, therefore, bridges the gap between theory-oriented cognitive architectures and task-oriented cognitive user models (Grant and Mayes, 1991; Simon, 1988).

Reason's taxonomy of human error (Reason, 1990) represents a conceptual classification of error, as opposed to a contextual or a behavioural one. The latter, exemplified for instance by Hollnagel's (1991) classification of error phenotypes, does not lend itself to the in-depth analysis of the underlying cognitive sources of error. For instance, a behavioural error category might include errors that exhibit the same surface characteristics without sharing the same cognitive basis.

### An Interactive System: Netscape Navigator

According to user population estimates, the Internet is gaining roughly 150,000 new users per month, joining 20 million existing Internet users (Pitkow and Recker, 1994). Internet browsers facilitate global communication by providing supporting hypertext navigation. Familiarity with such browsers, and therefore their usability constitutes a prerequisite for taking part in this novel information exchange. Maximizing this usability therefore represents a continuous concern for designers of successively modified versions of Internet browsers. The Netscape Interface (see Figure 1) will be used for illustration throughout this paper.

### Content and Structure of this Paper

The following section will take a closer look at the ICS architecture and Reason's theory of human error. The modeling capacities of ICS will be illustrated by a representation of an error-free user performance. Reason's error classification scheme will then be introduced. Readers familiar with ICS and Reason can move straight to the third section, where the benefits of this combined modeling approach are pointed out. ICS is used as a framework within which Reason's classification of human error can be expressed.

## A COGNITIVE ARCHITECTURE AND A HUMAN ERROR MODEL

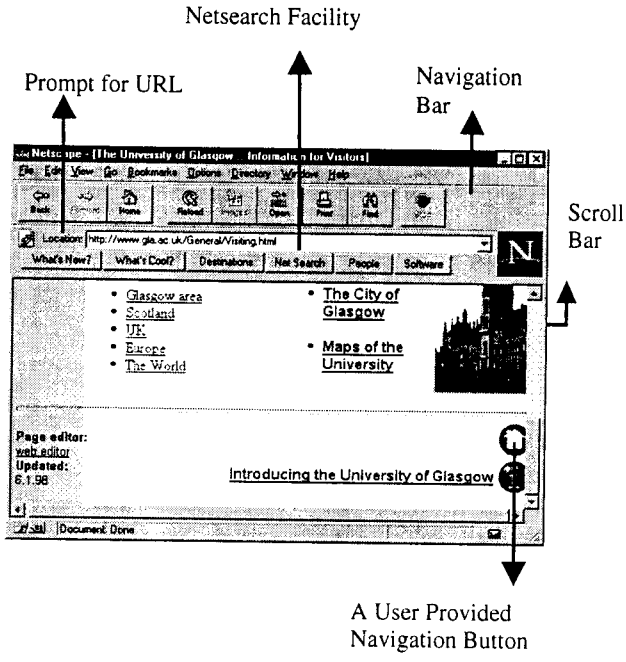



**Figure 1.** The Netscape Internet Browser

This section describes Barnard's ICS model and Reason's human error taxonomy. This provides the framework in which the representation of erroneous operator interaction can be placed.

### Interactive Cognitive Subsystems (ICS)

Cognition is represented in ICS as the flow of information between a number of different subsystems, and the processing performed on this data. Each of the subsystems has associated with it a unique mental code in which it represents the information it receives and processes. It will transform its data output into the corresponding mental code of the subsequently receiving subsystems. Each subsystem can receive several input streams and achieve a blending of these data streams under certain circumstances as described below (May and Barnard, 1995). Each subsystem also has at its disposal a local image store. This serves as an episodic memory buffer of infinite size. A copy of any input the subsystem receives will automatically be copied to the local image store, before being further processed.

The nine subsystems can be grouped into four categories. Figure 2 presents an overview.

#### *Modeling a Netscape Task in ICS*

Figure 3 illustrates how the error-free performance of a task of locating an object (an Up-Arrow, such as shown in the visual subsystem) is modeled in ICS in terms of

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

| | |
|---|---|
| Sensory subsystems: | |
| **VIS** | visual: hue, contour etc. from the eyes |
| **AC** | acoustic: pitch, rhythm etc. from the ears |
| **BS** | body-state: proprioceptive feedback |
| Effector subsystems: | |
| **ART** | articulatory: subvocal rehearsal & speech |
| **LIM** | limb: motion of limbs, eyes etc. |
| Structural subsystems: | |
| **OBJ** | object: mental imagery, shapes etc. |
| **MPL** | morphonolexical: words, lexical forms |
| Meaning subsystems: | |
| **PROP** | propositional: semantic relationships |
| **IMPLIC** | implicational: holistic meaning |

**Figure 2.** The Cognitive Subsystems

information flow between the subsystems, and thus the different resources that are employed. Visual information concerning the target arrives at the visual subsystem and is copied into the local store. It is then transformed into object code (1). The propositional subsystem has generated a representation of the target of the location task (by conferring with its local buffer) and transforms this into object code (2). This is sent to the object subsystem, and can there be blended with the incoming structurally encoded visual information (3). The matching representation can be sent back to the propositional subsystem – the target has been located.

Thus, Figure 3 illustrates how human mental processing underlying error-free performance can be represented within ICS. In the case of erroneous performance, however, usability designers might resort to an error classification scheme in order to analyse this particular instance of user behaviour. The following section will introduce one such taxonomy. We will then go on to show how a more detailed, cognitive analysis can be based on initial error classification, and thus provide a further perspective on user behaviour.

### Reason's Classification of Human Error

Reason (1990) investigated the more general underlying error production mechanisms within human cognition and produced a conceptual classification of error types which is widely referred to in research into error modeling (Green, 1985, Rasmussen, 1983; Rouse and Morris, 1987; De Keyser, 1989). He bases his error classification skill-based slips and lapses on the one hand, and rule- and knowledge- based mistakes on the other (see also Norman, 1981, and Rasmussen, 1983).

Reason furthermore asserts that instances of his three basic error types are indirect results of what he calls the 'underspecification' of cognitive operations. In case of an ambiguity of the situational requirements, the cognitive system defaults to contextually appropriate, high frequency responses. This idea of default assignments features in most other cognitive theories, such as Bartlett's (1932) theory of schemata, and is well backed up by empirical evidence.

This scenario particularly lends itself to being expressed in the 'cognitive language' provided by ICS. The limitations of human cognition in the face of information overload, or cognitive strain, is built into ICS as the architectural constraint of subsystems not being able to process simultaneously inputs which belong to distinct configurations. Using ICS might help expressing the details of Reason's 'underspecification' more precisely.

#### *Skill-based Slips and Lapses*

Slips and lapses are error types that these manifest themselves as actions or states that deviate from the current intention due to execution failures (slips) and/or storage

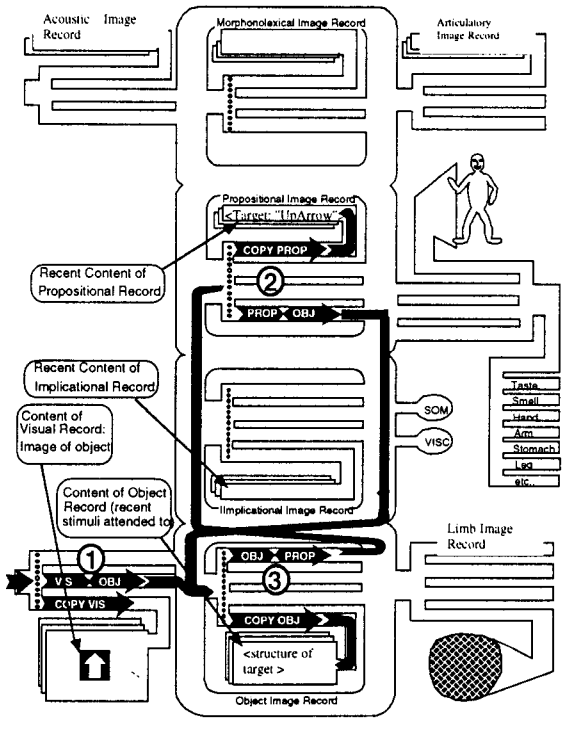



**Figure 3.** Processing associated with the task of locating an icon on Netscape

failures (lapses). Slips and lapses are observed at the skill-based level of performance, and originate from either the omission of attentional checks (inattention) during the routine action sequence or making an attentional check at an inappropriate moment (overattention).

A *slip* caused by inattention occurs in particular when current intention is to deviate from common practice. For instance, entering a well-known URL of a website constitutes a routine task. If the URL is changed and the user, although aware of that change, still happens to enter the old URL, then this is a typical example of an action slip.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

A *lapse* might arise from what Reason calls 'Reduced Intentionality'. For instance, if selecting a link on the current site results in a considerable delay for this site to be loaded, the users might become distracted, and then experience disorientation upon facing the loading site. This can be seen as one of Reason's described reduced intentionality states, such as a 'what-am-I-doing-here' experience (see below).

Skill based errors such as these contribute to the sources of user frustration when accessing the World Wide Web (as described in more detail in Johnson, C., 1997). These errors need to be taken into account in future design decisions. Applying Reason's categorization of error helps to identify error classes and presents a step towards dealing with the underlying usability problems of the system.

However, error taxonomies such as Reason's typically confine themselves to broad error categories such as slips and lapses. A more detailed, lower level description of such classes might aid the further investigation of its instances. Thus, the design process might be tuned more finely to the usability needs pointed to by the user error.

Cognitive modeling techniques such as ICS can provide a more precise vocabulary to augment the general descriptions of error taxonomies. Examples of this lower level modeling of classes of human error are given below.

*Rule-based Mistakes*

Mistakes are apparent in actions that may run according to plan, but where the plan is inadequate to achieve its desired outcome. For any task, rules must be selected by the cognitive system which describes methods to reach a given (sub)goal. The selection occurs according to certain criteria. These include best match, specificity, and rule strength. Rule strength is defined to be the number of times a rule has performed successfully in the past. Occasionally, rule strength might override the other factors resulting in misapplications of otherwise 'good' rules to inappropriate situations.

As an example, an animated icon at the bottom of a page, near the contact information is quite often the mail-me icon (commonly found are self-folding envelopes, self-writing letters, or moving mailboxes). A corresponding rule will be formed and strengthened over several successful applications. In the case of a home-page icon being animated and located at a similar position in the screen layout, this rule might be applied and could lead to non-intended actions such as clicking on the icon when intending to mail the author of the page.

Such error classes can be predicted as increasingly adding to usability deficiencies as the use of animated icons accelerates in web page design (Nielsen, 1997). By being able to predict these errors, preventative measures can be taken and further user frustration (Johnson, C., 1997; Ramsay et al., 1998) can be curbed.

## USING ICS TO EXPRESS REASON'S ERROR TYPES

In this section, we will examine more closely the modeling of errors as identified by Reason's taxonomy within the ICS architecture.

Commonly occurring errors and usability problems when interacting with Internet browsers' interfaces gave rise to numerous design guidelines and principles[1]. Interface design issues such as the use of counter-intuitive icons and download delays are all well known to aggravate usability problems (see for instance Nielsen, 1996; Johnson, C., 1997; Ramsay et al., 1998). Rarely, however, are the errors resulting from those usability problems described in detail, or even analyzed in terms of underlying psychological factors (Johnson, C., 1998). Expressing such errors within a cognitive model will allow us to investigate and reason about their underlying psychological causes. The model is thus used as a tool for reasoning about user error on a further, more detailed level.

### Analysis of Errors and their Underlying Cognition

High download latency of web pages was identified as major source of frustration and decreased satisfaction with the downloading site and also as attenuating user performance (Ramsay, Barabesi and Preece, 1998; Johnson, C., 1997). For instance, as introduced above, if selecting a link on the current site results in a considerable delay for this site to be loaded, the users might become distracted, and then experience disorientation upon facing the loading site.

This disorientation can be classed as the effect of a phenomenon which Reason termed 'Reduced Intentionality'. If a delay occurs between the formulation of an intention to do something and the time for this activity to be executed, the intention needs to be periodically refreshed. Other cognitive processes such as secondary intentions will otherwise claim the workspace resources. This mechanism can lead to lapses in the form of reduced intentionality states, the above described surprise and disorientation.

The cognitive processes underlying this scenario can be represented in ICS as shown in Figure 4.

[1] See for instance Yale C/AIM WWW Style Manual (URL: "http://info.med.yale.edu/caim/manual/index.html" current at 08.12.1997) or The Ten Commandments of HTML (URL:"http://www.visdesigns.com/design/commandments.html" current at 08.12.1997

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

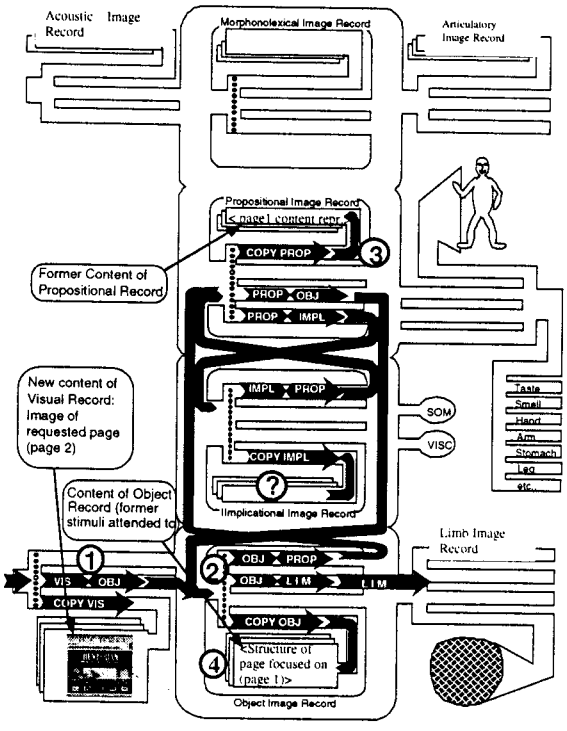



**Figure 4.** Reduced Intentionality: A Lapse

After processing the goal hierarchy for selecting a link, the cognitive system shifts its focus back onto the current page (3 and 4). If novel external (1) and the current internal input are not coherent, and thus cannot be blended (2), a decision must be made as to which of those to accept as valid input. The longer the delay, the stronger the influence of the novel input grows, with it eventually replacing the internal propositionally influenced representation (3). The recognition of this mismatch will lead to a lapse as described above.

By modeling the underlying mechanisms of manifestations of attenuated performance, such as user error, and the causes of decreased satisfaction within ICS we can shed some light on the processes fundamental to the production of the user error as mediated by the described usability problems.

### Reasoning about Alternative Analyses of Error Causes

Misinterpreting user interface icons is a common source for user error in interactive systems (Norman, 1988, 1993). However, the mistake might be grounded in varying cognitive processes, and not stem from one kind of cognitive mechanism alone.

Typically, user interface design manuals and textbooks stress the importance of intuitiveness of the icons chosen (Preece, 1994) and thus identify 'counter-intuitiveness' as a source of faulty identification of icons. However, further insight into the source of such user error can be obtained by investigating it in greater detail. As will be shown below, mistaking for instance a mail-me button with a homepage icon can be modeled in respect to two differing underlying cognitive mechanisms.

Unless these two different causes are considered these designs might misdiagnose an important problem in user utilization of icons. Using a cognitive architecture to reason about the potential underlying cognitive error production processes allows designers to investigate the detected usability problem in a systematic way.

The above described user error could according to Reason's scheme be classified as a slip termed 'Perceptual Confusion'. In perceptual confusion, something that looks like the proper object, is in the expected location, or does a similar job is accepted as a match for the proper object. These slips could arise because, in a routine set of actions, it is unnecessary to invest the same amount of attention in the matching process. Thus acceptance criteria concerning the expected input might degrade, and result in rough and ready matches.

The processing carried out can be modeled in ICS as shown in Figure 5.

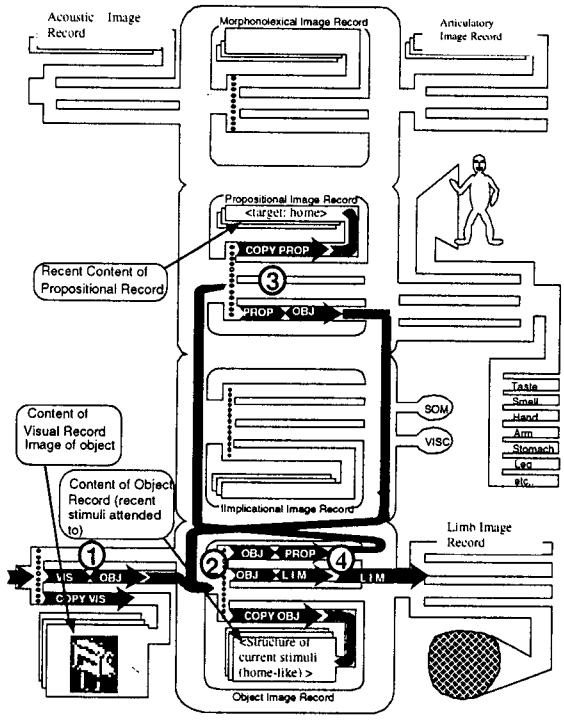



**Figure 5.** Perceptual Confusion: A Slip

The visual data is received at the visual subsystem (1), sent to the object subsystem for the recovery of a structural description (2), and finally interpreted by the propositional subsystem (3). A loop is entered in order to maintain a stable cognition. The resulting interpretation

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

on the propositional level influences the further view of the object. If, however, the object subsystem receives ambiguous visual information, it will make use of its local image record and fill in the assumed missing information. This principle of ICS resembles closely what Reason describes as the cognitive system's reaction to underspecification of a mental operation as described above.

The data thus acquired from the image record of the object subsystem might also fit in with the propositional interpretation of what is perceived, and thus stabilize in the cognitive system. If the assumption underlying the choice of what data is used to eliminate the underspecification is wrong, however, the representation of what is thought to be perceived will also be incorrect. The wrong icon will be chosen, and the information necessary for a mouse click sent to limb subsystem (4).

This represents one possible underlying cause of the described error. However, the same manifestation of user behaviour might also point towards a second, different underlying cognitive mechanism. Employing Reason's taxonomy, the mistaking of an icon can be classed as a perceptual slip as modeled above. On the other hand, it could also be classed as a rule based mistake. Using ICS to model the underlying cognition of the error provides a means to further investigate the behaviour trace and its associated usability problem.

Thus, the error described above could be classed as a rule-based mistake as opposed to a slip. Identifying the home-icon might well be based on rules that are utilized by the cognitive system in order to discriminate different sets of icons. Features which positively discriminate icons fulfilling one function from those fulfilling another might be listed in the set of conditions which when matched cause to fire the rule. Indiscriminative features in icons might thus lead to a rule wrongly being fired.

This can be modeled in ICS (see Figure 6) similar to the modeling approach applied to the perceptual confusion approach, but this time with the implicational subsystem playing the major role in accepting information augmented wrongly by the propositional subsystem and its local image store. Thus for the goal 'press home button', a subgoal hierarchy can be formulated as 'if locate home button, move cursor to click on it', and 'if object has X features, it is the home button'. By mistaking the icons on a propositional level, the mail-me button might be clicked instead.

The examples elaborated above show clearly how one overt form of user error can stem from several different 'errors' within the cognitive processing taking place. This M:N relationship between cause and error might have gone undetected if systematic error modeling within a cognitive architecture had not taken place, this helps analysts to explicitly consider the detailed causes of usability problems.

### Generating Design Recommendations

Since underspecification proved the major source of error in the above example, once for perceptually and then for semantically discriminative features of the icon, this should be targeted by designers to remedy misidentification of icons. Thus, two functionally dissociated sets of icons should not share the same superficial perceptual features.

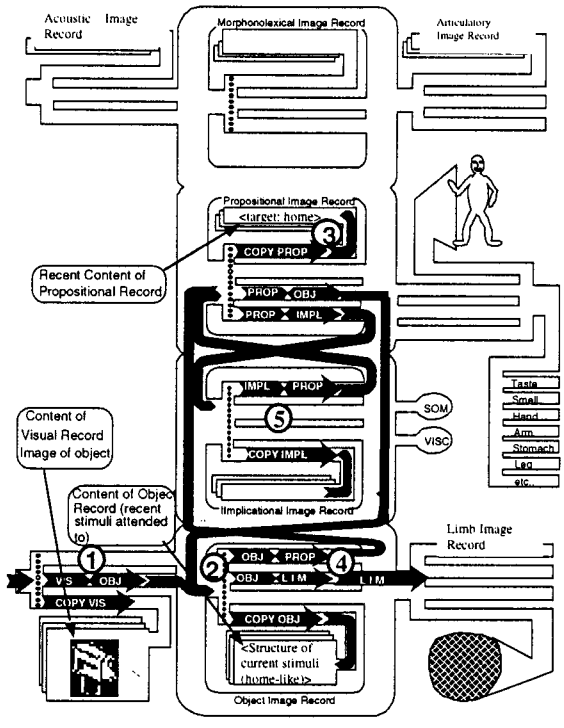



**Figure 6. Rule Strength: A Mistake**

Features commonly used to discriminate one set of icons from another should be taken into account when designing future sets (Moyes, 1995). These feature considerations should not limit themselves to ambiguity concerning structural characteristics of icons, but also to features such as those mentioned in the examples earlier. This included as discriminative features of mail-me buttons not only their shape and internal composition, but also for instance the location of the icon on the screen, and characteristics commonly unique to mail-me buttons such as animation as present in self-folding envelopes, self-writing letters, or moving mailboxes.

The important point to highlight here is that the modeling approach described does present a method for providing a grounded rationale for design decisions, and can guide the designer in making informed choices when faced with design alternatives.

Another example of how this modeling technique can aid

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

the generation of design decisions is introduced as this section progresses.

Johnson (C., 1997) describes how download latency of web pages affects the usability of the World Wide Web. The effects range from user dissatisfaction with time investment to the psychological devaluation of the anticipated page (Ramsay et al., 1998). Consider the following scenario of user error resulting from download latency: After having selecting a link on the current site, a delay in downloading might lead to attention being focused on reading the current page. An intention to scroll down the page just before the new page is downloaded might lead to the scrolling action being carried out on the new page instead.

This scenario fits Reason's description of 'behavioural spoonerisms', namely slips based on interference errors. As defined above, a slip is an action that deviates from intention due to failure in the execution stage of processing operations. An interference error occurs, when two concurrent actions compete for control over cognitive processing and a transposition of actions within the same sequence takes place. For instance, intending to speak and perform an action at the same time can lead to inappropriate blends of speech and action. In our example, waiting for the new page to load, and scrolling the old page can be seen as two concurrent actions interfering and leading to an execution failure, the scrolling of the new page.

This can be modeled in ICS very similarly to the skill-based example of reduced intentionality. Only this time the focus is not on the delay but on the shift of focus back to the current page. A 'mental model' of the current page will be constructed (or reactivated). The unexpected appearance of the new page might lead to a blending of representation and the action included in one cognitive configuration carried out as part of a secondary one.

As a consequence, future browser designers should beware of the error-inducing character of non-interrupted browser functionality when downloading a site. Alternatively, browser functionality should only be available to the current site accessed. A clear distinction should be made when transferring functionality to the downloading site to alert users to the new context. This design flaw in Internet Browsers has not received much attention. We hypothesize that it may become increasingly important as the interweaving of the user population of the Internet grows and the World Wide Web becomes an increasingly common tool for communication and information exchange. Detailed, error-oriented cognitive analysis of such design problems can help to predict future generations of interface problems.

## CONCLUSION AND FURTHER WORK

Cognitive user modeling enables engineers to gain a deeper understanding of the complexities of human task performance. Current techniques typically constrain this performance to be idealized, error-free and often at an expert level. However, human error during performance represents a major source of insights into the workings and limitations of operator cognition, and therefore into usability problems. By being based on cognitive models, the possibility of representing erroneous performance is inherent in these techniques. Few modeling techniques to date explicitly represent human error precisely, as embedded in cognitive theory. This paper showed the adoption of Reason's error taxonomy and Barnard's ICS for the systematic representation of operator error within a theoretical cognitive framework. The utilization of such a combined approach was illustrated to benefit several areas of application. User error can be described more precisely by linking it to its underlying cognition. Analysis can reach beyond surface categorization, and it is made possible to reason about the actual causes of error. As a consequence, an informed choice concerning competing design options is facilitated. This paves the way for usability design that takes full advantage of the insights expressed in cognitive theory.

Embedding human error modeling into a cognitive theoretical framework helps to express designers' understanding of the error sources. Communication of their reasoning, based on expertise and experience, is illustrated in this paper by using Reason's taxonomy and ICS. Further work might also take issues such as 'learn-ability' and level of complexity into account in the choice of the cognitive architecture employed. More easily learnable cognitive modeling techniques will further lend themselves for integration into the design process.

## ACKNOWLEDGEMENTS

This work was supported by UK EPSRC Grant No GR/L27800.

## REFERENCES

Ashcraft, M.H., *Human Memory and Cognition*, 2nd ed., Harper Collins College Publishers, 1994

Barnard, P.J., Cognitive Resources and the Learning of Human-Computer Dialogs. In: Carroll, J.M., (ed.) *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, ch.6. MIT Press, Cambridge, MA, 1987

Barnard, P.J., May, J., Cognitive Modeling for User Requirements. In Byerley, P.F., Barnard, P.J., & May, J. (eds) *Computers, Communication and Usability* (North Holland Series in Telecommunication) Elsevier: Amsterdam (1993).

Bartlett, F.C., *Remembering: A Study in Experimental and Social Psychology*. Cambridge: Cambridge University

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

Press, 1932

Booth, P.A., Modeling the User: User-System Errors and Predictive Grammars. In Weir G.R.S. and Alty, J.L. (eds.) *Human-Computer Interaction and Complex Systems*, Academic Press Ltd., 1991

De Keyser, V., Human Error, *Recherche*, 1989, Vol. 20, No. 216, p.1444

Duke, D.J., Barnard, P.J., Duce, D.A., and May, J., *Syndetic Modeling*, Amodeus-2 Technical Report ID/WP49, 1995

Grant, A.S. and Mayes,J.T., Cognitive Task Analysis? In Weir G.R.S. and Alty, J.L. (eds.) *Human-Computer Interaction and Complex Systems*, Academic Press Ltd., 1991

Green, R.G., Stress and Accidents, *Aviation Space and Environmental Medicine* 1985, Vol. 56, No. 7, pp.638 - 641

Hartson, H.R., Siochi, A.C., Hix, D. The UAN: A User-Oriented Representation for Direct Manipulation. *ACM Transactions on Information Systems*, 8(3), pp. 181-203, July 1990

Hollnagel, E., The Phenotype of Erroneous Actions: Implications for HCI Design. In: Weir, G.R.S. and Alty, J.L., (Eds.), *Human-Computer Interaction and Complex Systems*, Academic Press, 1991

Johnson, P., Diaper, D., and Long, J.B. Tasks, Skills, and Knowledge: Task Analysis for Knowledge based Descriptions. In: B. Shackel (ed.) *Human-Computer Interaction - INTERACT 1994*. Amsterdam: North Holland.

Johnson, C.W. Electronic Gridlock, Information Saturation and the Unpredictability of Information Retrieval Over the World Wide Web. In: Palanque, P. and Paterno, F. *Formal Methods in Human Computer Interaction: Comparison, Benefits, Open Questions.* Springer Verlag, Berlin, 1997

Johnson, C.W. *Why CHI (Computer-Human Interaction) Has Failed to Improve the Web*, at URL: "http://www.dcs.gla.ac.uk/~johnson/papers/web98.html" current at 02.02.1998

Kjaer-Hansen, J., Unitary Theories of Cognitive Architectures. In: Hoc, J., Cacciabue, P., Hollnagel, E., (Eds.), *Expertise and Technology: Cognition and Human-Computer Interaction*. LEA, Inc, 1995

May, J. and Barnard, P. (1995) Cinematography and Interface Design. In Nordby, K., Helmersen, P.H., Gilmore, D.J. and Arnesen, S.A. (eds.) *Human-Computer Interaction: Interact '95*. Chapman and Hall: London pp.26-31

Moyes, J., *Putting Icons in Context: The Influence of Contextual Information on The Usability of Icons*, PhD Thesis, Glasgow University, Computing Science Dept, 1995

Newell, A., *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press, 1990

Newell, A. and Simon, H.A., *Human Problem Solving*. Englewood Cliffs, NJ: Prentic-Hall, 1972

Nielsen, J. Alert Box for May 1996: *Top Ten Mistakes in Web Design* at URL: "http://www.useit. com/alertbox /9605.html" current 08.12.1997

Norman, D.A. Categorization of action slips. *Psychological Review*, 1981, 88, 1-15.

Norman, D.A., *The Psychology of Everyday Things*. New York: Basic Books. 1988

Norman, D.A., *Things that make us smart*, Reading, MA: Addison-Wesley, 1993

Pitkow, J.E. and Recker, M.M., *Results from the first World-Wide Web User Survey*, 1994, at URL: " http://www.gvu.gatech.edu/user_surveys/survey-01-1994/survey-paper.html" current 02.12.1997

Preece, J., *Human-Computer Interaction*, Addison-Wesley, 1994

Ramsay, J., Barabesi, A., and Preece, J., A Psychological Investigation of Long Retrieval Times on the World Wide Web. In: *Interacting with Computers*. Special edition on CHI and Information Retrieval, 1998, in press

Rasmussen, J. Skills, rules, knowledge: signals, signs and symbols and other distinctions in human performance models. *IEEE Transactions: Systems, Man and Cybernetics*, 1983, SMC-13, 257-267.

Reason, J., *Human Error*, Cambridge University Press, 1990

Rouse, W.B., Morris, N.M., Conceptual Design of a Human Error Tolerant Interface For Complex Engineering Systems, *Automatica*, 1987, Vol. 23, No. 2, pp. 231-235

Simon, T., Analysing the Scope of Cognitive Models in HCI: A Trade-Off Approach. In: Jones, D.M. and Winder, R. (eds.) *People and Computers IV, Proceedings of the Fourth Conference of the British Computer Society*. Cambridge University Press, 1988

Taylor, J.R., Using Cognitive Models to make plants Safer: Experimental and Practical Studies. In: Goodstein, L.P., Andersen, H.B., and Olsen, S.E. (eds.) *Tasks, Errors and Mental Models*, Taylor and Francis, London, 1988

Wilson, M.D., Barnard, P.J., Green, T.R, and Maclean, A. Knowledge-Based Task Analysis for Human-Computer Systems. In: Van Der Veer, G., Green, T.R., Hoc, J., and Murray, D.M. (eds.) *Working with Computers: Theory versus Outcome* Academic Press, 1988

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

# Automatic, action driven classification of user problem solving strategies by statistical and analytical methods: a comparative study

**M. Fjeld, S. Schluep & M. Rauterberg**
Institute of Hygiene and Applied Physiology (IHA)
Swiss Federal Institute of Technology (ETH)
Clausiusstr. 25, CH-8092 Zurich
{fjeld, schluep, rauterberg}@iha.bepr.ethz.ch
www.iha.bepr.ethz.ch/pages/forschung/mmi/mmi.htm

**ABSTRACT**
We have recorded the behaviour of several users solving the same tasks with an interactive database program and were able to identify several distinct strategies. Since the number of users exceeds the number of strategies, multiple users will have a strategy in common. Our aim was to find groups of users sharing the same strategy. Following each of the three methods (correlation, intersection, and exclusion) we define a metric among task solving sequences. For multiple users, we represent these measures by a matrix system, in order to find groups of users with common behaviour. Direct interpretation or multi dimensional scaling of such matrices indicates distinct user groups. The common denominator for each group can be interpreted as a strategy. A few distinctive solution strategies were found to exist.

**Keywords**
Mental models, observable behaviour, plan recognition, user strategies, statistical analysis, repetitive behaviour

## 1 MODELLING APPROACH

Humans express themselves in many ways. One of these ways is everyday problem solving. We will focus on problem solving in the domain of human computer interaction. In particular, we will examine how multiple users solve various tasks with a relational database application.

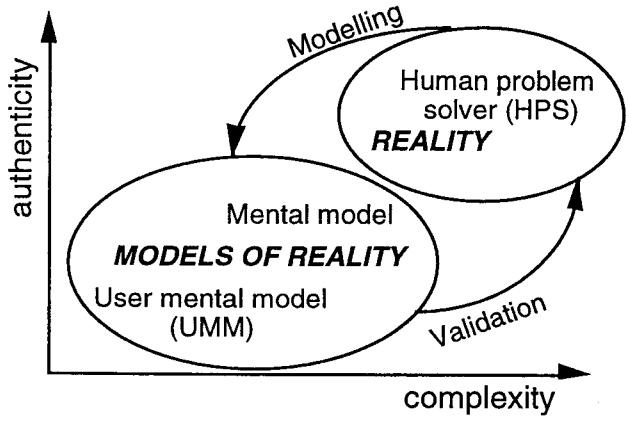



Fig. 1: A scheme showing the differences between models of reality and real humans (HPSs). Models are meant to represent objects and processes existing in reality.

It is hard to grasp how human problem solvers (HPS) really express themselves, since the persons we study are live beings. Nevertheless, a mental model (see Fig. 1) may give us an idea of the real HPS. Since we are interested in computer mediated, everyday task solving, we introduce a special case of mental models, called user mental model (Tauber, 1985) (UMM; see Fig. 1). UMMs can bring understanding about the strategies people use when solving specific problems. UMMs can be represented in many ways, using plain text, Petri nets or state-transition vectors. We choose the latter representation to elaborate UMMs based on observable task solving behaviour.

In general, we observe a lot of task solving behaviour that is not strictly *task related.* If we study one user solving a task, we are hardly able to single out the successful *strategy* from the *remaining behaviour.* One approach may be to study many users solving the same task. Since they all solve the same problem, we suppose that their common behaviour is what was required to solve the task. If there are several successful strategies, some users may have one strategy in common, other users a second one.

Successful strategies are most often defined by the given task-system combination. For users to accomplish a task, they must follow one of these strategies. As soon as a successful strategy has been accomplished, user behaviour is finished.

Which strategy a user prefers, as well as other kinds of user behaviour can tell us something about the particular HPS; for instance how the successful strategy was acquired. Given a behavioural task solving sequence, we want to separate the *strategy* (which is more related to the task-system combination) from the *remaining behaviour* (which is more related to the HPS). In the rest of this paper, *strategy* will mean one (of many), possibly error free, task solving behavioural sequences.

The aim of our work is to find which strategies are needed to solve a given task. We are looking for automatic methods to find these strategies. Under certain conditions, strategies may also be obtained by protocol analysis (Ericsson and Simon, 1984). Protocol analysis implies manual inspection of video and verbal utterances in addition to logfiles. With simple tasks, this work can be overcome. For more complex tasks, protocol analysis becomes cumbersome. Semi-automatic generation of

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

process models was studied by Ritter and Larkin (1994). Motivated by their work, we wish to suggest further principles for automatic recognition of user strategies and plans.

In this paper, human perception and verbalisation will not be considered as part of the problem solving. Hence, purely based on observable task solving behaviour, we set out for automatic methods, applicable with simple as well as with complex tasks. We only consider protocol analysis as a mean to validate the automatic methods we elaborate.

## 2 SYSTEM DESCRIPTION

The system we study is a relational database program with 153 different dialogue states. The possible transitions of the system are represented by a state-transition vector space. A state-transition-vector (STV) summarises a subject's task solving behaviour for one task. It has length n, where n is the total number of transitions (n=978) for the complete database program. Each STV element tells how many times a certain transition was activated to solve the task.

Since the order of activated transitions is not contained in the STV, the order of user behaviour is only partly conserved. It is stored in an implicit form, given by the system dialogue structure and is embedded in the structure of the STV.

To reduce complexity, it is possible to replace each STV element >1, by 1. We call the result binary-state-transition-vector (B-STV). It tells us which transitions were activated, but nothing about repetition.

## 3 TASK DOMAIN

An empirical investigation was carried out to compare different types of expertise (Rauterberg, 1992). For the reconstruction of UMMs we used logfiles of six novice and six expert users, all solving the same task. The task was to find out how many data records there are in a given database consisting of three file. An example UMM of a task solving process, based on one of the experts, is presented in Rauterberg et al. (1997). In that example, 15 *different* transitions (number of positive STV elements) were activated to solve the task. However, since some of them were activated repeatedly, the *total* number of activated transitions (the sum of STV elements) is 25.

## 4 INTERPRETING BEHAVIOURAL SEQUENCES

Studying an STV of one user can tell us which system states the user passed by, which transitions that were triggered in those states and how many times that happened. Different users working with the same system are directly comparable, since their behavioural sequences only differ by the value of the vector elements.

## 5 BASIC QUESTIONS AND METHODOLOGY

First, we want to find out how the behavioural sequences of two users can be related. A classical method is that of correlation. An alternative is to look for analytical methods. The user STVs can be represented by ellipses as in Fig. 2. The area of an ellipse corresponds to the sum of the STV element values. Intersection area can be understood as symmetric similarity between two user STVs. Exclusion areas can be understood as the asymmetric difference between two user STVs.

Based on such considerations, we raise the following questions and suggest corresponding methods as answer:

1) What is the proximity between two behavioural sequences? Method suggested: *correlation.*

2) What do two behavioural sequences have in common (similarity)? Method suggested: *intersection* (Fig. 2).

3) What do two behavioural sequences not have in common (difference)? Method suggested: *exclusion* (Fig. 2).

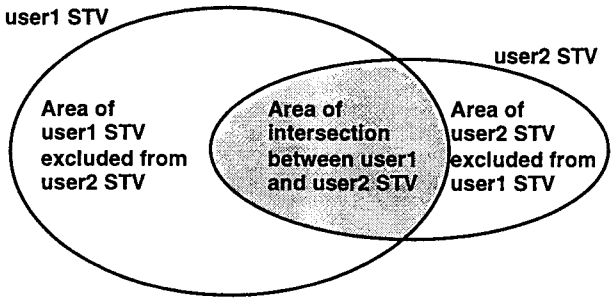



Fig. 2: Intersection area and exclusion areas between user1 and user2 STV.

For each method, we elaborate a metric (Table 1). The order of the metric may be symmetrical (the metric applied from user1 STV to user2 STV is the same as the metric applied from user2 STV to user1 STV) or asymmetric (the metric applied from user1 STV to user2 STV is *not* the same as the metric applied from user2 STV to user1 STV). Based on the metrics applied between all the user STVs, we then apply a grouping algorithm.

With each group suggested by the grouping algorithm, a strategy may be approximated. The procedure is to create a STV with a maximum number of non-zero elements common to all the users of the group.

In the following presentation, we will proceed from more statistically based to more analytically based methods.

Table 1: The three suggested methods and their characteristics. CORR means a standard correlation method, the other metrics are defined by Formula 1,2 and 3.

| Method | Metric name | Metric nature | Grouping algorithm |
|---|---|---|---|
| Correlation | CORR | Statistical | Statistical |
| Intersection | $M^{IS}_{p,q}$, $M^{BIS}_{p,q}$ | Analytical | Statistical |
| Exclusion | $M^{EX}_{p,q}$ | Analytical | Analytical |

### 5.1 CORRELATION METHOD

In this method the metric between user STVs is the degree of proximity. The metric values are analysed by multi-dimensional-scaling (MDS, Systat, 1989) to indicate groups of users.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

#### 5.1.1 METRIC

Correlation is one way to measure the proximity between behavioural sequences. We apply Pearson correlation as a measure for proximity between two STVs. By this procedure, we get an mxm (m=12) diagonal dominant symmetrical matrix with possible values between minus one, via zero (no proximity) and one (equality). For Fig. 3 the observed values are between -0.003 and 0.948 (without considering the diagonal elements).

#### 5.1.2 GROUPING ALGORITHM

The correlation matrix is interpret by MDS, giving the plot of Fig. 3. We have chosen to apply two dimensional MDS to allow visual interpretation of the plots.

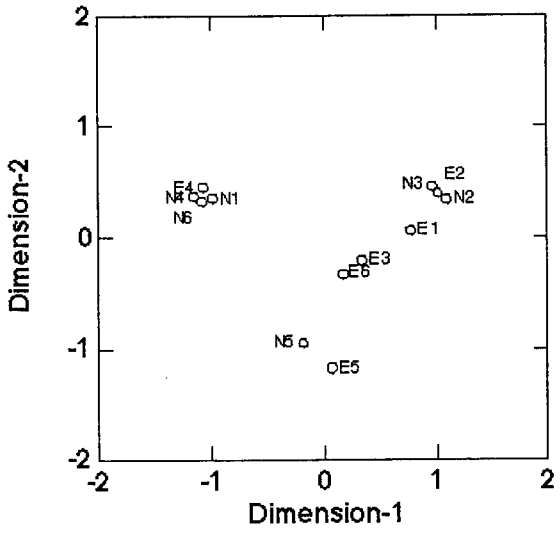



Fig. 3: MDS (r=1, Kruskal, Mono) plot with a Pearson correlation matrix gives RSQ=0.870.

#### 5.1.3 OUTCOME

From the plot in Fig. 3 we see how the users may be grouped: {N1, N4, N6, E4}, {N2, N3, E1, E2, E3, E6} and {N5, E5}. Some of these user STVs may well consists of parts of several strategies in addition to the successful one.

According to the proportion of variance (RSQ=0.870), MDS explains some of the variance of the user data, but a significant part remains unexplained.

### 5.2 INTERSECTION METHOD

This method is based on the observation that if two users followed the same strategy, that strategy will belong to the intersection of the two users STVs. The order of the an intersection metric is symmetric, since both user STVs have the same in common. These metric values are analysed by MDS to indicate groups of users.

#### 5.2.1 METRIC

Similar behaviour is measured by summing up the smaller STV elements of the two user STVs, thus considering the number of activated transitions common to both users.

It is reasonable to *normalise* the degree of intersection by the smaller of the sums of the STVs elements (which would be the maximum possible value for the intersection).

Formula 1:

$$M^{IS}_{p,q} = \frac{\sum_{i=1}^{n} \min\left(e_{p,i}, e_{q,i}\right)}{\min\left(\sum_{i=1}^{n} e_{p,i}, \sum_{i=1}^{n} e_{p,i}\right)}$$

where:

| | |
|---|---|
| $M^{IS}_{p,q}$ | : Intersection metric between user p and q |
| $i$ | : Summing Index STV elements |
| $n$ | : STV length, upper summing limit |
| $e_{p,i}$ | : STV element i for user p |
| $e_{q,i}$ | : STV element i for user q |

We may ignore repetitive behaviour, using B-STVs instead of STV. Results based on B-STVs are called *binary*.

Formula 2:

$$M^{BIS}_{p,q} = \frac{\sum_{i=1}^{n} \min\left(e_{p,i} \cdot e_{q,i}, 1\right)}{\min\left(\sum_{i=1}^{n} \min\left(e_{p,i}, 1\right), \sum_{i=1}^{n} \min\left(e_{p,i}, 1\right)\right)}$$

where:

| | |
|---|---|
| $M^{BIS}_{p,q}$ | : Binary intersection metric between user p and q |
| $i$ | : Summing Index B-STV elements |
| $n$ | : B-STV length, upper summing limit |
| $e_{p,i}$ | : B-STV element i for user p |
| $e_{q,i}$ | : B-STV element i for user q |

By this procedure, we get an mxm (m=12) symmetrical matrix with elements based on STVs (Formula 1) or B-STVs (Formula 2). The elements take possible values between zero (no similarity) and one (equality). For Fig. 4, based on STVs, the observed values are between 0.078 and 0.929 (without considering the diagonal elements). For Fig. 5, based on B-STVs, the observed values are between 0.182 and 0.882 (without considering the diagonal elements).

#### 5.2.2 GROUPING ALGORITHM

We interpret the symmetrical exclusion matrix by MDS, obtaining plots like Figs. 4 and 5. The users seem to represent three groups, {N1, N4, N5, N6, E4}, {N2, N3, E1, E2, E5} and {E3, E6}. E3 and E6 may as well be combinations of several strategies.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

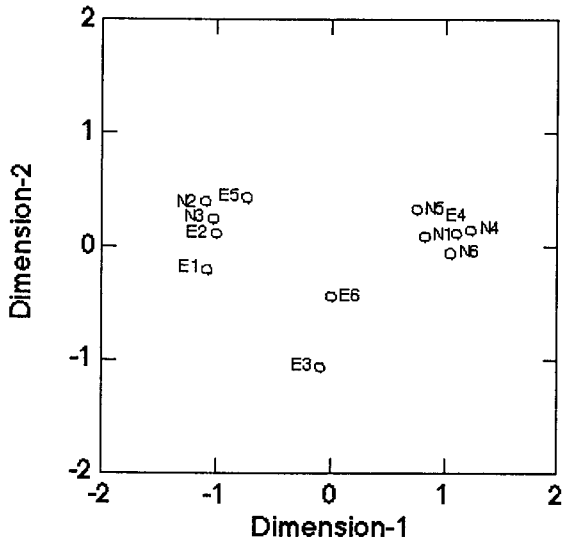



Fig. 4: MDS (r=1, Kruskal, Mono) plot with a normalised intersection matrix gives RSQ=0.975.

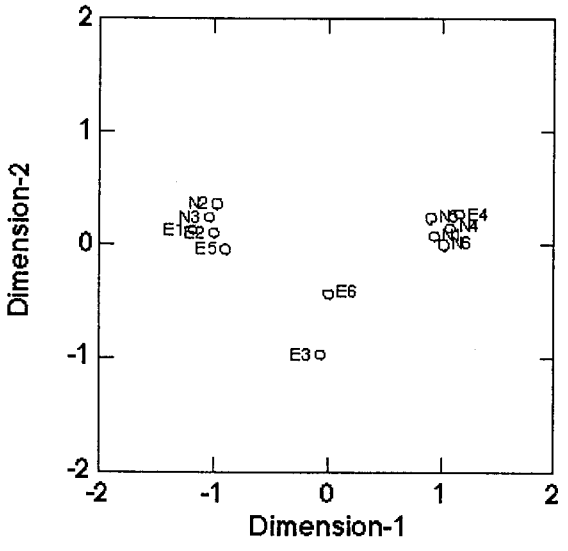



Fig. 5: MDS (r=1, Kruskal, Mono) plot with a binary normalised intersection matrix gives RSQ=0.995.

### *5.2.3 OUTCOME*

According to the RSQ of Fig. 4 (RSQ=0.975) and of Fig. 5 (RSQ=0.995), we can explain most of the variance among user data. However, the binary based plot of Fig. 5 (RSQ=0.995) is slightly better than that of Fig. 4 (RSQ=0.975). That is surprising, since the method ignores information about repetitive behaviour. Maybe such information is redundant in the context of this method.

## 5.3 EXCLUSION METHOD

This method is based on the exclusion as a metric of difference. Exclusion among two users is always given by two areas. The area of one user STV (user 1, Fig. 2) excluded from the area of a second user STV (user 2, Fig. 2), is not the same as the area of the second user STV excluded from the area of the first one. Since the two exclusion areas are asymmetric, the method does not allow for MDS as grouping algorithm.

### *5.3.1 METRIC*

This method measures the difference between two STVs by estimating how much of one user STV (column index in Table 2) is excluded from a second one (row index, Table 2).

Formula 3:

$$M_{p,q}^{EX} = \sum_{i=1}^{n} \left| \min\left(e_{p,i} - e_{q,i}, 0\right) \right|$$

where:

$M_{p,q}^{EX}$ : Exclusion metric between user p and q

$i$ : Summing Index STV elements

$n$ : STV length, upper summing limit

$e_{p,i}$ : STV element i for user p

$e_{q,i}$ : STV element i for user q

Following this procedure for all users, we get an mxm asymmetrical matrix (Table 2), where each element is a measure of exclusion (Formula 3). Since there were six novices (N1-N6) and six experts (E1-E6), m is 6+6=12.

Table 2: Numerical representation of exclusion matrix.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E6 | 6 | 43 | 47 | 51 | 70 | 50 | 47 | 35 | 73 | 5 | 171 | 0 |
| E5 | 17 | 15 | 14 | 69 | 48 | 67 | 23 | 7 | 64 | 21 | 0 | 24 |
| E4 | 9 | 44 | 47 | 56 | 70 | 55 | 47 | 35 | 73 | 0 | 171 | 8 |
| E3 | 17 | 41 | 41 | 68 | 77 | 62 | 28 | 28 | 0 | 21 | 162 | 24 |
| E2 | 17 | 15 | 16 | 68 | 81 | 67 | 19 | 0 | 66 | 21 | 143 | 24 |
| E1 | 20 | 16 | 19 | 73 | 85 | 72 | 0 | 7 | 54 | 21 | 147 | 24 |
| N6 | 3 | 41 | 44 | 28 | 48 | 0 | 47 | 30 | 63 | 4 | 166 | 2 |
| N5 | 3 | 39 | 42 | 41 | 0 | 33 | 45 | 29 | 63 | 4 | 132 | 7 |
| N4 | 2 | 41 | 42 | 0 | 55 | 27 | 47 | 30 | 68 | 4 | 167 | 2 |
| N3 | 16 | 11 | 0 | 68 | 82 | 69 | 19 | 4 | 67 | 21 | 138 | 24 |
| N2 | 18 | 0 | 15 | 71 | 83 | 70 | 20 | 7 | 71 | 22 | 143 | 24 |
| N1 | 0 | 41 | 43 | 55 | 70 | 55 | 47 | 32 | 70 | 10 | 168 | 10 |
| | N1 | N2 | N3 | N4 | N5 | N6 | E1 | E2 | E3 | E4 | E5 | E6 |

### *5.3.2 GROUPING ALGORITHM*

The grayscale representation (Fig. 6) of the exclusion matrix (Table 2) is generated by Mathematica (Wolfram, 1991) *ListDensityPlot* with the negative, inverted exclusion matrix as input. We use the negative matrix to obtain a consistent plot. Fig. 6 is only meant as a visualisation of Table 2, and is not an exact mapping. Since division by zero is not defined, the diagonal elements of Table 2 were directly mapped to the darkest graytone. Fig. 6 shows to what degree a *column* user STV is excluded from a *row* user STV. Darker matrix elements correspond to lower degree of exclusion.

To interpret the degrees of exclusion in Table 2, we suggest an iterative *predictor-corrector* algorithm. The *corrector* is an estimator for the threshold value so that only considering exclusion measures between that value and zero will give the predicted number (*predictor*) of user groups. The stop criterion for the iteration method is that the number of user groups given by the corrector, equals

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

the value of the predictor. Research on converge criteria is part of our future work, so now we simply assume convergence. For each iteration the corrector is modified in order to meet the stop criterion, according to the following rules: If we consider too few exclusion relations (i.e. the corrector is too close to zero), the number of groups will be higher than the predictor. If we consider too many exclusion (i.e. the corrector is too far from zero), many or all of the users will be related by exclusion statements, and the number of groups will be lower than the predictor. We give our predictor the value predictor=3. By visual inspection of Fig. 6 it appears reasonable to consider the darkest matrix elements only. Since these elements have numerical values equal to or below 8 (Table 2), we choose the initial value of the corrector to be 8.

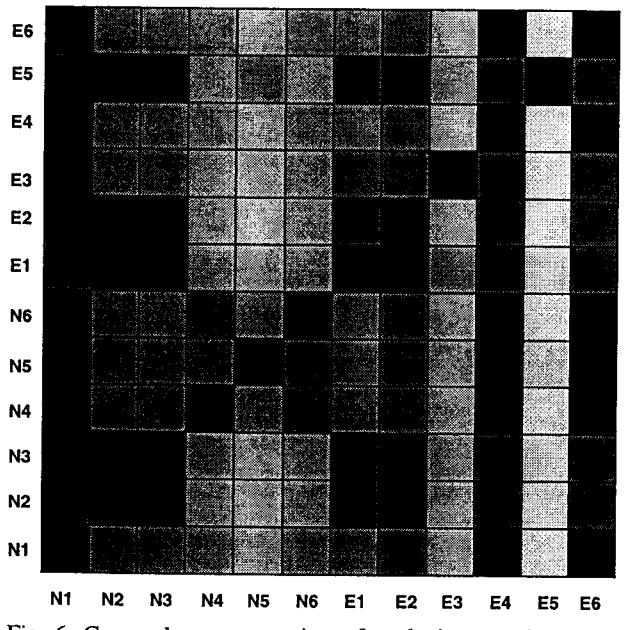



Fig. 6: Grayscale representation of exclusion matrix. Darker elements mean higher exclusion of *column* user STV from *row* user STV.

Diagonal elements are ignored, since each STV is fully similar to itself.

Since small differences indicate similarity, we can derive (based on Table 2) four similarity relations (Table 3).

Table 3: We can derive these four similarity relations.

| Similarity relation | User STVs of each relation |
|---|---|
| 1 | N1 ∈ N4, N5, N6, E6 |
| 2 | E4 ∈ N4, N5, N6, E6 |
| 3 | E6 ∈ N4, N5, N6, E4 |
| 4 | E2 ∈ N2, N3, E1, E5 |

All users that are related by an similarity relation are defined to belong to one group. Since the three first similarity relations (Table 3) are interrelated, this gives one group. The remaining, fourth similarity relation (Table 3) gives a second group. Users not appearing in any similarity relation define a separate group.

*5.3.3 OUTCOME*

Hence, the algorithm gives the following groups: {N1, N4, N5, N6, E4, E6}, {N2, N3, E1, E2, E5} and {E3}. We assumed that the number of groups should be three, so the stop criterion has already been met. If our prediction had not been met, we would have to try with a higher or lower corrector (according to the above mentioned rules) and go back to the start of the predictor- corrector algorithm. This algorithm is repeated until the stop criterion is met (convergence).

## 6 DISCUSSION

In order to validate the outcome of these three automatic methods, we performed a protocol analysis (Ericsson and Simon, 1984) of the task. This is manual work, based on analysis of video and verbal utterances in addition to logfiles. This is mostly feasible for simple tasks, where users basically follow one or a few strategies. This analysis showed that there are three distinct strategies solving the task. We call these strategies S1, S2 and S3. Table 4 shows the users according to their successful strategy.

Table 4: Manual protocol analysis of the task shows three distinct strategies and gives information about which user succeeded by which strategy.

| Strategy | Users according to strategy |
|---|---|
| S1 | N1, N4, N5, N6, E4, E6 |
| S2 | N2, N3, E1, E2, E5 |
| S3 | E3 |

The strategies are represented as STVs and have the same qualitative interpretation as the STVs of the users (N1-N6) and (E1-E6). We see that the correlation method and intersection method do not correspond fully with the outcome of the protocol analysis. The exclusion method, however, gives exactly the same results. So, the exclusion method is the best one with our combination of system, task and users behaviour. In the future, we want to find out how the different methods, especially the exclusion method, perform with other, more complex tasks.

We have seen that for a relatively simple task, the method which is purely analytical (exclusion method) is the best one. Measured by the RSQ-values, the intersection method is better than the correlation method, which is purely statistical. This indicates that in our context, statistical methods offer less explaining power than the analytical methods for strategy and plan recognition.

## 7 CONCLUSION AND FUTURE PERSPECTIVES

We have acquired results for one task only. To make our methods more reliable, we need to evaluate several tasks. For each task, we will validate our methods by manual protocol analysis.

We also plan to study learning experiments, in order to recognise the acquisition process of strategies.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

## 8 REFERENCES

Ericsson, K. A., & Simon H. A. (1984). Protocol analysis, verbal reports as data. The MIT Press.

Rauterberg, M. (1992). An empirical comparison of menu selection (CUI) and desktop (GUI) computer programs carried out by beginners and experts. *Behaviour and Information Technology 11*, pp. 227-236.

Rauterberg, M. (1996). A Petri net based analyzing and modelling tool kit for logfiles in human computer interaction. In (Yoshikawa, H., & Hollnagel. E., eds.) *Proceedings 'Cognitive Systems Engineering in Process ControlSCEPC'96*. Kyoto University, pp. 268-275.

Rauterberg, M., & Fjeld, M., & Schluep, S. (1997). Parallel or event driven goal setting mechanism in Petri net based models of expert decision behaviour. In (Bagnara, S., & Hollnagel, E., & Mariani, M., & Norros, L., eds.) *Proceedings of CSPAC'97*. CNR, Roma, pp. 98-102.

Ritter, F. E., & Larkin, J. H., (1994). Developing Process Models as Summaries of HCI Action Sequences. *Human Computer Interaction* 9, pp. 345-383.

SYSTAT Inc. (1989). SYSTAT®: The system for statistics. pp 93-166. SYSTAT program version 7.0.1 for PC.

Tauber, M. J., (1985). Top down design of human-computer systems from the demands of human cognition to the virtual machine - an interdisciplinary approach to model interfaces in human-computer interaction. In Proceedings of the IEEE workshop on languages for automation (Palma De Mallorca (E) June 28-29), pp. 132-140.

Wolfram, S. (1991). Mathematica®, A system for Doing Mathematics by Computer. 2nd Ed., AddisonWesley, pp. 164, 395, 819. Mathematica program version X3.0.1.1 for Silicon Graphics IRIX.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

# A Psychological Model of Air Traffic Control and Its Implementation

**Cornelia Niessen, Sandro Leuchter, & Klaus Eyferth**
Centre for Man-Machine-Systems Studies
Technical University of Berlin
TIB 4/5-3, Gustav-Meyer-Allee 25
D - 13355 Berlin
{niessen|leuchter|eyferth}@zmms.tu-berlin.de

**ABSTRACT**
In this paper, we describe a model of en-route air traffic controllers' cognitive activities in a dynamic man-machine system. The implementation of the model MoFl (*Modell der Fluglotsenleistungen*) is based on a production system in the programming language ACT-R (Adaptive Control of Thought - Rational, Anderson, 1993).

**KEYWORDS**
ACT-R, dynamic mental representation, air traffic control

**INTRODUCTION**
For various reasons, it can be useful to have a computer model of the operator's cognitive skills (see e.g., Opwis & Spada, 1994). The implementation of complex psychological assumptions

- can provide a more detailed and explicit description of every cognitive process involved than a verbal description,
- can test a theoretical framework by showing if the anticipated effects can be reproduced,
- can serve as a framework for generating hypotheses that support the empirical work, and
- can be used to analyse and predict the effects of future technological changes on the operator's cognitive activities in complex man-machine systems. These insights into the consequences affecting cognitve performance can be helpful for future system design or training concepts.

On the basis of a broad empirical work - interviews, simulation experiments, memory tests, and a card sorting task with experienced and less experienced en-route air traffic controllers and of theoretical considerations, the interdisciplinary research group "En-route Controller's Representation" (EnCoRe) constructed a model MoFl (*Modell der Fluglotsenleistungen*) of the cognitive activities of experienced en-route air traffic controllers. The air traffic control domain serves here as an example to model cognitive processing during control of complex and dynamic situations. The focus has been on issues concerning problems inherent to dynamic situations: mental representation of the changing situations, and the context-dependent flexible coordination of concurrent cognitive tasks. In comparison to other research (Freed & Johnston, 1995, Bass et al., 1995) in our approach we concentrated on modelling the cognitive abilities of air traffic controllers rather than perceptual and motor skills. According to the rate at which traffic situations changes, and the cognitive task of air traffic controllers, perceptual and motor skills were only treated in order to ensure a realistic model - environment interaction.

The implementation of the model is based on a production system in the programming language ACT-R 3.0 (Adaptive Control of Thought - Rational, Anderson, 1993). As programming environment, ACT-R includes a broad and detailed theoretical framework of human cognition. For the most part, ACT-R is suitable for modelling the cognitive performance of en-route air traffic controllers. But, for some aspects of dynamic situations ACT-R does not provide convincing solutions.

The aim of this paper is to present the construction and the implementation of the model. This includes the principles of construction and implementation of our model, and the discussion of two special issues concerning the cognitive architecture of ACT-R: "dynamic representation" and "executive control". This paper is divided into three sections:

- short description of the air traffic control task
- the framework for the implementation: the cognitive architecture ACT-R
- description of the psychological assumptions of the model and its implementation

**THE AIR TRAFFIC CONTROL TASK**
On the basis of different sources of information (e.g., radarscreen, flight strips, head-phone communication with pilots), air traffic controllers have to control complex, dynamic, and time-constraint traffic situations in order to diagnose risky relationships between aircraft and to solve potential conflicts. Therefore, they have to perceive, comprehend, and anticipate multiple characteristics of many aircraft while new incoming aircraft create new traffic relationships for evaluation. It's a common assumption, that in complex technological systems of a dynamic nature operators develop a mental representation

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

of the task environment with which they interact. Diagnosis, decisions on future cognitive activities and actions are based on these insights into current and anticipated structures of the changing situation. Air traffic controllers express with the term *picture* (e.g., Whitfield & Jackson, 1982; Falzon, 1982) what is often described as *situation awareness* (e.g., Endsley, 1995; Flach, 1995): a mental representation of the current and future traffic situation.

By modifying the framework of cognitive task analysis (the "decision ladder", Rasmussen, 1986), extensive interviews with seven experienced controllers provided a first explorative functional analysis of main tasks used to build up and maintain this mental *picture* of the traffic situation.

According to verbal reports of the air traffic controllers, the diagnosis of potential conflicts between aircraft contains stages, which are characterized by an increasing restriction and specification of the problem space. These stages are: *observing* the whole situation, *analysing* the parameters of selected aircraft, and *anticipation*. In the first step (*observation*) the operator monitors the whole situation in order to get a quick overview of the whole traffic situation. The goal of conflict detection demands selection strategies during radar-screening to structure the representation (see e.g. Amaldi & Leroux, 1995). According to the verbal reports, experienced controllers classify the aircraft on the basis of these signals (proximity, vertical movement, etc.) into two groups: those aircraft which have to be further analyzed (*analysing the parameters*) and anticipated (*anticipation*) in order to check for future conflicts, and those which are separated safely during that moment. The initial steps towards intervention and conflict resolution could be described according to Rasmussen´s stages (define task, fomulate procedures, and execute).

In order to model the air traffic controller's *picture* and the processes used to build up and to maintain this mental representation of the changing traffic situation, experiments provided a more detailed analysis of the following topics:

- information selection and recall,
- relational structure of the representation, and
- anticipation and conflict management.

The experimental work with real time simulation was based on a realistic simulation system of the control task called "En-route Controllers Representation - Programmable Airspace Simulation" (EnCoRe-PLuS) (Bierwagen, 1996). This system simulates air traffic control scenarios providing radar screen runs, electronic flight strips, and head-phone communication with a ghost-pilot; it also allows the user to set up experimental procedures and to keep logfiles of all system activities.

The results of this empirical work led to the conceptualization and the implementation of a model that describes the cognitive activities of air traffic controllers. The implementation of the model is connected with a modified version of EnCoRe-PLuS. EnCoRe-PLuS provides a real-time simulation environment. Predefined traffic builds up a simulation scenario that interacts with the model:

- The model can actively access new information about the changing traffic situation and can integrate it to its representation of the current situation.
- The model is informed about events within the task environment (e.g., incoming aircraft)
- The model can intervene with the traffic environment in order to solve conflicts.

## MODELLING MENTAL PROCESSES OF EXPERIENCED OPERATORS DURING CONTROL OF A DYNAMIC MAN-MACHINE SYSTEM

For modelling mental processes of experienced air traffic controllers during control we have used the production system ACT-R 3.0. ACT-R provides a suitable framework: 1. as a psychological framework of human cognition, it also describes an environment for implementation, 2. ACT-R is based on explicit and very detailed assumptions about the cognitive architecture, and 3. as an environment for implementation, it is available in the public domain at no costs. In addition ACT-R has been applied to modelling a great number of problem solving tasks and is still in progress (e.g., ACT-R Perceptual - Motor Layer, RPM).

Even within such a framework, the conceptualization and implementation of mental processes in dynamic environments, as in the case of air traffic control, demand additional assumptions about three aspects of the dynamic task environment. 1. The continous changes of the situation. These changes do not allow fixed sequences of cognitive processing, they rather call in a cyclic update of varying relations as a basis of situational awareness. 2. The necessity to predict future states of the situation in order to predict potential conflicts. Such predictions alter the goals of ongoing control activities. 3. The demands to coordinate and to sequence simultanious requirements of the control task.

Widely used concepts for adaptive control of complex task enviroments (e.g., Anderson, 1993; Rasmussen, 1986; Hacker, 1978) concentrate on rather static tasks and on invariant goal structures. For example the cognitive architecture of Anderson's ACT-R does not take into account that in dynamic situations the operator has to continuously update her or his mental representation. In addition, such production systems are directed by a fixed goal hierarchy. But in the case of the changing and complex situation requirements, the controller has to coordinate the cognitive activities. This coordination is context-dependent: it does not follow a pre-defined goal hierarchy.

Recently there are some promising attempts to formulate cognitive architectures that deal with the specific demands of a dynamic task environment. For example, as a conceptual neighbor to ACT-R and SOAR, a new computational framework, the executive - process

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

interactive control (EPIC), is proposed for this kind of human performance (Meyer & Kieras, 1997a,b; Meyer et al., 1995). Perceptual, cognitive, and motor processors have been built up for modelling cognitive processes during the performance of multiple concurrent tasks. The perceptual processor provides a continously update of the task environment. Within the cognitive processor, concurrent tasks can be scheduled by flexible executive processes that control relative task priorities. Also the architecture for human representation in complex system, "Man Machine Interactive Design and Analysis System" (MIDAS), promises a modelling environment that provides an updateable mental representation of the task environment and flexible scheduling of multiple task performance (Corker & Smith, 1993).

The implementation of the model "MoFl" (*Modell der Fluglotsenleistungen*) is based on ACT-R 3.0. The basic assumption is that cognitive skills are composed of production rules. A production rule is a modular piece of knowledge. Combining these rules into a sequence represents complex cognitive processes. ACT-R includes two kinds of knowledge representation: declarative and procedural knowledge. The basic units in declarative memory are so-called working memory elements (WMEs). A WME is an object with identity. It has named slots that can be filled with Lisp objects or references to other WMEs. References to other WMEs can be interpreted as relations, so that a semantic net with WMEs as nodes and references for relations is spread out. ACT-R defines an object-oriented structure for declarative memory. Every node in the net is an object of a certain class. A class is declared by naming all slots an object of this class will have. Subclassing is possible. Every WME has an activation level. It is manipulated by the programming environment. A special structure within the declarative part of the memory is the goal-stack. WMEs can be pushed onto and popped from this structure. The topmost WME is the current goal.

Production rules are the procedural part of memory. They consist of a condition and an action part. Conditions and actions refer to WMEs. The application of a production rule is realized by a simple pattern-matching mechanism. In order to support goal-directed performance, the first condition of every production rule must match the current goal. If all conditions of a production rule are true, then the action part is executed. Possible actions are: manipulation of the goalstack (push and pop), creation and deletion of WMEs, and modification of the slots of already retrieved WMEs. An ACT-R run consists of the continous application of production rules.

The prioritizing of processing is controlled by the activation parameter in ACT-R as well as by the current goal. A production is applied if it fires. A rule can fire if all conditions are fulfilled. Typically the fastest production will fire. The speed of application is mainly computed by the time it takes to retrieve the condition WMEs.

Activation signifies the current relevance of a WME for the processing of information. Sources of activation are the encoding process, execution of a production (addition of new WMEs), and creation of a goal node. The more activated a WME is, the faster it is retrieved. This means that if various WMEs match the pattern of a production rule, the most activated WME is retrieved. If various production rules can be applied, that production rule fires that retrieves the most activated WMEs. A WME can only get retrieved if its activation is above a certain level. But in the case of air traffic control there are three cases in which an inactive WME also has to be retrieved. In the first case, the controller has to update his mental representation continuously. Empirical work showed that controllers reduce the problem space by paying attention to meaningful signals for conflict detection during radar-screening. Because of these signal features, aircraft become focal. That means that they are attention demanding objects, therefore highly activated. Aircraft without these features are *extrafocal* (less activated). For these extrafocal aircraft there is no further demand for processing and they become inactive. But, in contrast to ACT-R, these inactive WMEs have to be retrieved in order to update them. Second, activation is increased not only by the encoding process. It is also guided by the encoding of signal features of aircraft. The third case concerns the context-dependend coordination of a goal. The high activation level of a goal that targets the solution of a detected conflict between aircraft can be decreased, it may be put aside for a while if there is enough time remaining for the solution. But at a certain point, activation has to increase suddenly in order to retrieve this WME and to apply the appropriate production rule in order to solve the conflict. Otherwise the both *inactive* aircraft will collide.

Additional features of ACT-R are learning mechanisms to adjust WME and production parameters, partial matching, and the aggregation of production rules. These features are not used in our model.

## THE MODEL

In this section, the psychological assumptions, based on experimental work and theoretical considerations, and the implementation of the main components and functions of the model MoFl are summarized.

MoFl describes three main cycles of information processing, (i.e., *monitoring*, *anticipation*, *problem resolution*) operating on different parts of the situation representation, called the *picture* (see Figure 1). The coordination of these processes is driven by *control procedures*. *Monitoring* and *anticipation* are diagnostic processes (conflict detection), *problem resolution* is the preparatory step for intervention by the controller.

### The Monitoring Cycle: Data Selection and Update

The *monitoring cycle* includes data selection procedures and the regular update of aircraft features. In an experiment on data selection, 36 en route controllers had to control familiar and unfamiliar dynamic airspace situations. In order to investigate information selection, data of aircraft on the radar screen and the flight-strip-system were masked, but could be unmasked by moving the pointer of the mouse to the respective location.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

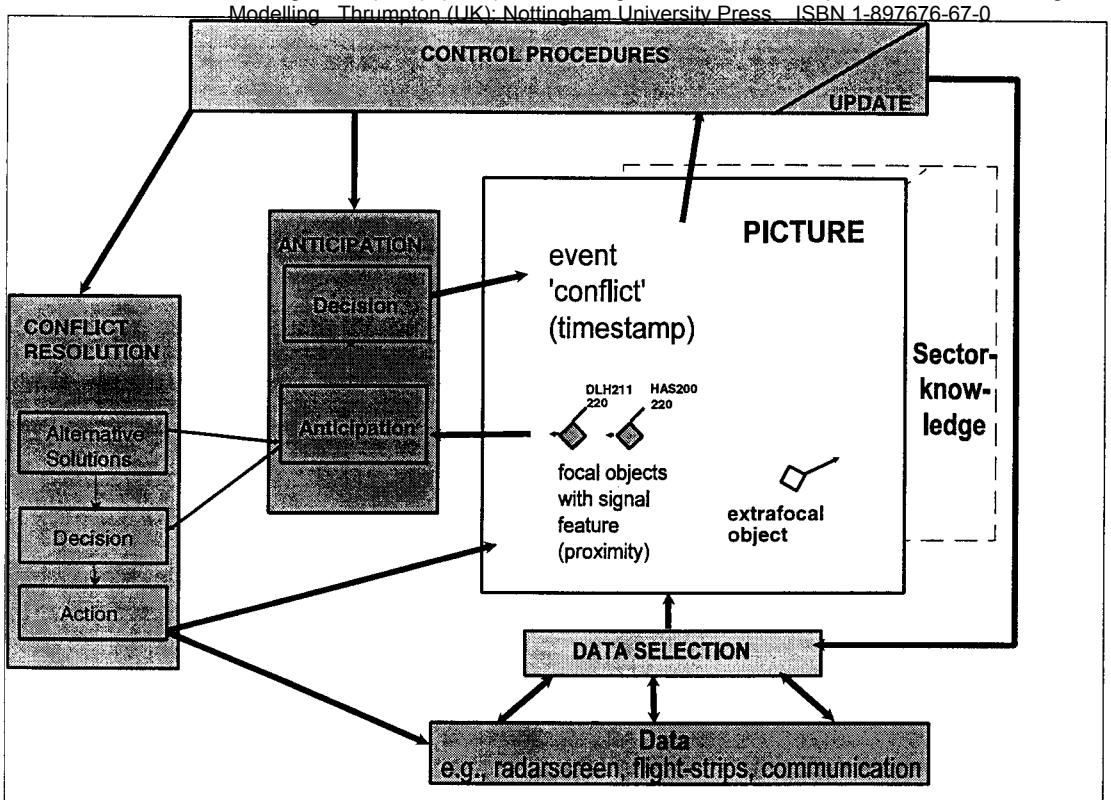



Figure 1: The structure of the air traffic controller's model of cognitive activities

Frequencies and durations of the unmasking were recorded. The data showed, that the representation of the current traffic situation was build up under considerable reduction of information. The controller selects relevant features of aircraft, especially identification codes, the horizontal and vertical positions of objects, and flight directions. In addition, our interviews and the literature indicate that the controller searches for meaningful signals in order to detect conflicts during radar-screening. These are aircraft features like vertical movements, proximity to other aircraft or to points in airspace where conflicts frequently occur (e.g., Niessen et al., 1997; Amaldi & Leroux 1995).

According to these signal features, aircraft become *focal* (highly activated), that means that they are attention demanding objects. Aircraft without such features are *extrafocal* (less activated). In the dynamic environment of air traffic control, objects have to be updated continuously. There is a relationship between the semantics of objects and the frequency of updating: focal, attention-demanding objects demand a higher monitoring frequency than extrafocal objects. This assumption has been supported by results of a memory test: positions of extrafocal (inrelevant) aircraft were reproduced back in time, whereas positions of attention demanding objects (e.g., conflictions, and climb or decend) were reproduced correctly (for similar results, see Boudes et al., 1995). This bias indicates, that there is an interaction between the semantics of objects and the updating frequency: the more the current position of aircraft demands attention the better they were reproduced.

The communication between the controller and the task environment, and the data selection were implemented as follows: Communication between MoFl and EnCoRe-PLuS is realized by socket communication. Two ways of communication are provided:

- *asynchronous communication*: Special events in the task environment, like pilot-initiated radio communication or signals suddenly appearing on the radar-screen, are announced to MoFl by EnCoRe-PLuS. After every application of a production rule, a Lisp function hooked to the ACT-R specific production-cycle-hook, checks for new messages and triggers appropriate Lisp call-back functions that create new WMEs for further processing.
- *synchronous communication*: MoFl identifies an internal demand for new information about a specific object within the task environment or the internal control-flow suggests to update aircraft information. This demand is fulfilled by an active request to the simulation environment. The response is integrated into the *picture* by call-back functions.

If the data selection procedures are triggered, appropriate goals are put onto the goal-stack to enable the following processing sequence:

1. *choose aircraft*: according to aircraft focality and state of the *picture*, decide which aircraft has to be updated.
2. *make an information request*: according to the state of the object which is going to be updated, choose which information has to be requested, and trigger

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

the appropriate Lisp function. The response of EnCoRe-PLuS is handled by a call-back function that generates a goal.

3. *take new information into the picture*: This goal is processed by a production that modifies the WME representing this information.
4. *test new data for signal features*: the updated WME is tested for changes of signal features such as changing flightlevel (vertical movement), or proximity to other aircraft.

### Anticipation

The next step in diagnosis consists of an *anticipation cycle* which operates on the focal objects. For each attention-demanding (*focal*) aircraft or aircraft relationships, a future state is anticipated seperately. The goal of the anticipation cycle is to create new cognitive processing information about aircraft. Depending on the results of anticipation, aircraft with signal features can then be represented as *events*. An event reflects the type of relation between aircraft or relations between aircraft and airspace features in future time and space. The anticipation allows to decide (*decision*) if the future trajectories of aircraft result either in a conflict, in a safe separation, or the demand for more monitoring. In an experiment on conflict-management, different types of clearcut and potential conflicts were varied in a 70 minutes traffic scenario according to the *Eurocontrol Air Space Model* (EUROCONTROl, 1994). The EUROCONTROL classification has two dimensions: 1. different tracks (same, opposite, crossing), and 2. level- or climb/ decend-flight. 36 controllers had to detect and to solve the conflicts. The data showed that controllers did not differentiate between conflicts (separation minimum: 5 nautical miles) and potential conflicts (10 nautical miles): they intervened in all cases. This indicates that conflict detection is not based on a calculation but on fuzzy estimation. The controllers always chose the safer way by overestimating the risk.

We assume that, if a conflict is detected, the event *conflict* includes an estimation of the time remaining for conflict solution (*timestamp*). Relations which have proved to be safe, are no longer in the focal part of the *picture* and become extrafocal at this time. This indicates that there is almost no demand for cognitive processing, except for updating. If the operator is not sure about the potential conflict, the event *monitoring* becomes *focal*, indicating both a higher frequency of monitoring and also a high demand for further anticipation. This distinction of aircraft relationship has been supported by the results of a card sorting task with 18 air traffic controllers. As expected the controller showed a tendency to classify traffic scenarios on the basis of anticipation.

The anticipation cycle is implemented by sequenced production rules testing four questions:

1. Are aircraft on the same airway, or on crossing airways?
2. Have aircraft the same altitude or is at least one in climb or descend?
3. Simulation of the future movement of aircraft using *velocity leaders*. A velocity leader is an graphical arrow element on the radar screen showing the estimated movement of aircraft for a certain lapse of time. Will there be a violation of the separation criterion (*anticipation*)?
4. How certain was this simulation? Certainty is measured by the time remaining for the violation of the separation criteria. In addition the latest time for conflict solution is calculated (*timestamp*).

According to this sequence focality of aircraft-WMEs is modified, or events are created.

### The Picture

The resulting *picture* is characterized as a representation of objects, events, and objects with reference to other objects, and / or airspace structure. Objects with signal features are represented focally, objects without these features extrafocally. In addition, events which indicate the meaning of aircraft relations in future time and space are represented focally. Within the air traffic control domain, the term *picture* describes the idea of a global mental representation of the current and future traffic situation in working memory. From a psychological perspective, we assume the *picture* as an analogous non-symbolic mental representation of the situation. There is some empirical evidence that experienced controllers anticipate future states of aircraft without calculating the trajectories. This indicates that they build up a non-metric, analogous representation of the situation. In assuming such an analogous representation, we follow Craik's (1943) and Johnson-Laird's (1983) basic ideas of a functional internal model that parallels processes of the external world.

The *picture*

- is understood as an active knowledge-based construction of meaningful relations between elements of a situation, and not as an addition of perceptions.
- is incomplete with regard to the content of information and is temporary. The representation is build up by schemata in order to serve current functions, and is not stored in long term memory.
- can be manipulated by drawing inferences, by making predictions, by understanding phenomena, by deciding what further processing or action to take, and by controlling the execution.

The implementation emulates the *picture* as the totality of the cognitively available objects at a given time, their features, and their perceived and infered relations in actual and future time and space in terms of WMEs. Since it is not possible to model an analogous representation of space on digital computers, the implementation's *picture* is a semantic net of airspace objects, anticipated events, and inferenced actions that are represented as WMEs. Some of these objects have spatial positions that make it possible to define them by positions. More sophisticated operations such as retrieval by distance to other airspace objects have to be emulated.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

We used the object-oriented features of ACT-R to define the structure of the *picture* (see, Figure 2). Every airspace object has a position on the radar screen. Derived classes are *airways*, *sector boundaries*, and *aircraft* which have additional slots including callsign, speed, and altitude. Aircraft are specialized to *incoming*, *changing altitude*, and *near to another airspace object* (proximity). For every class, instances are generated and modified as WMEs in working memory by data selecting productions during the monitoring cycle. Events represent infered knowledge about aircraft. All events refer to aircraft objects. Instances are generated by production rules in the anticipation and conflict resolution module. They belong to the event-subclasses: *monitoring*, *conflict*, and *resolution*. Conflicts can be *crossing* or *chain*. Conflict events have an additional slot that holds a reference to the conflict partner.

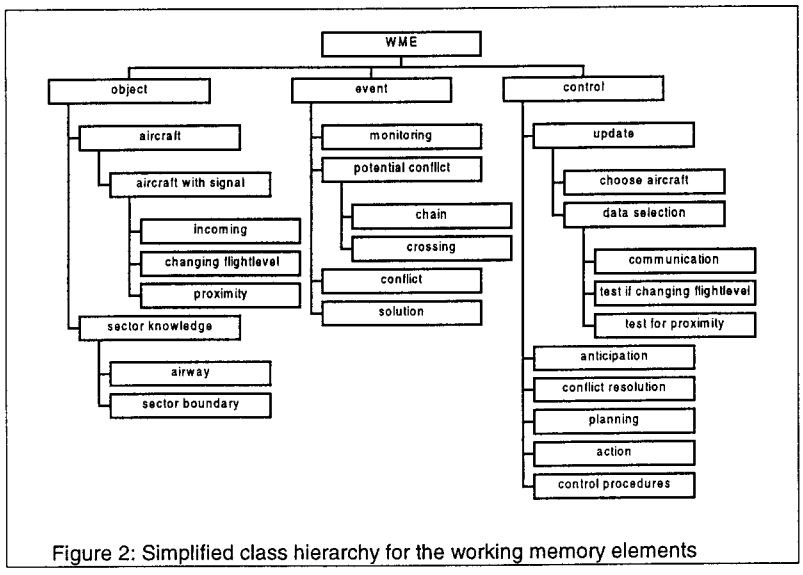


Figure 2: Simplified class hierarchy for the working memory elements

### Conflict Resolution

If conflicts are detected, the *problem resolution cycle* initiates several steps to prevent an impending conflict. The controller has to select the most urgent conflict in order to generate or recall solutions (*alternative solutions*). Next, the operator has to check that the solution does not generate new follow up conflicts (*decision*). We assume that the controller checks by running a mental simulation of the solution (as in the *anticipation cycle*). The results of this model are executed (*action*).

The implementation uses a predefined set of standard solutions fitting certain types of conflicts. To use this set the class of the conflict is determined by production-rules. According to this classification some solutions are generated from the standard solution set. The production rules of the simulation in the *anticipation cycle* are triggered by goals indicating the solutions that have to be taken into account. If a solution does not produce follow-up conflicts a solution-WME is generated. A solution consists of a sequence of actions that have to be executed by the model. The time remaining for the first intervention of the sequence is stored in the solution-WME. To execute an intervention sequence Lisp functions interact with the task environment EnCoRe-PLuS.

### Control Procedures

The multitude of represented objects, relations, and features within the *picture* demands that the controllers prioritize the processing at any one time. The coordination of the above describes modules (data selection and update, anticipation, conflict resolution, and action) is driven by *control procedures*. We assume that the different processing components cannot be interrupted. The controller has to switch between them: for example, between the solution of a conflict and further monitoring (update including data selection). On the basis of the state of the *picture*, control procedures select the most important and most urgent processing demand.

In ACT-R, Anderson postulates a hierarchical goal structure that directly reflects the task dependency in the environment. To model this hierarchy of goals, several WMEs can be pushed onto the goalstack, a special structure within working memory. Processing is controlled by the current goal, which is the first element of the goalstack. The current goal spreads activation among its neighbors in the semantic net. The system focusses only on this top goal at this time. But, because of the dynamic task environment of air traffic control, there is no fixed hierarchical goal structure. Therefore, the continuously changing situation demands another prioritizing of the processing of simultaneously on-going events at any particular time. In addition, time contraints in this context force a flexible and appropriate selection of the most relevant demand for processing. In order to model this contextualized scheduling of processing, we had to postulate a different concept. Our assumption is that the scheduling of processing is determined by the state of the whole mental representation of the traffic situation.

Several tasks are active at every moment. Every task is done by one of the modules *data selection*, *anticipation*, or *conflict resolution*. The superior control procedures module has to build up an ad hoc process flow depending on the current structure of the *picture*. To achieve this, we assume that the modules cannot be interrupted and are exclusive. The process flow is done by meta productions in the *control procedures* module that trigger a module with an object or event as parameter. In order to trigger a module and make it not interruptible, we introduced a new class of WMEs. These *control*-WMEs are the only ones that get onto the goalstack.

The start of every module is a *top level production*. It is triggered by a *top level goal*. This kind of production will push new subgoals onto the goalstack that will trigger

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

other productions of that module. Every production has to clean the goalstack by popping its trigger-WME. When a module is finished the goalstack should then be clean. The productions of the *control procedures* are triggered by the *controlflow*-goal, which has no parameter. This goal is never popped. Thus when the goalstack is "clean" it is on top of the goalstack and thus the current goal triggers the *control procedures*-module again. Processing radio communication when a plane announces that it is going to enter the sector, is the only reason to interupt a module, make a mark in the working memory, and continue the module. The mark has a high priority so that it will be processed soon.

The meta production rules of the *control-flow*-module for the air traffic controller model use this prioritizing rules:

1. if a solution-WME exists in the *picture* and it is time to solve, then do *action* on this solution, else
2. if a conflict-WME exists and it is time to do, then *conflict resolution*, else
3. if a monitoring-event or an aircraft-WME with a signal (*incoming*, *changing altitude*, or *proximity*) exists in the *picture*, then do *update* and *anticipation* on this WME, else
4. if an aircraft-WME exists, then do *monitoring* on it.

Every solution-WME and every *conflict*-WME has a slot, where it represents when it is supposed to happen. The control productions use a function, that compares this ideal time with the current time. It fires the appropriate action according to a predefined bias.

If the current goal is *controlflow*, only the meta-productions are able to fire. They match patterns against the *picture* according to the prioritization scheme listed above. The chosen action will generate a new *control*-WME (CF) of the appropriate subclass. It refers to the detected aircraft-WME or event-WME. The goalstack consists now of (*controlflow*,CF). This triggers the toplevel production for CF. It will produce new control-WMEs probably refering to the detected WME, pop CF, and put the new control-WMEs onto the goalstack. They trigger new sublevel productions that all pop their trigger. When the module for CF is finished, the goalstack is (*controlflow*), meaning that only the meta-productions are able to fire.

The model deals well with the dynamic environment by using this control scheme. If another task needed interruptible modules, the control procedures would have to be triggered after every production cycle within the module, and the controlflow WMEs would have to be stored in the *picture*, when they are inactive. The meta productions would then trigger the most important controlflow-WME or generate a new one.

## CONCLUDING REMARKS: EVALUATION OF THE MODEL

The construction and implementation of the above described model is based on a broad experimental work. Early in 1998 we will evaluate our model with empirical data. Three simulation experiments with experienced air traffic controllers are planned in order to investigate time parameters of conflict detection, the content of the *picture*, and the distribution of activation within the controller's *picture*. These data will be compared to the results of model simulation runs using the same task environment.

## ACKNOWLEDGMENTS

This research project has been sponsored by the Deutsche Forschungsgemeinschaft, Ey 4/16-2 , Fr 375/48-2. We gratefully acknowledge the contributions of Thomas Bierwagen (DFS) to the conceptions of our model.

## REFERENCES

Amaldi, P. & Leroux, M. (1995). Selecting Relevant Information in a Complex Environment: The Case of Air Traffic Control. In: Norros (Eds.), *5th European Conference on Cognitive Science Approaches to Process Control* (pp. 89-98). Finland: VTT Automation.

Anderson, J.R. (1993). *Rules of the Mind.* Hillsdale: Lawrence Erlbaum.

Bass, E.J., Baxter, G.D. & Ritter, F. (1995). Creating Models to Control Simulations: A Generic Approach. AI & Simulation, pp 18-25.

Bierwagen, T. (1996). Programmsystem EnCoRe-PLuS. Über die Möglichkeiten der programmierbaren Luftraumsimulation. *ZMMS-Forschungsbericht, 96-2.* Technische Universität Berlin.

Boudes, N., Amaldi, P., Cellier, J.M. & Leroux, M. (1995). Forseeing Judgement in an Informationally Rich Environment: The Case of Air Traffic Control. In: Norros (Eds.), *5th European Conference on Cognitive Science Approaches to Process Control* (pp. 76-88). Finland: VTT Automation.

Corker, K.M. & Smith, B.R. (1993). An Architecture and Model for Cognitive Engineering Simulation Analysis: Application to Advanced Aviation Automation. Presented at AIAA Conference on Computing in Aerospace, San Diege, CA. available at http://george.arc.nasa.gov/af/aff/midas/www/AIAA_Final.txt

Craik, K.J.W. (1943). *The Nature of Explanation.* Cambridge: University Press.

Endsley, M.R. (1995). Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors, 37(1),* 32-64.

Eurocontrol (1994). The Eurocontrol Airspace Model: An Introduction for Potential Users. Eurocontrol-Workshop "Cognitive Aspects in ATC", May 1996.

Falzon, P. (1982). Display Structures: Compatibility with the Operator's Mental Representation and Reasoning Process. *Proceedings of the 2nd European Annual Conference on Human Decision Making and Manual Control,* pp 297-305.

Flach, J.M. (1995). Situation Awareness: Proceed with Caution. *Human Factors, 37(1),* 149-157.

Freed, M. & Johnston, J.C. (1995). Simulating Human Cognition in the Domain of Air Traffic Control. In M.T. Cox & M. Freed (Eds.), 1995 AAAI Symposium on Representing Mental States and Mechanism. Menlo Park, CA: AAAI Press.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

Hacker, W. (1978). *Allgemeine Arbeits- und Ingenieurspsychologie*. Bern: Hans Huber.

Johnson-Laird, P.N. (1983). *Mental Models*. Cambridge, Massachusetts: Harvard University Press.

Meyer, D.E. & Kieras, D.E. (1997a). A Computational Theory of Executive Processes and Multiple-Task Performance: Part 1. Basic Mechanisms. *Psychological Review, 104(1)*, pp 3-65.

Meyer, D.E. & Kieras, D.E. (1997b). A Computational Theory of Executive Processes and Multiple-Task Performance: Part 2. Accounts of Psychological Refractory-Period Phenomena. *Psychological Review, 104(4)*, pp. 749-791.

Meyer, D.E., Kieras, D.E., Lauber, E., Schumacher, E.H., Glass, J., Zurbriggen, E., Gmeindl, L. & Apfelblatt, D. (1995). Adaptive Executive Control: Flexible Multiple-Task Performance Without Pervasive Immutable Response-Selection Bottlenecks. *Acta Psychologica 90*, pp. 163-190.

Niessen, C., Eyferth, K. & Bierwagen, T. (1997). Modelling Cognitive Processes of Experienced Air Traffic Controller. In: S. Bagnara, E. Hollnagel, M. Mariani & L. Norros (Eds.), *Sixth European Conference on Cognitive Science Approaches to Process Control, Time and Space in Process Control*, 23.9.-26.9.1997 Baveno, Italy.

Opwis, K. & Spada, H. (1994). Modellierung mit Hilfe wissensbasierter Systeme. In: *Enzyklopädie der Psychologie* (pp 199-248). Göttingen: Hogrefe.

Rasmussen, J. (1986). *Information Processing and Human-Machine Interaction.* New York: North-Holland.

Whitfield, D. & Jackson, A. (1982). The Air Traffic Controller's Picture As an Example of Mental Model. In: G. Johannsen & J. E. Rijnsdorp (Eds.), *Proceedings of the IFAC Conference on Analysis, Design and, Evaluation of Man-Machine Systems* (pp 45-52). London: Pergamon Press.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

# Spatial Learning and Localization in Animals: A Computational Model and Behavioral Experiments

**Karthik Balakrishnan, Rushi Bhatt, and Vasant Honavar**
Artificial Intelligence Research Laboratory
Department of Computer Science
Iowa State University
Ames, IA - 50011, USA.
+1 515 294 3588
{balakris|rushi|honavar}@cs.iastate.edu

## ABSTRACT

This paper describes a computational model of spatial learning and localization. The model is based on the suggestion (based on a large body of experimental data) that rodents learn metric spatial representations of their environments by associating sensory inputs with dead-reckoning based position estimates in the hippocampal place cells. Both these sources of information have some uncertainty associated with them because of errors in sensing, range estimation, and path integration. The proposed model incorporates explicit mechanisms for information fusion from uncertain sources. We demonstrate that the proposed model adequately reproduces several key results of behavioral experiments with animals.

**Keywords:** cognitive modeling, cognitive maps, Hippocampus, probabilistic localization.

## INTRODUCTION

Animals display a wide range of complex spatial learning and navigation abilities (Schone, 1984; Gallistel, 1990), far more impressive than the capabilities of contemporary robots. Considerable research effort has been devoted to understanding different aspects of these spatial behaviors through cognitive, behavioral, neurophysiological, and neuropharmacological studies. This has resulted in a large corpus of experimental data, a number of theories and models of animal spatial learning, and several implementations of such models in robots and other artificial automata (Mataric, 1992; Kuipers and Byun, 1991; Kortenkamp, 1993; Bachelder and Waxman, 1994; Recce and Harris, 1996). However, animal spatial learning is still far from being completely understood or successfully imitated.

Based on a large body of experimental data it has been suggested that rodents learn *cognitive maps* of their spatial environments (Tolman, 1948). These cognitive maps have been postulated to contain *metric* information, i.e., the places in the environment are represented in a metric coordinate system, allowing the animal to take novel short-cuts and measured detours. In addition, there is also a vast body of experimental data from *lesion studies* of hippocampal regions and *cellular recordings* of hippocampal cells that directly implicate the *hippocampal formation* in rodent spatial learning (O'Keefe and Nadel, 1978). Based on this data, O'Keefe and Nadel proposed the *locale system hypothesis*, suggesting that the hippocampal place cells learn metric cognitive maps by associating *sensory inputs* with *dead-reckoning*[1] position estimates generated by the animal.

In the two decades since the locale hypothesis was first proposed, a number of computational models of hippocampal spatial learning have been developed (Trullier et al., 1997). Surprisingly, only a few of the models support *metric* spatial representations. Furthermore, the few models that are based on the locale hypothesis make the unrealistic assumption that the two information streams, namely, sensory inputs and dead-reckoning, are largely error-free. However, sensory and dead-reckoning systems of animals are prone to several sources of errors (e.g., errors in place recognition, distance estimation, dead-reckoning drifts, etc.), and any computational model of hippocampal spatial learning and localization must therefore be capable of satisfactorily dealing with these associated uncertainties.

In this paper we develop a computational model of hippocampal spatial learning that allows the animal to learn a metric place map (or a *cognitive map*) and that explicitly addresses information fusion from uncertain sources. Following a brief discussion of experimental data supporting the model, we present the key features of the model and simulation results that demonstrate that the proposed model satisfactorily reproduces the results of behavioral experiments on gerbils reported by Collett et al., (1986). We also discuss the relationship between this neuro-cognitive model and some approaches to spatial learning that have been employed in contemporary robotics.

## HIPPOCAMPAL SPATIAL LEARNING

The *hippocampal formation* is one of the highest levels of association in the brain and receives highly processed sensory information from the major associational areas of the cerebral cortex (Churchland and Sejnowski, 1992). It is composed of the *dentate gyrus* (Dg), and areas *CA3* and *CA1* of Ammon's horn as shown in Figure 1. It receives input primarily from the *entorhinal cortex* (EC), which is a part of a larger convergence area called the *parahippocampal cortical area*, and outputs to the *Subiculum* (Sb) and back to the EC (Churchland and Sejnowski, 1992). (For other anatomical and physiological details the reader is referred to (Churchland and Sejnowski, 1992).)

The *hippocampal formation* has been strongly implicated in animal spatial learning and localization based on evidence from *hippocampal lesion studies* and *cellular*

[1] Dead-reckoning or path-integration refers to the process of updating an estimate of one's position based on self-knowledge of time, speed, and direction of self-motion.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Nottingham (UK): Nottingham University Press. ISBN 1-897676-67-0

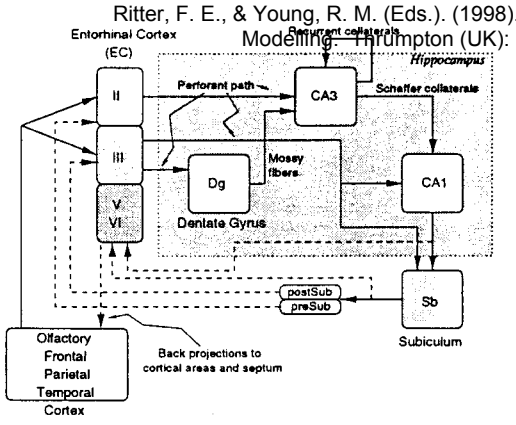



Figure 1: Anatomy of the hippocampal formation.

*recordings*. While hippocampal lesions have been found to produce *severe deficits* in learning *spatial tasks* such as the *object-place task* (Churchland and Sejnowski, 1992), and the ability of rodents to traverse complex mazes (cf. appendix of O'Keefe and Nadel, (1978)), cellular recordings have led to the discovery of *place cells* and *head-direction cells* which demonstrate *highly correlated* firings during the execution of such tasks. Pyramidal cells in regions CA3 and CA1 of the rat hippocampus have been found to fire selectively when the rat visits particular regions of its environment. These cells thus appear to code for specific places and have been labeled *place cells* (O'Keefe and Dostrovsky, 1971). Cells with such location-specific firing have been found in almost every major region of the hippocampal system, including the EC, the Dg, regions CA3 and CA1, the Sb, and the postsubiculum.

In addition to place cells, *head-direction cells* have also been discovered in the hippocampal region (Taube et al., 1990). These cells respond to particular orientations of the animal's head irrespective of its location in the environment and fire only when the animal faces some particular direction (over an approximately 90 degree range) in the horizontal plane. These cells thus appear to function as some sort of an *in built compass*.

A number of experiments have served to identify crucial properties of place cells and head-direction cells (see McNaughton et al., (1996) for a detailed exposition of the properties). In brief, these cells have been found to respond to sensory as well as path-integration inputs. Further, places appear to be represented by an *ensemble* of cell firings, with the cells being active in *multiple* environments and often at *multiple places* in the *same* environment. The firing of these cells is conserved in darkness, provided the animal is first allowed to orient itself under lighted conditions. Further, any restraint on active motion ceases the cell firings.

## HIPPOCAMPAL COGNITIVE MAP

Based on extensive experimental evidence it has been suggested that rodents learn *cognitive maps* of their environments (Tolman, 1948). These cognitive maps are metric in nature, i.e., the spatial representation encodes distances and directions between the environmental cues. Against this background, (O'Keefe and Nadel, 1978) forwarded the *locale system hypothesis* (based on an immense corpus of neurophysiological and behavioral data) suggesting that the cognitive map resides in the hippocampus and that the place cells use sensory and dead-reckoning inputs to encode the metric map. A computational implementation of this locale system hypothesis of hippocampal spatial learning has been developed which allows the animal to learn its environment in terms of *distinct places*, with the *center* of each place also being labeled with a *metric position estimate* derived from dead-reckoning. A detailed treatment of this model can be found in (Balakrishnan et al., 1997); here we will only present a brief summary.

As the animal explores its environment the model creates new EC units that respond to landmarks located at particular positions *relative* to the animal. Concurrent activity of EC units defines a *place* and CA3 place cells are created to represent them. These sensory input-driven CA3 place cells are then associated with position estimates derived from the dead-reckoning system to produce place firings in the CA1 layer. Thus, the firing of CA1 cells is dependent on two information streams: sensory inputs from CA3 and the animal's dead-reckoning position estimates. The dead-reckoning input is used to learn the center of the place in terms of metric coordinates.

When the animal revisits familiar places, incoming sensory inputs activate a place code in the CA3 layer that corresponds to a familiar place. Since multiple places in the environment can produce the same sensory input (called *perceptual aliasing* in robotics), the CA1 layer uses dead-reckoning estimates to disambiguate between such places and produces a *unique* place code that corresponds to the current place. The hippocampal system then performs spatial localization by *matching* the *predicted* position of the animal (its current dead-reckoning estimate) with the *observed* position of the place field center (dead-reckoning estimate previously associated with the activated CA1 place code). Based on this match, the dead-reckoning estimate as well as the place field center are updated as shown in Figure 2.

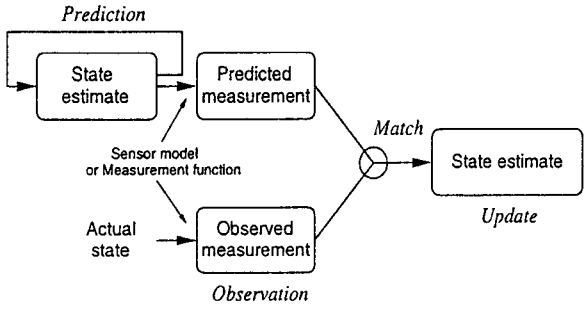



Figure 2: A schematic of hippocampal localization.

Thus, not only does the hippocampal model learn a metric cognitive map of the environment, but it also permits the metric estimates to be *updated* when the animal revisits familiar places. Further details of the model may be found in (Balakrishnan et al., 1997).

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

## Hippocampal Kalman Filtering

In the locale system hypothesis of hippocampal spatial learning, information is integrated from two streams: the sensory inputs and the dead-reckoning system (O'Keefe and Nadel, 1978). It should be noted that information provided by both these streams is uncertain because of errors in object recognition, distance estimation, and path integration. For instance, the firing of place cells and head-direction cells drift in darkness, suggesting errors in path-integration. Thus, in order for the hippocampus to perform robust spatial localization using these uncertain information sources, it must have adequate mechanisms for handling uncertain information sources. Although several hippocampal models of spatial learning have been proposed, including some that are closely related to the model described above, *none* of the models are capable of *explicitly* handling such uncertainties.

As with animals, mobile robots too have to deal with uncertainties in sensing and action. This has led to many probabilistic localization approaches for mobile robots. One such localization tool is the *Kalman filter* (KF) (Gelb, 1974) (or some extension or generalization of it), which allows the robot to build and maintain a *stochastic spatial map*, propagate sensory and motion uncertainties, and localize in *optimal* ways (Ayache and Faugeras, 1987). A schematic for a KF is shown in Figure 3.

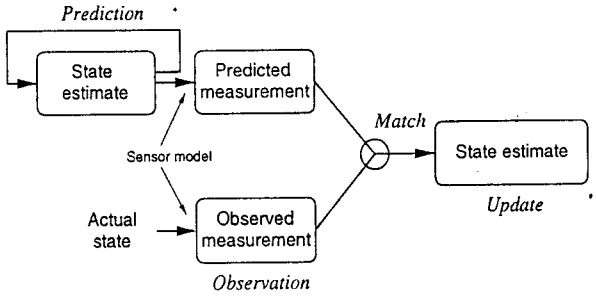



Figure 3: A schematic of Kalman filtering.

As can be observed from Figures 2 and 3, the computational model of hippocampal function and KF both embody the same *predict-observe-match-update* principle. Further, KF provides a framework for performing *stochastically optimal* updates even in the presence of prediction and observation errors. Based on the similarities between the two, Balakrishnan et al. (1997) developed a KF framework for uncertain information fusion in the hippocampal localization model described above. In this framework, KF helps the animal in maintaining and updating an estimate of its own position as well as the estimates of the place field centers. These estimates, referred to as the *state*, include:

$$\mathbf{x}_k = [x_{0,k}, x_1, \ldots, x_n]^T$$

where $x_{0,k}$ denotes the position of the animal at time instant $k$, $x_i$ denotes the center of place field $i$, and $n$ is the number of distinct places that have been visited by the animal. Without loss of generality, these position estimates are assumed to be specified in 2D Cartesian coordinates, i.e., $x_i = (x_{i_x}, x_{i_y})$. The animal also computes and updates the covariance matrix associated with this state vector, denoted by $\mathbf{P}_k$, which is given by:

$$\mathbf{P}_k = \begin{pmatrix} \mathbf{C}_{00} & \mathbf{C}_{01} & .. & \mathbf{C}_{0n} \\ \mathbf{C}_{10} & \mathbf{C}_{11} & .. & \mathbf{C}_{1n} \\ . & . & . & . \\ . & . & . & . \\ \mathbf{C}_{n0} & \mathbf{C}_{n1} & .. & \mathbf{C}_{nn} \end{pmatrix}$$

where

$$\mathbf{C}_{ij} = \begin{pmatrix} C_{i_x j_x} & C_{i_y j_x} \\ C_{j_x i_x} & C_{j_y i_y} \end{pmatrix}$$

denotes the covariance between the 2D Cartesian representations of the state elements $x_i = (x_{i_x}, x_{i_y})$ and $x_j = (x_{j_x}, x_{j_y})$.

When a new place is visited, the state vector is augmented by the center of this new place and the state estimate and its covariance matrix are modified accordingly. If the animal motions are assumed to be *linear* and the *measurement function* in Figure 3 is also a *linear* function of the state, a framework for *hippocampal Kalman filtering* can be developed that updates the place field centers and the animal's position estimate in *stochastically optimal* ways. These details can be found in Balakrishnan et al., (1997).

## Frame Merging

The procedure described above allows the *animat*[2] to learn a metric place map. However, it does not allow the animat to learn and integrate *independent local* metric maps corresponding to different regions of the environment, or to learn and integrate a *new* map into an existing one. We have developed an extension of the computational model described above that permits the animat to learn separate place maps in different *frames* and to *merge* frames together in a well-defined manner.

Suppose the animat has learned a place map, labeling the places with metric position estimates derived from its dead-reckoning system. Let us refer to this frame as $f_{old}$. Suppose the animat is now reintroduced at another place. The animat stores away $f_{old}$ in its memory, and begins a new frame $f_{new}$ at the point of reintroduction. It also resets its dead-reckoning estimates to zero, thereby making the point of reintroduction the origin of its new dead-reckoning frame. Now it proceeds as before, learning places and creating EC, CA3, and CA1 cells using the algorithms detailed in (Balakrishnan et al., 1997). At each step it also checks to see if sensory inputs excite CA1 cells residing in $f_{old}$. If this happens, the animat is at a place it has seen earlier in the older frame ($f_{old}$). It then *merges* the two frames, labeling the places in the two frames in a uniform coordinate system as follows.

Suppose CA1 unit $c$ fires in $f_{new}$ and $m$ fires in $f_{old}$. The goal is to merge $f_{old}$ into $f_{new}$. We do this by changing the position labels of all CA1 units in $f_{old}$ to equivalent labels in $f_{new}$. Let $\hat{\mathbf{x}}_c^{f_{new}}$ and $\hat{\mathbf{x}}_m^{f_{old}}$ denote the estimated center of the animat's current place in the two frames $f_{new}$ and $f_{old}$. Since $\hat{\mathbf{x}}_c^{f_{new}}$ and $\hat{\mathbf{x}}_m^{f_{old}}$ correspond to the center of the same

[2] A simulated animal

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

place field, albeit in different frames, $\Delta x = x_m - x_c$ denotes the amount by which frame $f_{old}$ has to be transformed to coincide with $f_{new}$. Assuming a metric coordinate representation, we can update the place field centers of $f_{old}$ to $f_{new}$ via the transformation:

$$\hat{x}_i^{f_{new}} = \hat{x}_i^{f_{old}} - \Delta x \qquad \forall i \in f_{old} \qquad (1)$$

The covariances between units in $f_{old}$ and $f_{new}$ can be updated using the following expressions (details of the derivations can be found in (Balakrishnan et al., 1998)):
**Case I**: i and j were both units in $f_{old}$

$$C_{ij}^{f_{new}} = C_{ij}^{f_{old}} - C_{mj}^{f_{old}} - C_{im}^{f_{old}} + C_{mm}^{f_{old}} + C_{cc}^{f_{new}}$$

**Case II**: i was a unit in $f_{new}$ and j was in $f_{old}$

$$C_{ij}^{f_{new}} = C_{ic}^{f_{new}}$$

where $C_{ij}^{f}$ refers to the covariance between units $i$ and $j$ in a particular frame $f$.

Once these updates have been carried out, frame $f_{old}$ has been effectively merged into $f_{new}$. However, it must be borne in mind that this frame merge procedure is currently blind to *perceptual aliasing*. Consequently, the animat localizes to the first place that sensorily matches a place it has seen before. If multiple places in the environment produce similar sensory inputs, this procedure will lead to localization problems.

### Goal Representation

Since the computational model of (Balakrishnan et al., 1997) allows the animat to learn places in a metric framework, goals encountered by the animat can also be remembered in terms of their metric positions. Thus, when an animat visits a goal location, it computes an estimate of the goal position based on its current dead-reckoning estimate. However, since dead-reckoning is error prone, the remembered (or computed) position of the goal is also erroneous. We need a procedure that explicitly handles this uncertainty, much like the KF for updating place field centers. We have developed a mechanism that maintains and updates the goal location estimate and its variance using the expressions in equation 2

$$\hat{x}_G = \frac{\sigma_0^2}{\sigma_0^2 + \sigma_G^2}\hat{x}_G + \frac{\sigma_G^2}{\sigma_0^2 + \sigma_G^2}\hat{x}_0 \qquad (2)$$

$$\sigma_G^2 = \frac{\sigma_G^2 . \sigma_0^2}{\sigma_0^2 + \sigma_G^2}$$

where $\hat{x}_G$ is the estimated goal position and $\sigma_G^2$ its variance, $\hat{x}_0$ is the current dead-reckoning estimate with associated variance $\sigma_0^2$. It can be shown that this update expression *minimizes* the variance of the goal position estimate (Balakrishnan et al., 1998). These update expressions are applied each time the animat reaches the goal. If the animat has never encountered the goal before, the goal variance $\sigma_G^2$ is set to $\infty$. Thus, when the animat encounters the goal for the first time, the above expressions result in the setting of the goal position estimate to the value of the dead-reckoning estimates.

The animats in our experiments navigate to goal locations through two means. If the goal is visible, the animats directly move towards the goal (*goal approaching*). However, if the goal is not visible but the animat has previously visited the goal location and thus remembers its position, it simply moves in a fashion that reduces the discrepancy between its current position estimate and the remembered position of the goal. We call this the *goal seeking* behavior. The goal seek behavior takes the animat along the shortest path to the goal. It is possible that the direct short-cut to the goal is blocked or has obstacles that the animat must then avoid. However, for the purposes of the experiments described in this paper the environments are assumed to be largely open and obstacle-free.

## SIMULATION DETAILS

In this paper we attempt to simulate the behavioral experiments of Collett et al.(1986) using the computational model of hippocampal spatial learning described earlier. The experimental setup of Collett et al. consisted of a circular arena of diameter 3.5 meters placed inside a light-tight black painted room. Gerbils were trained to locate a sunflower seed placed in different geometric relationships to a set of visible landmarks. The floor of the arena was covered with black painted granite chips to prevent the gerbil from spotting the seed until it was very close to it (Collett et al., 1986).

In our simulations, we used a square arena of size $20 \times 20$ units. The walls of the arena were assumed to be impenetrable and devoid of any distinguishing sensory stimuli. This is in keeping with the original experiment in which the walls were in complete darkness and presumably not visible to the animal. The landmarks, on the other hand, were assumed to be visible to the animat from all points in the arena. The animats could also estimate landmark positions relative to themselves, but this estimate was assumed to be corrupted by a zero-mean Gaussian sensing error with standard deviation $\sigma_S = 0.01$ units per unit distance. Sensory inputs obtained in this fashion were used to generate the activations of the EC layer as well as the place firings of the CA3 and CA1 layers, using the algorithms described in (Balakrishnan et al., 1997). The animat motions were also error-prone, with motion error modeled by zero-mean Gaussians with $\sigma_M = 0.5$. The animats possessed means for fairly accurate dead-reckoning with errors being modeled as zero-mean Gaussians with $\sigma_D = 0.05$ units. Animats could approach a visible goal and were said to have consumed the goal if they entered a circular region of radius 0.33 units around it.

The experiments of Collett et al. were simulated by first setting up the arena with the landmark(s) in the appropriate positions. The animat was then introduced into the arena at a random position and allowed to perform 500 steps of *sensing, processing*, and *moving*. In this mode the animats learned places by inducting EC, CA3, and CA1 units in appropriate ways, and updating the position estimates using the Kalman filtering mechanism described in (Balakrishnan et al., 1997). If the animat happened to see the goal during these sessions, it was made to approach and consume it. This constituted one *training trial*. Once a trial

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

was complete, the animat was removed from the environment and reintroduced at another random position for the next trial. Each animat was subjected to five such training trials. In each trial the animat learned places in a new *frame* and merged frames if they lead to the same place. The firing threshold of CA3 units (`CA3Threshold`), which signals place recognition based on sensory inputs, was set to 0.75 during training.

Once training was complete, the animat was subjected to ten *testing trials* in which the landmarks in the arena were manipulated in specific ways and, importantly, the goal was absent. During these tests the animat was released at predetermined positions in the arena with its dead-reckoning variance set to $\infty$. Further, spatial learning was turned off in these animats and they were only capable of localizing. The animats had a maximum of 150 steps within which to localize by visiting a familiar place. Un-localized animats were removed from the environment, with that testing trial being dubbed a failure, and the process continued with the next testing trial. During testing, `CA3Threshold` was lowered to 0.25 to enable the animats to localize even if the landmark arrangements had been changed in critical ways. A localized animat was allowed a maximum of 300 timesteps to navigate to the goal using the goal seek behavior described earlier. Since the goals were absent during testing, the animats searched in the region of the remembered goal location. If the animat reached a circular region of radius 0.5 units around the predicted goal location, it was allowed to spend 25 timesteps searching for the goal. After this, the variance of the position estimate of the animat was once again set to $\infty$ and the animat was permitted to *re-localize* to enable it to correct its localization if it had wrongly localized earlier. This had interesting behavioral consequences as will be explained later.

For the training as well as testing trials, the trajectories followed by the animats were recorded. Also, the 20 $\times$ 20 arena was decomposed into cells of size 0.33 $\times$ 0.33 and a count of the amount of time spent by the animats in each cell was kept. These statistics for training and testing were computed for *five different animats*. The cell with the largest value (amount of time spent by the five animats) was used to normalize the values in the other cells, and was plotted in the form of a search histogram. Thus, darker cells in the histogram indicate that the animats spent more time in that region of the arena compared to the regions corresponding to the lighter ones. It must be mentioned that the arena size, the histogram cell size, as well as the goal visibility range were roughly chosen to correspond to actual values used by Collett et al.

## EXPERIMENTS AND RESULTS

In this section we present simulations of Collett et al.'s behavioral experiments, using the computational model of spatial learning and localization detailed in (Balakrishnan et al., 1997; Balakrishnan et al., 1998).

### One Landmark Experiment

In this experiment, Collett et al. placed the seed at a constant distance and orientation from a single landmark and trained gerbils to reliably approach the goal position. They found that well-trained gerbils run directly to the seed when introduced into the environment. Further, in testing trials the gerbils were found to concentrate their search efforts at the expected location of the seed even though the seed was absent (Figure 1 in (Collett et al., 1986)). In our simulation of this experiment, the goal location was 4 units to the south of a single landmark, as shown by the search distribution concentrated in that region (Figure 4, Left). In these figures, filled squares represent landmarks. This compares rather well with the observations of (Collett et al., 1986).

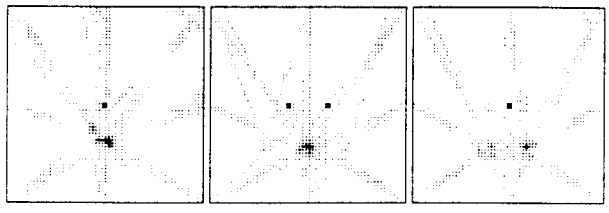


Figure 4: Left: One landmark experiment. Middle: Two landmarks experiment. Right: Two landmarks experiment with one landmark removed.

### Two Landmark Experiments

In the next set of experiments, Collett *et al.* trained gerbils to locate a sunflower seed placed to the south of a line connecting two identical landmarks. In this case, the goal was equidistant from the two landmarks. In our simulations, the goal was placed 4 units to the south of the line connecting two landmarks placed 4 units apart. As seen in Figure 4 (Middle), the search effort of the animats is reliably concentrated in a region rather close to the position of the goal in the training trials. This figure compares well with Figure 7b in (Collett et al., 1986).

Collett *et al.* also trained gerbils on the two landmark task and tested them with one landmark removed. They found that the gerbils searched on both sides of the sole landmark apparently matching the landmark either to the left or the right landmark of the original configuration (Figure 7c in (Collett et al., 1986). Our animats demonstrated a similar behavior as seen in Figure 4 (Right).

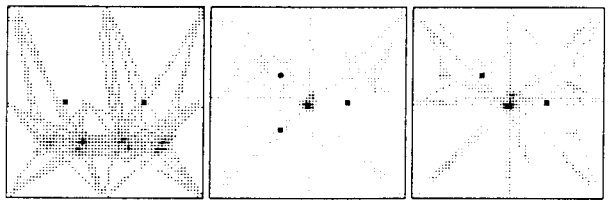


Figure 5: Left: Two landmarks experiment with landmark distance doubled. Middle: Three landmarks experiment. Right: Three landmarks with one removed.

Also, when the gerbils were trained with two landmarks and tested with the landmark distance doubled, Collett *et al.* found that the gerbils searched predominantly at the two interior locations each at the correct distance and orientation from one of the landmarks (Figure 7d). We observed similar search histograms in our experiments,

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

as seen in Figure 5 (Left). We also found that *all* the animats that first searched at the outer locations later searched in one of the interior two locations, when asked to relocalize. Further, *most* animats that first searched at the interior locations, *did not* search at the outer locations upon relocalization.

### Three Landmark Experiments

In this experiment, three identical landmarks were arranged to form the vertices of an equilateral triangle with the goal located at the centroid of the triangle. Animats trained in this environment produce search histograms concentrated reliably at the correct position of the goal, i.e., the centroid of the triangle as shown in Figure 5 (Middle). This compares favorably with Figure 6b in (Collett et al., 1986).

Collett *et al.* also trained the gerbils on the three landmark task and tested them in environments with one or two of the landmarks removed. With one landmark removed they found that the gerbils searched at a location at the correct distance and orientation from the two remaining landmarks (Figure 6c). As can be seen from Figure 5 (Right), our animats demonstrate largely similar search behaviors.

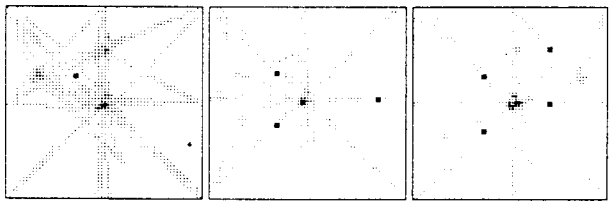


Figure 6: Left: Three landmarks with two removed. Middle: Three landmarks with one distance doubled. Right: Three landmarks with an extra landmark added.

With two of the three landmarks removed, Collett *et al.* found that the gerbils distributed their search time between three sites, one for each of the three possible matches of the sole landmark (Figure 6d). This can be compared directly with our simulation results in Figure 6 (Left). Similarly, when the gerbils were trained on the three landmark task but tested with one landmark distance doubled they were found to search at a goal location at the correct distance and bearing from the two unmoved landmarks (Figure 8 in (Collett et al., 1986)). Our animats display similar behaviors (Figure 6 (Middle)).

When gerbils were trained on the three landmark task, but tested in an environment with an additional landmark placed so as to create another equilateral triangle with a different orientation, Collett *et al.* found that the gerbils reliably searched at the goal location within the correctly oriented triangle. Our simulation of this experiment produced similar results as shown in Figure 6 (Right).

## DISCUSSION

In this paper we have extended the spatial learning and localization model developed in (Balakrishnan et al., 1997) along several significant directions. We have developed mechanisms to learn local place maps in *disjoint* frames, and to merge these frames to produce global place maps. We have also incorporated a mechanism for learning and remembering goals in terms of their metric positions, with an associated mechanism for updating goal positions in a stochastically consistent manner. With these additions, animats can not only learn maps of environments in a piecemeal fashion but also learn and reliably navigate to goals in the environment.

This allowed us to simulate the behavioral experiments of (Collett et al., 1986). The primary goal was to test whether our computational model of hippocampal spatial learning and localization was capable of explaining their behavioral data with gerbils. We simulated a number of their experiments and the search histograms generated by our animats were found to be very similar to those produced by the gerbils in their experiments. This is especially interesting because our animats did not remember goals in terms of *independent vectors* to individual landmarks, as suggested by (Collett et al., 1986). Our results indicate that if goals are remembered in terms of metric position estimates, localization errors are enough to explain the search distributions of the gerbils observed in environments with landmark configurations changed.

To the best of our knowledge, the only computational simulation of the (Collett et al., 1986) experiments, apart from the work presented in this paper, is that of (Redish and Touretzky, 1996). In their simulations, the animat was placed at different random positions in the arena and was given its position relative to the goal (which was assumed to coincide with the origin). The animat then created place cells using a combination of this position estimate and sensory inputs from the visible landmarks. *Ego-centric* angles between landmarks were also encoded in the place cells, which allowed the animat to initialize its head-direction if it happened to be disoriented. In test trials they introduced the animat at a random position and allowed it to localize, i.e., the animats performed head-direction and position estimate resets. Once the animat had localized, it could *predict* the goal location which was simply the origin of the coordinate frame with respect to its current localized position. They repeated this process a number times and calculated a histogram of predicted goal positions (Redish and Touretzky, 1996).

Our computational model of hippocampal spatial learning is closely related to that of (Redish and Touretzky, 1996) (referred to hereafter as the RT model) since both models are based on the *cognitive map* concept of (Tolman, 1948) and its implicated substrate in the hippocampus (O'Keefe and Nadel, 1978). Further, both these models make use of the *locale system* hypothesis of (O'Keefe and Nadel, 1978) with places being learned using a combination of sensory inputs and dead-reckoning information. Finally, both simulations represent goals in terms of metric position estimates derived from dead-reckoning.

Despite these similarities, there are some significant differences between the two models and the behavioral results generated by them. Our model assumes that errors exist in the sensory and dead-reckoning input streams and our computational framework explicitly addresses the issue of information fusion from erroneous (or uncertain)

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

sources. By formulating the the place learning and localization problem within the framework of Kalman filtering, we have been able to derive *update expressions* that can be proven to be *stochastically optimal*. The RT model incorporates a mechanism for initializing the head direction. However, doing so makes the place cells *directional*, which appears to be at odds with experimental results that suggest the *non-directionality* of the CA3 and CA1 pyramidal cell firings. Our model assumes that the place cells are non-directional and this requires that the animats have reliable head-direction information, i.e., we assume that the animals have not been disoriented. Further, animals learn and remember multiple goal locations, and plan and execute multi-destination routes. Extending our model to handle learning and representation of multiple goal locations is rather straightforward. However, it is not clear how one could represent multiple goals in the RT model considering that goals in their model correspond to the origin of the dead-reckoning system. Finally, animats in our simulations were capable of actually moving in their environment, whereas the animats used in the RT simulations do not move. Consequently, the histograms reported in (Redish and Touretzky, 1996) correspond to *predictions* of the goal position rather than the time spent by the animat in different regions of the environment. Thus, a dark histogram cell that is far from the goal in the RT model implies that the animat has a completely wrong estimate of the goal position and hence a completely wrong localization, while a similar cell in the histograms of Collett et al. simply means that the animal spent some time in that region localizing (or moving slowly on its way to the goal), and does not necessarily imply that the animal's localization or its prediction of the goal position is wrong. Since the animats in our simulations were capable of navigating, the search histograms generated in our experiments correspond more closely to those reported by Collett et al. (1986).

### Other Robot Localization Approaches

Owing to the Kalman filtering framework, our computational model of hippocampal spatial learning is directly related to KF approaches for robot localization (Crowley, 1995; Leonard and Durrant-Whyte, 1992). However, these KF based approaches require a *sensor model* of the environment (as shown in Figure 3) and often run into *matching problems* in environments with multiple identical landmarks and limited sensor ranges. The hippocampal model, on the other hand, provides a *place-based* extension of KF and easily addresses these problems (Balakrishnan et al., 1997). A number of robot localization approaches based on *cognitive mapping* theories (or *multi-level space representations*) have also been developed (Levitt and Lawton, 1990; Kuipers and Byun, 1991; Kortenkamp, 1993). Although closely related to the hippocampal spatial learning model, they are not formulated to computationally characterize a specific brain region and differ in this regard. Finally, a number of *neurobiological models* of robot navigation have been developed (Mataric, 1992; Bachelder and Waxman, 1994; Recce and Harris, 1996). However, these models deal with *topological* space representations (not metric ones), and are thus at discord with the *cognitive map* theory of (Tolman, 1948) and the *locale hypothesis* of (O'Keefe and Nadel, 1978). These differences are treated at length in (Balakrishnan et al., 1997).

### Future Work

As we mentioned earlier, our computational model assumes that the animat has an accurate head-direction estimate. This may not be the case if the animal has been disoriented. We are currently exploring the possibility of such a head-direction reset mechanism being implemented by place cells in the *subiculum* with the correction being performed by the *head-direction* cells in the *post-subicular region*. We have also developed a method to incorporate multiple goal locations in the model (Balakrishnan et al., 1998).

Given the fact that Kalman filter based models of place learning and localization satisfactorily reproduce an interesting collection of results from behavioral experiments in animals, it is natural to ask: *Can the hippocampus perform the Kalman filter computations? If so, how?* Some suggestions have been forwarded for the neural basis of these computations in the hippocampus, including the role of CA3 *recurrent collaterals* in the propagation and update of estimates and covariances of the places, *sharp waves* in the consolidation of position and covariance estimates, and the CA1 region in the computation of matrix inversions required for KF (Balakrishnan et al., 1997). These issues remain to be explored and explained, both through computational modeling efforts of neuro-physiological and behavioral phenomena, and through biological studies in living, behaving animals.

### ACKNOWLEDGMENTS

Olivier Bousquet's contributions to the theory of Hippocampal Kalman filtering is gratefully acknowledged. Karthik Balakrishnan and Rushi Bhatt are supported by an IBM Cooperative Fellowship and a Neuroscience Graduate fellowship from Iowa State University respectively. This research was partially supported by grants from the National Science Foundation (IRI-9409580) and the John Deere Foundation to Vasant Honavar.

### References

Ayache, N. and Faugeras, O. (1987). Maintaining representation of the environment of a mobile robot. In *Proceedings of the International Symposium on Robotics Research*, Santa Cruz, California, USA.

Bachelder, I. and Waxman, A. (1994). Mobile robot visual mapping and localization: A view-based neurocomputational architecture that emulates hippocampal place learning. *Neural Networks*, 7:1083–1099.

Balakrishnan, K., Bhatt, R., and Honavar, V. (1998). Spatial learning in the rodent hippocampus: A computational model and some behavioral results. (in preparation).

Balakrishnan, K., Bousquet, O., and Honavar, V. (1997). Spatial learning and localization in animals: A computational model and its implications for mobile robots. Technical Report CS TR 97-20, Department of Computer

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

Science, Iowa State University, Ames, IA 50011. (To appear in Adaptive Behavior).

Churchland, P. and Sejnowski, T. (1992). *The Computational Brain*. MIT Press/A Bradford Book, Cambridge, MA.

Collett, T., Cartwright, B., and Smith, B. (1986). Landmark learning and visuo-spatial memories in gerbils. *Journal of Neurophysiology A*, 158:835–851.

Crowley, J. (1995). Mathematical foundations of navigation and perception for an autonomous mobile robot. In *Proceedings of the International Workshop on Reasoning with Uncertainty in Robotics*, pages 9–51. Springer-Verlag.

Gallistel, C. (1990). *The Organization of Learning*. Bradford-MIT Press, Cambridge, MA.

Gelb, A. (1974). *Applied Optimal Estimation*. MIT Press.

Kortenkamp, D. (1993). *Cognitive Maps for Mobile Robots: A Representation for Mapping and Navigation*. PhD thesis, University of Michigan, Electrical Engineering and Computer Science Department.

Kuipers, B. and Byun, Y.-T. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8.

Leonard, J. and Durrant-Whyte, H. (1992). Dynamic map building for and autonomous mobile robot. *International Journal of Robotics Research*, 11(4).

Levitt, T. and Lawton, D. (1990). Qualitative navigation for mobile robots. *Artificial Intelligence*, 44(3):305–360.

Mataric, M. (1992). Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, 8(3).

McNaughton, B., Barnes, C., Gerrard, J., Gothard, K., Jung, M., Knierim, J., Kudrimoti, H., Qin, Y., Skaggs, W., Suster, M., and Weaver, K. (1996). Deciphering the hippocampal polyglot: the hippocampus as a path-integration system. *The Journal of Experimental Biology*, 199(1):173–185.

O'Keefe, J. and Dostrovsky, J. (1971). The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely moving rat. *Brain Research*, 34:171–175.

O'Keefe, J. and Nadel, L. (1978). *The Hippocampus as a Cognitive Map*. Oxford:Clarendon Press.

Recce, M. and Harris, K. (1996). Memory for places: A navigational model in support of marr's theory of hippocampal function. *Hippocampus*, 6:735–748.

Redish, D. and Touretzky, D. (1996). Navigating with landmarks: Computing goal locations from place codes. In Ikeuchi, K. and Veloso, M., editors, *Symbolic Visual Learning*. Oxford University Press.

Schone, H. (1984). *Spatial Orientation: The Spatial Control of Behavior in Animals and Man*. Princeton University Press, Princeton, NJ.

Taube, J., Muller, R., and Ranck, J. (1990). Head direction cells recorded from the postsubiculum in freely moving rats: I. description and quantitative analysis. *Journal of Neuroscience*, 10:420–435.

Tolman, E. (1948). Cognitive maps in rats and men. *Psychological Review*, 55:189–208.

Trullier, O., Wiener, S., Berthoz, A., and Meyer, J.-A. (1997). Biologically-based artificial navigation systems: Review and prospects. *Progress in Neurobiology*, 51:483–544.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

# A Connectionist Model of Perceptual Field Dynamics in Homogeneous Stimulus Areas

**Eliano Pessa** **Maria Pietronilla Penna**

Dipartimento di Psicologia
Università di Roma "La Sapienza"
Via dei Marsi, 78 I-00185 Roma, Italy
+39 6 49917631
pessa@axcasp.caspur.it

ABSTRACT

We present a connectionist architecture to model perceptual-motor processing of subjects engaged in the task of drawing a reproduction of a previously observed point on a white paper sheet. Such a task was designed to investigate the structure of perceptual field. Computer simulations showed a satisfactory agreement between model's forecastings and the experimental data obtained from an experiment performed on human subjects.

Keywords

Perceptual field, neural networks, spatial memory

## INTRODUCTION

Every model of human visual perception must take into account the evidence, given by Gestalt psychologists (see, e.g., Koffka, 1935), for global factors of wholistic nature. In most cases, however, the study of such factors was done only in a qualitative way. For this reason Gestalt psychologists were unable to build a formalized theoretical model of visual perceptual processing, designed to do quantitative forecastings of experimental data. Notwithstanding they introduced a fundamental concept, the one of *perceptual field*, viewed as similar to a vector field of forces acting within perceptual space. The lines of force of such a perceptual field should coincide, on one hand, with the paths followed in apparent movement phenomena, whereas, on the other hand, they should be orthogonal to perceived figural contours. A semi-quantitative investigation of perceptual field was undertaken already by Brown & Voth (1937), and by Orbison (1939). Such a task, however, requires to face strong theoretical and experimental difficulties in the case of nonhomogeneous stimulus areas, due to the great number of possible different situations, and of factors to be controlled.

In more recent times some authors (Stadler & Kruse, 1990; Stadler *et al.*, 1991) proposed an experimental procedure to investigate in a quantitative way the perceptual field structure in the case of homogeneous stimulus areas. Such a procedure was, in some way, inspired by Bartlett's early observation of the *wandering point* phenomenon (Bartlett, 1951). The experimental paradigm used to detect this latter can be described as follows. To a first subject is shown a white paper sheet on which, in a particular position, a black point was drawn. After the sheet has been removed, the subject is asked to draw, on a second white paper sheet, a point exactly in the same position in which was placed the point previously observed on the first sheet. After the first subject has drawn the point, the second paper sheet is shown to a second subject which, subsequently, is asked to do, on a third paper sheet, the same task as the first subject. Then the third paper sheet is shown to a thirs subject, and so on. In this way it is possible to obtain an ordered sequence of reproduced points, starting from the first presented one. Such a sequence, once transferred on a single sheet, evidences a wandering path, starting from the first point, which can be considered as a visualization of the line of force of perceptual field passing through this point.

Bartlett's idea appears as very appealing, mainly because the drawn point behaves like a probe, useful to investigate a perceptual field - the one created by sheet boundaries - in an homogeneous stimulation condition, without influencing in an essential way the field itself. However, such a procedure is practically unsuitable to study perceptual field structure in all locations belonging to paper sheet, as it would require a too great number of experimental subjects. A more easily implementable method is the one which makes use of a previous suitable sampling of locations, and, for each sampled location, ask the same subject to reproduce the point drawn in this location. In this way the data coming from a single subject let us obtain the *displacements* (of the reproduced point with respect to the observed point) associated to all sampled locations. These displacements, in turn, are proportional to the vector forces acting in each one of sampled points. We can thus obtain a quantitative representation of perceptual field structure and of its lines of force.

Such a representation, once obtained,, should be considered as a remarkable result, because it lets us characterize in a quantitative way the perceptual field postulated by Gestalt psychologists. However it raises an important problem, concerning the origin of observed perceptual field structure. Does this latter derive from some general Maximum (or Minimum) Principle, such as the one of goodness of form? Or it is a byproduct of sensorimotor processing, required by experimental task

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Nottingham, UK: Nottingham University Press. ISBN 1-897676-67-0

described above, and of principles ruling the operation of neural architectures involved? In order to support the evidence for the latter alternative we built a connectionist model designed to represent perceptual-motor processing by the experimental subject engaged in such a task, and to forecast the displacements observed in an experiment we performed, according to the paradigm presented above, on 10 subjects. Such a model was implemented through an architecture constituted by several different neural networks reciprocally interconnected, each one designed to do a particular task. Such a choice was dictated by the complexity of the experimental situation to be modelled. Namely this latter involves, first of all, an *acquisition system*, to grant for input of stimulation patterns, both of the sheet with the drawn point, and of the empty sheet where the point has to be reproduced, together with the instantaneous position of the point of pencil used to draw the reproduction. Moreover, we need a *spatial memory* , to store the information relative to the observed point, and a *motor system*, able to command hand motion in order to move pencil point up to the location where the point should be reproduced. Such a model was implemented through a computer program, and the outcomes of simulations we did were compared with the mean displacements observed in experiment with human subjects. We found a satisfactory agreement between computer simulation results and experimental data. Such an effect was essentially a consequence of general principles underlying the operation of single neural networks belonging to the architecture we described, rather than a consequence of *ad hoc* mechanisms already embodied within our model. Notwithstanding we feel that, in order to obtain a better agreement, some further experimental and theoretical problems remain to be solved.

Before undertaking a detailed illustration of proposed model, we will describe, in the second section, the experiment done on human subjects. The third section will contain a description of the component of our model we consider as the most critical one: the spatial memory. The other networks belonging to model architecture will be presented in a fourth section. The fifth section, then, will be devoted to a description of simulations done, and to a comparison between the results so obtained and experimental data coming from human subjects. The conclusion will be the object of sixth section.

## THE EXPERIMENT

The experiment was designed with a procedure similar to the one described, e.g., in Stadler *et al:* (1991), but with a systematic control of experimental variables.

### Subjects

The experiment was performed on 10 subjects, all students of Psychology, 5 males and 5 females, all with normal vision, or correct to normal.

### Stimuli

The stimuli were constituted by 609 A4-sized paper sheets, each one with a single point in a particular location. Each point had a circular form, whose radius was 1mm. The set of all locations filled a lattice with 29 rows and 21 columns, in which the distance between two neighbouring points, both along the horizontal and the vertical direction, was 1 cm.

### Procedure

To each subject were presented, once at time and each one for a duration of 1 s, all 609 stimulus sheets. The subject was sitting in a dark room, before a suitably built device, constituted by a box, with an upper opening to look inside and a lateral opening to insert subject's hand holding a pencil. Only the inner box was enlightened, so that the subject was forced to focus his/her attention only on stimulus sheet. After 1 s the sheet was removed through a suitable opening, existing in the box, by an experimenter , located in the dark , which substituted the stimulus sheet with an A4-sized blank sheet. The subject was asked to draw on this sheet a point exactly in the same location occuped by the point contained within the stimulus sheet presented before. The experimenter controlled that the initial position of subject's hand was always the same across all trials. Once the subject drew the reproduction of the observed stimulus point, the sheet was removed a new stimulation sheet was presented. The presentation order was randomized, and different from subject to subject. Each experimental session was preceded by a training period, to ensure the understanding of the task by the subject.

### Results

For each stimulus point and for each subject we measured the difference between the position of the reproduced point and the one of the stimulus point. Such a difference led us individuate the vector field acting in the location of stimulus point, and whence the tangent vector to the line of force of perceptual field passing through this point. Afterwards, we computed for each point a mean tangent vector, by averaging the results relative to the different subjects. The spatial distribution of mean tangent vectors thus obtained evidenced a regular trend (see Fig. 1). More precisely, the majority of straight lines individuated by each tangent vector were crossing in a small number of points, which Stadler *et al.* (1991) identified with the *attractors* of perceptual field. We found a strong evidence for the presence of two attractors located near the two corners on the upper part of the sheet (here the attribute "upper" refers to the observational point of view of experimental subject), in agreement with the findings by Stadler *et al.* On the contrary, we found only a weak evidence for the presence of other two attractors located near the two corners on the lower part of the sheet , differently from what found by the Authors quoted above.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

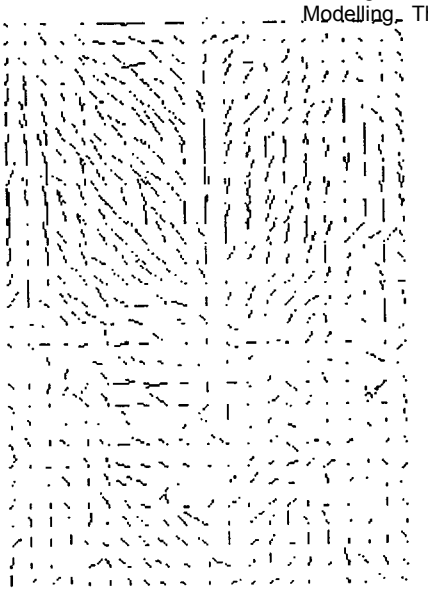


Fig.1
Observed distribution of tangent vectors to lines of force of perceptual field (averaged on all subjects).

## MODELLING SPATIAL MEMORY

The general architecture of the model we proposed, to describe perceptual-motor processing by a subject within the experiment described above, consists of the following interconnected neural networks: 1) a *retina* designed to receive input patterns, 2) a *spatial memory*, designed to process retinal output values, and to store the location of the point to be reproduced, 3) two *filtering networks*, designed to detect, respectively, the position of the point to be reproduced (determined as output of spatial memory), and the one of pencil point during reproduction, 4) a *motor network*, designed to give the right motor commands to the hand holding the pencil, as a function of the location of the point to be reproduced, and of the instantaneous position of pencil point.

The choice of implementing the above described subsystems through neural networks was dictated by the following reasons:

a) neural networks algorithms appear as more suitable to model, by using only a small number of rules of interaction between network units, behaviours such as the ones implied by perceptual or motor processing, which, stated in terms of traditional symbolic rule systems (such as the ones expressed through usual Predicate Calculus), would be too difficult to describe; such a circumstance is proved by fast diffusion, in recent times, of neural-network-based systems which do in a very efficient way artificial vision tasks, such as pattern recognition, visual scene analysis, object identification, and motor control tasks;

b) neural network structures appear as closer than usual symbolic rule systems to biological structures involved in visual and motor tasks, so that an interrelation between neurophysiological study and cognitive modelling becomes easier;

c) a parallel hardware implementation of neural network models can be faster than any serial processing of symbolic rules; such an argument would become crucial if our model would be used to command in real time an autonomous robot;

d) neural network algorithms appear as more robust, with respect to traditional symbolic rule systems, with respect to errors, variations of input patterns, variations of model parameter values.

We underline that the previous arguments, within this paper, have nothing to do with the traditional contraposition between symbolic and subsymbolic approach. Our neural network algorithms are symbolic, in the same way as usual symbolic rule systems. We feel only they are more convenient.

Within our model architecture the retina is modelled as a planar lattice of units, each one of which can be, at a given instant of time (henceforth we will suppose the time be discretized: $t = 1, 2, 3, ...$), in one of two states: activated or non-activated (corresponding to the activation levels 1 and 0, respectively). As regards neural network representing spatial memory many different modelling possibilities exist. They can be grouped within two fundamental categories: models which make use of *correlation matrices*, and are based on long-range connections, and models implemented through *cellular neural networks*, based on short-range connections. The prototype of models belonging to the first category is the celebrated Hopfield's associative memory model (Hopfield, 1982). There exist, however, memory models belonging to this category, but not directly implemented under the form of neural networks (see, e.g., Pike, 1984; Humphreys, Bain & Pike, 1989). A more recent neural network model of spatial memory of this type is the one proposed by Fukushima *et al.* (1997). A feature common to all these models is that spatial patterns are stored as contributions to a matrix of connection weights, each element of which captures the correlation between two elements of a pattern lying in different locations. This implies that the neural network implementing spatial memory must be constituted by a number of units equal to the one of pattern elements, with connection lines linking every pair of units, independently from the spatial distance between the elements corresponding to the units. The presence of such long-range connections not only is biologically implausible, but can give rise to strong interference effects between stored patterns, if we need to memorize more than one pattern. Such effects can worsen in a dramatic way network performance in recall phase. Moreover, this kind of neural networks appear as particularly suitable to memorize complex patterns, rather than very simple ones, as it is the case in our experiment, where the pattern is constituted by a single point.

The second category of neural network models of spatial memory, the one based on Cellular Neural Networks (CNN), derives from the fundamental paper by Chua & Yang (1988). Shortly, a CNN is constituted by a spatial lattice of units, each one endowed with a particular

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

activation fuction (of neural-like nature), and with a neighbourhood function, stating what units can send their output signals to the input lines of the unit itself. Each line connecting a given unit to its neighbouring units is characterized by a suitable connection weight. In practical applications CNN showed very good performances in artificial vision tasks relative to processing of simple spatial patterns. For this reason we choose this category of models to implement our spatial memory.

Within our model spatial memory was represented as a planar lattice whose dimensions and number of units were identical to the ones of the retina. Each spatial memory unit received input signals both from the retinal unit lying immediately under it, and from its neighbouring units within spatial memory. To this regard, we choose as neighbourhood of a given unit the classical 8-neighbourhood. This means that the neighbouring units of the unit with coordinates $(i, j)$ were the ones with coordinates $(i-1,j-1)$, $(i-1,j)$, $(i-1,j+1)$, $(i, j-1)$, $(i,j+1)$, $(i+1,j-1)$, $(i+1,j)$, $(i+1,j+1)$. If we denote by $x_{ij}(t)$ the activation level of the unit with coordinates $(i, j)$ at the time $t$, we can write the activation law we choose under the form:

$$(1) \quad x_{ij}(t+1) = a\, Q_{ij}\ tgh\,[P_{ij}(t)] - d\, x_{ij}(t),$$

where:

$$(2) \quad P_{ij}(t) = \Sigma_{r,s \in D}\, w_{ijrs}\, x_{rs}(t) + g\, x_{ij}(t) + I_{ij}(t) - s,$$

and $D$ denotes the neighbourhood of the unit $(i,j)$, $I_{ij}(t)$ is the input signal coming from the retina, whereas $s$ is a suitable threshold parameter. The quantities $a$, $d$, $g$ denote other parameters to be fixed by the experimenter. Moreover $Q_{ij}$ denotes a factor, depending on $x_{ij}(t)$, we varied, in order to investigate the effect of different choices of activation function on spatial memory performance. The forms of $Q_{ij}$ we used within our computer simulations were the following:

$$(3.a) \quad Q_{ij} = 1$$

$$(3.b) \quad Q_{ij} = 1 - x_{ij}(t)$$

$$(3.c) \quad Q_{ij} = 1 - /x_{ij}(t)/^{1/3}$$

$$(3.d) \quad Q_{ij} = 0.5 + x_{ij}(t) - 1.5\,[x_{ij}(t)]^3$$

The connection weights $w_{ijrs}$ associated to the lateral connections were varying with time according to a Hebb-like law of the form:

$$(4) \quad w_{ijrs}(t+1) = w_{ijrs}(t) + b\, M_{ijrs}\, x_{ij}(t)\, x_{rs}(t-1) + - d\, w_{ijrs}(t),$$

where $b$ and $d$ are other parameters, whereas $M_{ijrs}$ is another factor, depending on $x_{ij}(t)$ and $x_{ij}(t+1)$, which we modified in order to investigate the effect of different forms of the Hebbian law on spatial memory performance. The explicit forms of $M_{ijrs}$ we used within our computer simulations were the following:

$$(5.a) \quad M_{ijrs} = 1$$

$$(5.b) \quad M_{ijrs} = 1 - x_{ij}(t+1)\, x_{rs}(t)$$

$$(5.c) \quad M_{ijrs} = 1 - |\, x_{ij}(t+1)\, x_{rs}(t)|^{1/2}$$

In all simulations we performed the operation of spatial memory was observed for a number of time steps, previously fixed by the experimenter. At the end of this period, the activation levels of the units were filtered in the following way. First of all, we searched for the units whose activation level was the maximum one. Once found these units, their activation level was set to 1, whereas the activation level of all other units was set to 0. The units whose activation level was 1 were considered as representing what was stored within spatial memory. In other words, they specified the locations where should be placed the point to be reproduced. Of course, in all computer simulations, only one unit of spatial memory was characterized by an activation level equal to 1. We underline that, apart from specific choices of the factors $Q_{ij}$ and $M_{ijrs}$, the laws (1) and (4) are nothing but an expression of very general principles ruling neural activation and synaptic facilitation. Thus, the effects of spatial memory operation are to be viewed, essentially, as a consequence of the adoption of such principles.

## FILTERING AND MOTOR NETWORKS

When applying our general architecture to modelling human subjects performance in point reproduction task, we needed two filtering networks: one to detect the position of the point to be reproduced, as deriving from spatial memory processing, and another to detect the actual position of the point of the pencil used to draw the reproduction of the point itself. The former network received as input the pattern of activation levels of spatial memory, whereas the latter received as inputs the activation levels of retinal units in presence of the pencil. To do our simulations, we were forced to introduce a particular schematic representation of the pencil together with the hand holding it (as it is perceived by human subjects in the real laboratory experiment). More precisely, we choose to represent the hand through a rectangular array of 3x2 units, to which was attached, in the middle of the longest side, a line of 3 units representing the pencil. We underline that both choices of filtering networks, and of pencil representation, were dictated by the need for proving that a neural-like, and

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Nottingham (UK): Nottingham University Press. ISBN 1-897676-67-0

somehow realistic, representation of the information flow from spatial memory to motor network is possible. We acknowledge that other different representations would be possible without changing the operation principles of the neural architecture we proposed. However, we feel that the representation we adopted should be particularly suitable if we would implement our architecture through a particular hardware to be installed within an artificial device, such as a robot able to draw a reproduction of a visually observed pattern.

The filtering network receiving inputs from spatial memory was designed in such a way as to let survive only patterns consisting of a single activated unit. It was implemented through a 2-dimensional array of units (essentially a time-discrete CNN), of slightly greater dimensions with respect to the ones of the retina, in such a way as to include the representation of the hand holding the pencil. Each unit had a 8-neighbourhood and its activation potential was given by:

$$(6) \qquad P_{ij}(t) = x_{ij}(t) - \Sigma_{r,s \in D} w_{ijrs} x_{rs}(t) + I_{ij}(t)$$

where $D$ denotes the neighbourhood, all other symbols have the meaning defined in the previuous paragraph, and the connection weights $w_{ijrs}$ were all positive. The activation law had the form:

$$(7) \quad x_{ij}(t+1) = 1 \text{ if } P_{ij}(t) > 0.5\,, \text{ otherwise } x_{ij}(t+1) = 0.$$

In our simulations the operation time of this network was limited to only one time step.

As regards the second filtering network, the one receiving inputs from the retina and devoted to detect the position of the point of the pencil, we designed it in such a way as to let survive only the unit corresponding to the position of this latter. To this end, we adopted a 2-dimensional array of units, whose dimensions were identical to the ones of the first filtering network. Moreover, by taking again a 8-neighbourhood, we defined the activation potential as:

$$(8) \quad P_{ij}(t) = \Sigma_{r,s \in D} w_{ijrs} x_{rs}(t) - w_{OFF} x_{ij}(t) + I_{ij}(t),$$

where $w_{OFF}$ and $w_{ijrs}$ were all positive. In our simulations we choose all $w_{ijrs}$ values as identical to a common value $w_E$. The activation law had the form:

$$(9) \quad x_{ij}(t+1) = 1 \text{ if } 0 < P_{ij}(t) < (2w_E - w_{OFF})/2\,, \quad \text{otherwise } x_{ij}(t+1) = 0\,.$$

Also in this case the network operation lasted only for one time step.

As regards the motor network, it was designed to transform the knowledge of the actual position of the point to be reproduced, and of the point of the pencil, in a motor command able to induce a displacement of the hand, and whence of the point of the pencil. To this end the coordinates of the point to be reproduced (as deriving from the first filtering network), and of the point of the pencil (as deriving from the second filtering network), were first transformed into a binary form, by using 5 binary digits for each coordinate. Thus, all knowledge relative to the actual positions of the points quoted above was coded through a 20-components binary vector. This latter was used as input for a 3-layer Perceptron, whose output layer contained two units, one devoted to code the motor activation along the horizontal direction, and another to code this activation along the vertical direction. As the allowed motions along these directions could be both positive and negative, we choose, as activation function of the Perceptron units, the hyperbolic tangent one (with a suitable amplification factor).

The Perceptron was trained on a sample of input patterns, containing different relative positions of the point to be reproduced and of the point of the pencil. The desired output to each input pattern was obtained by putting the wanted motor activation along a given direction as directly proportional to the difference between the coordinates of the points quoted above along the same direction. Such a choice was made in conformity with neurophysiological findings (cfr. Schwartz & Georgopoulos, 1987), which evidenced a direct proportionality between the electrical activity of motor cortex neurons and perceived target distance. Of course, the proportionality factor had to be considered as a parameter to be chosen by the experimenter. The training was done through usual error-backpropagation rule. To avoid computational problems, the wanted outputs were divided by a suitable scale factor.

Once trained, the Perceptron was used as a simple input-output device, giving motor activation as a response to the 20-component binary input vector. To compute the effective displacement of the point of the pencil, we set the velocity component of this latter along a given direction as directly proportional to the motor activation along the same direction. Such a choice was made in conformity with recent neurophysiological findings on the correlation between motor cortex activation and limb movement velocity (cfr. Schwartz, 1992; 1993). Once computed the velocity components, the new coordinates $x_{new}$, $y_{new}$ of the point of the pencil were computed from the old ones $x_{old}$, $y_{old}$ through the relationships:

$$(10) \qquad x_{new} = x_{old} + (kv_x + v_{bx})*\Delta t\,,$$
$$y_{new} = y_{old} + (kv_y + v_{by})*\Delta t\,,$$

where $k$ is a proportionality factor, $v_x$ and $v_y$ are the components of the velocity computed as afunction of the corresponding motor activations, $v_{bx}$ and $v_{by}$ are the components of a "base" velocity, whereas $\Delta t$ is the time step amplitude. The introduction of a base velocity was made to represent the cerebellar modulation of limb movement, whereas the velocity obtained from motor

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Nottingham, UK: Nottingham University Press. ISBN 1-897676-67-0

activations represented the power impressed to the movement itself, in conformity with the hypothesis put forward by Flash & Hogan (1985).

## COMPUTER SIMULATIONS AND COMPARISON WITH DATA OBTAINED FROM HUMAN SUBJECTS

We used our model architecture to simulate the behaviour of human subjects in the experiment previously described. A number of preliminary trials suggested the following parameter values: $a = 0.5$, $d = 0.1$, $g = 0.3$, $s = 0$, $b = 0.4$, $d = 0.05$, $k = 1$, $\Delta t = 1$, $v_{bx} = 0.1$, $v_{by} = 0.1$, $w_E = 0.15$, $w_{OFF} = 0.1$. Moreover all non-zero connection weights $w_{ijrs}$ appearing in formula (6) were set to 0.8. The motor network was consisting of a 3-layer perceptron whose hidden layer had 2 units. The proportionality factor between motor activation and velocity was chosen as 0.6. The spatial memory processing lasted for 10 time steps after the disappearance of each stimulation pattern, and the hand movement had a limit duration of 10 time steps. We tested our model on the reproduction of the 609 points presented to human subjects. From the positions reached by the point of the pencil, as computed through our model, at the end of the movement period, we derived the tangent vectors through the same procedure used in the case of human subjects.

We did many different simulations with the same parameter values, corresponding to different combinations of choices relative to $Q_{ij}$ and $M_{ijrs}$. In all cases the results evidenced very clearly the presence of four attractors located near the corners of the sheet, two in the upper part and two in the lower part. As a quantitative measure of model performance we choose for each stimulus point, the euclidean distance between the position reached, within the model, by the point of pencil and the corresponding average position of the point reproduced by human subjects. We then computed the mean value $\delta$ of such a distance, averaged on all stimulus points. As other two measures of model performance we choose:

1) the Bravais-Pearson correlation coefficient $cy$ between the vertical components of the tangent vectors, obtained in our simulations, and the ones of mean tangent vectors, obtained from human subjects' data;

2) the Bravais-Pearson correlation coefficient $cx$ between the horizontal components of the tangent vectors, obtained in our simulations, and the ones of mean tangent vectors, obtained from human subjects' data.

The values of $\delta$, $cy$, and $cx$ obtained in correspondence to the different choices of $Q_{ij}$ and $M_{ijrs}$ are listed in the following (to shorten the exposition, every choice is indicated through the numbers of the corresponding formulae).

A) choice (3.a), (5.a):

$$\delta = 21 \ , \ cy = 0.34 \ , \ cx = 0.24$$

B) choice (3.a), (5.a):

$$\delta = 17 \ , \ cy = 0.43 \ , \ cx = 0.20$$

C) choice (3.a), (5.b):

$$\delta = 48 \ , \ cy = 0.16 \ , \ cx = 0.17$$

D) choice (3.c), (5.a):

$$\delta = 12 \ , \ cy = 0.58 \ , \ cx = -0.20$$

E) choice (3.c), (5.b):

$$\delta = 11 \ , \ cy = 0.60 \ , \ cx = -0.21$$

F) choice (3.d), (5.b):

$$\delta = 25 \ , \ cy = 0.64 \ , \ cx = 0.18$$

G) choice (3.d), (5.a):

$$\delta = 12 \ , \ cy = 0.62 \ , \ cx = 0.17$$

H) choice (3.d), (5.c):

$$\delta = 24 \ , \ cy = 0.64 \ , \ cx = 0.19$$

In order to have an idea of the meaning of these numbers, we remember that a value $\delta = 10$ means that the average distance from the points reproduced by our model and the ones reproduced by human beings is only of one lattice cell. We could thus hold that the results obtained from the choices D), E), G) evidence a very good agreement between our model behaviour and the one of human subjects. We should, however, take into account also the values of $cy$ and $cx$, which show a very strange trend. On one hand, namely, the correlations regarding vertical components evidence a very good agreement between our model and human data, chiefly in correspondence to the choices D), E), F), G), H). The choice G), then, seems to have realized the best compromise between a high value of $cy$ and a small value of $\delta$. On the other hand, the correlations regarding horizontal components appear as too small, in some cases even negative. The highest value was obtained in correspondence to the choice A), which, however, doesn't appear as particularly good, when we look at the values of $\delta$ and of $cy$. From simulation results it appears as evident that neither the choice of $Q_{ij}$ nor the one of $M_{ijrs}$, isolately considered, can improve the performance of our model. This latter depends on both choices. The best one appears to be G), but the improvement in performance on $cx$, without a great worsening on $cy$ and $\delta$, suggests that a good research strategy would be the one of investigating what happens by replacing in (5.c) the exponent 1/2 with smaller

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Nottingham (UK): Nottingham University Press. ISBN 1-897676-67-0

exponents. In any case, such a circumstance evidences how model's performance depends essentially on general form of laws such as (1) and (4), rather than on particular choices of factors such as $Q_{ij}$ and $M_{ijrs}$ . Another possible explanation of the results we obtained can be found in the existence of some bias which influenced the performance of subjects during the experiment. Namely the variance of their behaviours is very high. Moreover, a comparison we did between our simulations and the results obtained from single subjects showed, in some cases, an agreement better than the one evidenced by a comparison with average subject behaviours, whereas, in other cases, such an agreement was worse. However, only a careful repetition of the experiment with human subjects can tell us whether this is or not the reason for the observed trend of $cx$.

## CONCLUSION

The computer simulations so far done evidenced that our model was able to reproduce in a satisfactory way some qualitative (the presence of attractors) and quantitative features of subjects' performance in the point reproduction task. It is, thus, possible, to conclude that our model was able to reproduce some Gestalt-like properties of visual perception, owing essentially to a suitable choice of the dynamical laws underlying spatial memory operation. The usefulness of our proposal stems also from the fact that the neural network architecture we introduced is of modular nature, so that it becomes very easy to investigate the effects on model performance of different choices of the laws ruling the operation of each module. Besides, our model can be easily adapted to represent the cognitive processing of a subject engaged in other sensorimotor tasks, different from the one of point reproduction. A continual , and mutual, interaction between experimental and modelling activity is, however, needed in order that complex model architectures, such as the one we proposed, be useful to improve our knowledge about cognitive system.

## REFERENCES


Bartlett, F.C. (1951). *The Mind at Work and Play*. Allen and Unwin, London.

Brown, J.W., & Voth, A.C. (1937). The path of seen movement as a function of the vector-field.*American Journal of Psychology, 49*, 543-563.

Chua, L.O., & Yang, L. (1988). Cellular Neural Networks: Theory. *IEEE Transactions on Circuits and Systems, 35*, 1257-1272.

Flash, T., & Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neurophysiology*, 5, 1688-1703.

Fukushima, K., Yamaguchi, Y., & Okada, M. (1997). Neural network model of spatial memory: Associative recall of maps. *Neural Networks, 10*, 971-979.

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of U.S.A., 79*, 2554-2558.

Humphreys, M.S., Bain, J.D., & Pike, R. (1989). Different ways to cue a coherent memory system: A theory for episodic, semantic, and procedural tasks. *Psychological Review, 96*, 208-233.

Koffka, K. (1935). *Principles of Gestalt Psychology*. Harcourt & Brace, New York.

Orbison, W.D. (1939). Shape as a function of the vector-field. *American Journal of Psychology, 52*, 31-45.

Pike, R. (1984). Comparation of convolution and matrix distributed memory systems for associative recall and recognition. *Psychological Review, 91*, 281-294.

Schwartz, A. (1992). Motor cortical activity during drawing movements: single-unit activity during sinusoid tracing. *Journal of Neurophysiology, 68* 528-541.

Schwartz , A. (1993). Motor cortical activity during drawing movements: population representation during sinusoid tracing. *Journal of Neurophysiology, 70*, 28-36.

Schwartz, A., & Georgopoulos, A. (1987). Relations between the amplitude of 2-dimensional arm movements and single cell discharge in primate motor cortex. *Society of Neuroscience Abstracts, 13*, 244.

Stadler, M., & Kruse, P. (1990). The self-organization perspective in cognition research: historical remarks and new experimental approaches. In: H.Haken and M.Stadler (Eds.). *Synergetics of Cognition*, 32-52. Springer, Berlin, Heidelberg, New York.

Stadler, M., Richter, P.H., Pfaff, S., & Kruse, P.(1991). Attractors and perceptual field dynamics of homogeneous stimulus areas. *Psychological Research*, 53, 102-112.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

# Generating and Classifying Recall Images by Neurosymbolic Computation

**Ernesto Burattini, Massimo De Gregorio, Guglielmo Tamburrini**

Istituto di Cibernetica C.N.R.
I-80072 Arco Felice (NA), Italy
+39 81 853 4230
{ernb, massimo, gugt}@sole.cib.na.cnr.it

## ABSTRACT

The neurosymbolic hybrid system ARCS, which extends a classifier for certain kinds of visually presented objects, generates recall images it is then capable of classifying. The modules performing classification are exploited for imagery, too. In particular, each weightless neural discriminator has been modified so as to generate a non-crisp example of the class of simple visual features it was trained to detect; a symbolic process prescribes how to assemble more complex patterns from such non-crisp examples; both generated features and their compositions are correctly classified, even though the system was originally conceived for actual visual inputs only. These cognitively significant aspects of the hybrid system are examined in the framework of a more general discussion of neurosymbolic integration for cognitive modelling.

### Keywords

Hybrid systems, neurosymbolic integration, recall images, weightless neural systems, multidiscriminators, production rules

## INTRODUCTION

"If you were going to program a computer to mimic human imagery," Kosslyn (1995, p. 269) remarks "perhaps the most fundamental problem the program would have to solve is the *generation* of images." The neurosymbolic system ARCS (Arches Recall and Classification System) illustrates a way of solving this problem, relative to images of simple features and their compositions it is already capable of classifying.

The task domain contributes to highlighting the current interest of neurosymbolic integration for AI and cognitive modelling. The generation and classification of the selected, more complex patterns, which seem to elude a purely neural network approach, are naturally handled by means of the hybrid neurosymbolic system. The selected features are various line segments and angles; the more complex patterns represent various *portal shapes*. ARCS grew out of a hybrid classifier for portal shapes (De Gregorio, 1996), embedded into an architectural expert system for landmark building classification and preservation (Burattini, 1994).

There are aspects of the process by which ARCS generates recall images that are significant for cognitive modelling. Human image generation seems to involve a process for producing image parts, and another process for positioning individually activated parts in the image, so as to form more complex visual objects (Farah *et al.*, 1985, Kosslyn, 1994, Kosslyn, 1995, pp. 270-273). The same division of labour applies to the image generation mechanism of ARCS: one can distinguish between simple visual features and more complex patterns, and between two corresponding stages of image generation.

The first stage of image generation is carried out by the neural module of ARCS. This module is a weightless neural system formed by RAM-discriminators (Aleksander & Morton, 1990). The standard weightless discriminator model was slightly modified in order to make a wider repertoire of behaviours available (De Gregorio, 1997). In particular, such a modified discriminator can generate a grey-level, non-crisp example of the class of simple visual features it was trained to detect. The generated example differs from, but bears a precise relationship (spelled out in detail in the following sections) to *every* binary pattern of the corresponding training set. Roughly speaking, the grey intensity level of each non-white pixel in the example is proportional to the number of times that the corresponding memory locations of the discriminator were addressed by input training patterns. Since this relationship shows that each image in the training set contributes to forming the generated class example, a question that naturally arises is whether such recall images might be regarded as *typical* examples of the classes of visual patterns detectable by ARCS. From a computational perspective, it is worth pointing out that the first stage of image generation is carried out by neural nodes that are endowed with functionalities akin to (but not identical with) those of bidirectional associative memories (Kosko, 1988).

In the second stage of image generation, more complex objects are formed by properly assembling the elementary features together. This latter process is governed by the symbolic module of ARCS, a system of production rules determining the features to be assembled together and their categorical spatial relationships in the complex recall image.

In addition to generating recall images by a two-stage process, ARCS can inspect and classify them. It achieves this goal using a pre-existing hybrid classifier for actual visual inputs. Thus, classification of both mental images and actual visual inputs is taken care of by the same process. This is consistent with the widespread conviction (Damasio and Damasio, 1994, Finke, 1985, Kosslyn, 1994) that visual imagery exploits the mechanisms of visual perception; more generally, that mental imagery, in any of its modalities (visual, auditory, tactile, etc.), exploits the mechanisms of same-modality perception. Yet another aspect of ARCS which is worth mentioning in this connection is the coarse internal organisation of the (recall) image classification process, as it closely reflects Kosslyn's protomodel of visual perception (Kosslyn, 1994, p. 69). In ARCS, shape and location data are handled by different processes and trigger classificatory hypothesis formation and the hypothesis-driven testing of proposed classifications by means of additional perceptual clues.

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

To sum up, the following claims concerning image generation and classification in ARCS seem, in our view, to capture the more relevant aspects of this computational system for cognitive modelling.

(i) The recall images generated by ARCS are non-crisp *examples* of the classes of visually presented features and objects the system is capable of classifying. (This claim is supported by (ii).)

(ii) The classification of recall images is correctly achieved by the same mechanism performing classification of actual visual inputs.

(iii) Image generation results from the composition of two distinct computational processes: simple visual features are first *generated* and then *assembled* into more complex patterns.

(iv) The coarse internal organisation of the image generation and classification mechanisms reflects distinguishing traits of current models of high-level vision.

These claims are more precisely specified in the next four sections, which describe the symbolic and neural components of the hybrid image classifier and generator. In the final section, the cognitively significant aspects of this system are rehearsed in the light of a more general discussion on neurosymbolic hybrid approaches to cognitive modelling.

## RAM-DISCRIMINATORS AND CLASSIFICATION

In this section, we briefly describe the structure of RAM-discriminators, their training procedure, and the feature classification task performed by the multidiscriminator system of ARCS.

### RAM-discriminators

A RAM-discriminator consists of a set of N one-bit word RAMs with X inputs and a summing device (Σ). Any such RAM-discriminator can receive a binary pattern of X·N bits as input. The RAM input lines are connected to the input pattern by means of a so-called "random mapping". The summing device enables this network of RAMs to exhibit — just like other artificial neural nets that more directly model features of biological neural networks — generalisation and noise tolerance. (See fig. 1 for a schematic representation of a particular RAM-discriminator.)

In order to train the discriminator one has to set to 0 the RAM memory locations and to choose a training set formed by binary patterns of X·N bits (see fig. 2 in which a possible training set for the feature *vertical line* is proposed). For any training pattern a 1 is stored in that memory location of each RAM which is addressed by this input pattern. Once the training is completed, the RAM memory contents will be set to a certain number of 0's and 1's.

The information stored by the RAM during the training phase is used to deal with previously unseen patterns. When one of these is given as input, the RAM memory contents addressed by the input pattern are read and summed by Σ. The number $r$ thus obtained, which is called the discriminator *response*, is equal to the number of RAMs that output a 1. $r$ reaches the maximum value N if the input pattern belongs to the training set (in the present example, if the input pattern is one of the patterns in fig. 2). $r$ is equal to 0 if no three-bit component of the input pattern appears in the training set (no RAM outputs a 1). The other, intermediate values of $r$ express some kind of "similarity measure" of the input pattern with respect to the patterns in the training set.

We selected RAM-discriminators as digital neural components for our system on the basis of the following considerations: their training algorithm can be easily modified as needed for image generation tasks; RAM-discriminators are tailored for efficient implementation on conventional computers; the use of artificial neurons more closely reflecting biological neurons would not make a difference at the coarse level of cognitive modelling sketched in the introduction.

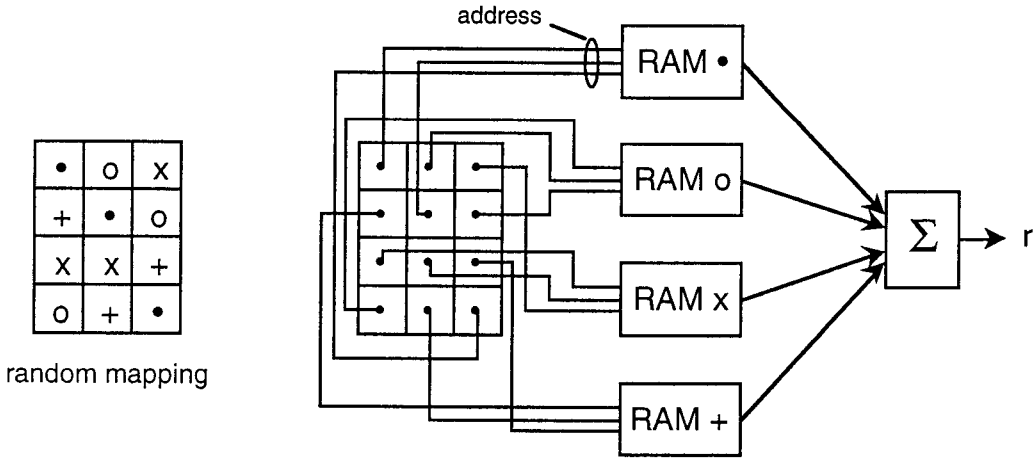



Figure 1 - RAM-discriminator

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

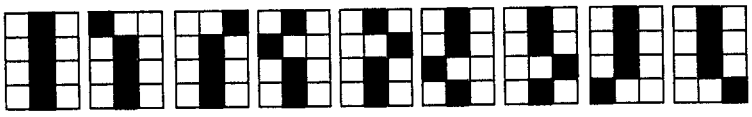


Figure 2: a possible training set for the feature *vertical line*

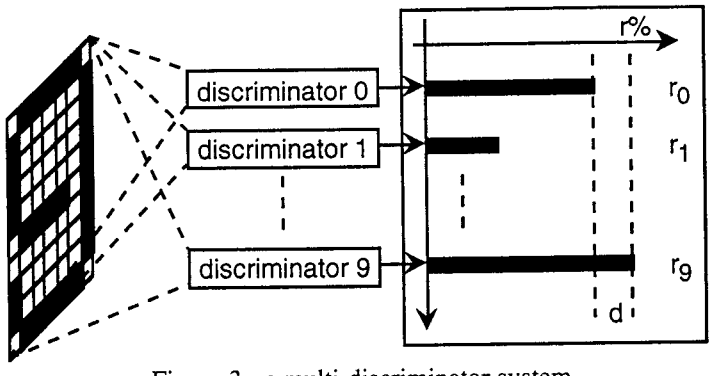



Figure 3 - a multi-discriminator system

### Multidiscriminators

Multidiscriminator systems are formed by various RAM-discriminators (Aleksander *et al.*, 1984). Each discriminator is trained on a particular class of patterns, and classification by the overall multidiscriminator system is achieved in the following way.

When a pattern is given in input, (see fig. 3 for a schematic representation of a multidiscriminator system formed by 10 RAM-discriminators), each discriminator gives a response on that input. The various responses are evaluated by an algorithm which compares them and computes the relative confidence *c* of the highest response (that is, the difference *d* between the highest and the second highest response, divided by the highest response).

In ARCS, six discriminators were trained with drawings representing variations (in angle width, size, or position) on the simple geometric features of fig. 4. The discriminators are organised into a multidiscriminator system which ranks their responses.

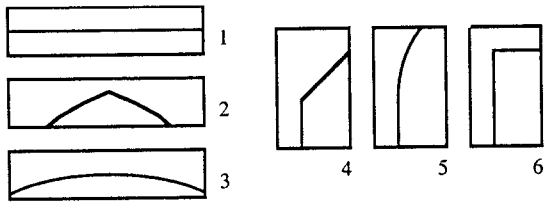



Figure 4 - geometric features

### (MODIFIED) RAM-DISCRIMINATORS AND FEATURE GENERATION

RAM-discriminators were modified in what their memory locations may hold and, correspondingly, in their training algorithm. These changes, which produce something very similar to the PLN nodes introduced in (Aleksander, 1988), allow one to store *q*-bit words in memory locations (where *q* is usually not greater than 8); in turn, this information can be exploited for producing recall images (and improving in other ways the behaviour of RAM-discriminators).

### Another training algorithm

The training algorithm of RAM-discriminators was changed in one respect only: instead of storing 1's, one just increases by 1 the memory location contents that are addressed by the input patterns. At the end of the training phase, the values of the memory contents will vary between 0 and M (where M is the number of training patterns). Fig. 5 shows the result of training the same RAM-discriminator of fig. 1 on the patterns of fig. 2, by means of the new algorithm.

The various memory content values can now be associated to subpattern frequency in the training set. For instance, the memory content of the address 010 associated to the +-th RAM is 5. This value indicates that the subpattern 010 is present 5 times in the training set of fig. 2. Moreover, one has to notice that the newly obtained memory contents do not give rise to different behaviours with respect to regularly trained RAM-discriminators, if one replaces the Σ device with another summing device, outputting the number of addressed memory locations whose content differs from 0.

One may take advantage of the new values stored in the RAMs in order to produce recall images (De Gregorio, 1997). This behaviour is significantly related (but not identical) to the exact input/output reversibility exhibited by the Bidirectional Associative Memories (BAM) introduced in (Kosko, 1988). The form of bidirectional behaviour we want to obtain from a RAM-discriminator D, trained with the new algorithm to pick out the elements of class X, must satisfy the following conditions:

(a) in one direction, D has to perform the usual classification process of RAM-discriminators;

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

| | • | x | o | + |
|---|---|---|---|---|
| 000 | 2 | 1 | 2 | 2 |
| 001 | 0 | 0 | 0 | 0 |
| 010 | 5 | 6 | 5 | 5 |
| 011 | 1 | 1 | 1 | 1 |
| 100 | 0 | 1 | 0 | 0 |
| 101 | 0 | 0 | 0 | 0 |
| 110 | 1 | 0 | 1 | 1 |
| 111 | 0 | 0 | 0 | 0 |

Figure 5 - RAM-discriminator of fig. 1 trained with the new algorithm

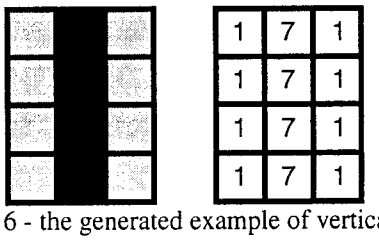
(b) in the opposite direction, D has to provide, when given the name of class X in input, an example of X.
It is not required that the example be identical to a member of the training set for D. Furthermore, we regard (b) as satisfied for any example of geometric feature in fig. 4 if the example is correctly classified by the multidiscriminator of ARCS.

The solution outlined here involves the construction of grey-level (rather than black and white) images exploiting the information held in the modified RAM memory locations. (A mathematical framework for approaching the reversibility problem for weightless systems is briefly sketched in (Redgers & Aleksander, 1992).)

### Generating grey-level images

The procedure for constructing grey-level images is the following. Let $b_1$, $b_2$, and $b_3$ be the first, second, and third bit forming the address of a memory location (for instance, $b_1 = 0$, $b_2 = 1$ and $b_3 = 1$ represent the address of the 011 memory location). To each of these bits a particular pixel of the image is associated (see the mapping in fig. 1). For any RAM, let $B_i$, for $i = 1, 2, 3$, be the sum of all memory location contents for which $b_i$ is 1 and the value stored is not equal to 0. For instance, for the •-discriminator in fig. 5 we obtain: $B_1 = 1$, $B_2 = 7$ and $B_3 = 1$. Applying this condition to every RAM in fig. 5 we obtain: $\forall j : j \in \{\bullet, x, o, +\}$, $B_{1j} = 1$, $B_{2j} = 7$, $B_{3j} = 1$. This regularity over the four RAMs depends on the fact that each pixel in the left-hand and right-hand columns of the matrix assumes value 1 (black) only once in the training set, whereas each pixel in the central column assumes value 1 (black) seven times in the training set.

Now, one can set the grey intensity level of each pixel associated to the bit $b_{ij}$ in such a way that it is proportional to the corresponding value $B_{ij}$: the higher is $B_{ij}$ the darker will be its grey intensity level. The result of this procedure applied to the modified RAM-discriminator trained for the feature *vertical line* is shown in fig. 6.



| | | |
|---|---|---|
| 1 | 7 | 1 |
| 1 | 7 | 1 |
| 1 | 7 | 1 |
| 1 | 7 | 1 |

Figure 6 - the generated example of vertical line

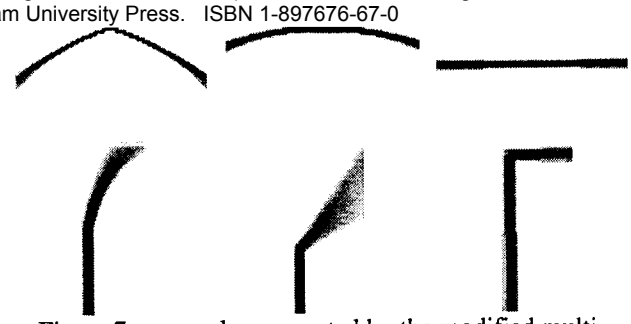


Figure 7 - examples generated by the modified multi-discriminator

Let us now turn to consider the wider class of simple visual features exemplified in fig. 5. In recognising these features, the multidiscriminator system of ARCS (trained by the modified algorithm described in this section) works in a canonical way, i.e. just as any regular multidiscriminator system. Moreover, the system may also provide, upon request, an example of each geometric feature it can classify. The results are shown in fig. 7.

## CLASSIFYING COMPLEX PATTERNS

ARCS was originally conceived for classifying actual photographs of portal shapes into one of the classes *a* to *f* exemplified in fig. 8. The hybrid approach was pursued after direct classification through a multi-discriminator system failed.

### The purely neural approach

The training set employed in the first, purely neural approach contained several drawings varying from the examples in fig. 8 only in the way of their position and size. The results obtained in a test made with 85 actual photographs of portals showed that only portal shapes belonging to classes *a* and *b* were correctly classified in a systematic way. The main reason for this failure emerges clearly from fig. 9, where the differences *a-b*, *e-f*, and *c-d* are shown.

While the relative complement of *b* in *a* is a rather large set of points, the other relative complements are much smaller and more localised. Thus, the information enabling one to discriminate between some such classes concerns the geometrical properties of small collections of points. It seems that spatial reasoning about geometrical features is crucially involved in this classification task. In particular, the more useful geometric cues are the top, the horizontal, and the vertical parts of portals, as exemplified in fig. 10 for polygonal portals. (Notice, however, that the horizontal parts are not essential for the non-linear portals *a*, *e*, and *f* in fig. 8.)

In order to mimic this geometrical reasoning capability, a hybrid system composed of a neural module and a symbolic module was adopted (De Gregorio, 1996): a

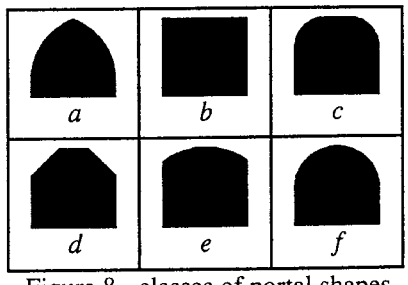



Figure 8 - classes of portal shapes

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

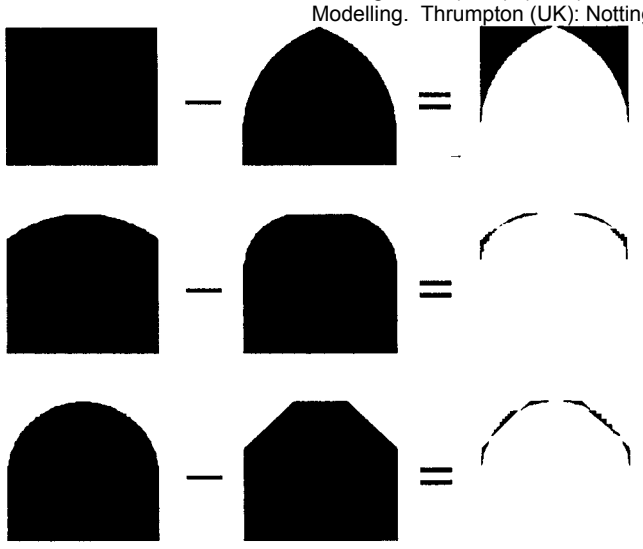


Figure 9

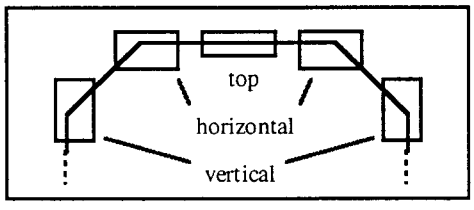



Figure 10

neural network for recognising geometric features was combined with a set of production rules specialised in classifying special arrangements of these features.

### The symbolic module

In the symbolic module one can distinguish between three different sets of production rules.

The first set of rules enables the system to evaluate the geometric "coherence" of the discriminator responses. For instance, suppose that on the current input "straight angle" and "obtuse angle" are the best and the second best response for both left and right horizontal parts, respectively. Then, the system uses these rules to verify whether the left and right recognised features are almost at the same height, almost aligned with the top, symmetric with respect to the top. If the "straight angle" responses do not satisfy these conditions while the "obtuse angle" responses do, the system selects the "obtuse angle" ones as possible responses, because they are geometrically "coherent".

The second set rules implements an *abduction-prediction-test* inference cycle (Burattini and De Gregorio, 1994) which can be roughly described in the following terms. From the ranked list of responses for the top feature, which is provided by the multidiscriminator system, the best response is picked out to start the cycle. The system abduces the portal shapes (hypotheses) that are consistent with the higher-ranked top feature. Given these hypotheses on overall portal shape, the system predicts which horizontal features may be detected, and activates the appropriate discriminators. Then, if one of these horizontal features is actually detected, the associated hypothesis is selected for further scrutiny, and the system activates the relevant discriminator to test again that hypothesis with respect to the vertical features; otherwise, the cycle is repeated on the next hypothesis (with the obvious termination conditions).

The third set of rules is formed by six rules, one for each portal shape, and enables the system to infer the final portal classification (if any). For example, the rule concerning polygonal portals can be informally stated as follows:

($R_p$) If top feature is in class no. 1 of fig. 4 and the horizontal and vertical features are in (or are obtainable by 90° rotation or specular reflection from instances of) class no. 4, then portal shape is polygonal (as in fig. 8, d).

The hybrid classifier has correctly classified the 85 actual photographs of portals that showed the inadequacy of the previously attempted, one-step neural classification approach (see fig. 11 for an input image - left - which is filtered - centre - and eventually classified - right; the higher the responses the darker the lines).

### GENERATING AND CLASSIFYING RECALL IMAGES OF COMPLEX PATTERNS

We have pointed out that the third set of production rules of the symbolic module enables the system to infer portal classification from portal components. By exchanging condition and action parts of these rules, one obtains new rules specifying which parts are to be assembled together to form the recall image of a given portal shape. For example, from rule ($R_p$), one obtains a rule which can be informally stated as follows:

($R_p$<–) If portal shape is polygonal (as in fig. 8, d) then top feature is in class no. 1 of fig. 4 and the horizontal and vertical features are in (or are obtainable, by 90° rotation or specular reflection, from instances of) class no. 4.

Similarly, by exchanging condition and action parts of the "geometrical coherence" rules, we obtain rules determining categorical spatial relationships between parts of complex recall images (ruling that vertical and horizontal components must be aligned and symmetrical with respect to the top, that the horizontal components must be aligned with the top).

An assembly problem which is not solved in the current implementation is how to determine the proportions of the components and their metric spatial relationships. A dynamic solution would require the recording of the corresponding data during the classification of actual visual inputs — in order to set, and then progressively refine the values of such spatial relationships as new examples are being classified. We have not yet implemented a process of this sort, which takes dynamically into account all past experience of the system. Currently, the system assigns fixed default values to such relationships.

The grey-level recall images obtained by means of the "assembly" rules are correctly recognised by the classifier of ARCS (see fig. 12). As with the actual photographs of portals given in input to the system, these recall images are first processed by a grey-level filter, which produces a binary image sharpening the non-crisp input image. The latter is given in input to the hybrid classifier described in

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

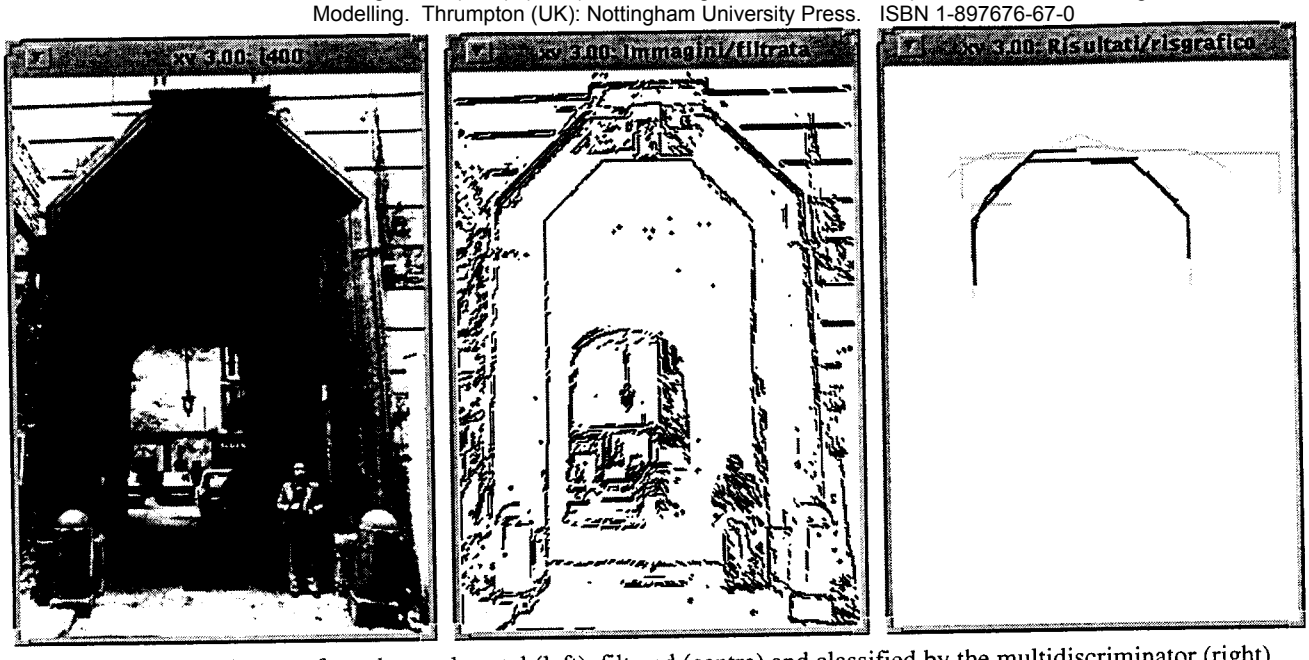


Figure 11 - input image of a polygonal portal (left), filtered (centre) and classified by the multidiscriminator (right).

the previous section. (Let us notice, in passing, that the existence of a filter editing visual information, by filling in missing details or sharpening fuzzy data is postulated in various cognitive models. See, for instance, Kosslyn (1994, p.389) and Rocha (1997, p. 157).)

Fig. 12 shows the grey-level recall image of a polygonal portal (left) which is filtered into a black and white image (centre). The responses of the discriminators for the various features (right) are represented in different intensities of grey: darker lines correspond to higher response values.

Is a recall image obtained in this way a *typical* example of the corresponding class? To the extent that frequency and typicality can be assimilated, the (filtered) recall images might be regarded as typical pictorial representatives of their classes. The non-crisp recall image preserves in its darker, more noticeable parts a trace of the more frequently encountered patterns during the training phase. In the filtered recall image (see fig. 12, centre), the sharpened trace induced by such patterns stands out even more clearly.

Under the hypothesis that frequency and typicality are identifiable in the domain under consideration, one can assert that the various classes of portals in fig. 8 are pictorially represented in the system by means of recall images, whereas the more abstract, general class *portal* can only be represented by a disjunction of statements such as ($R_p$). (See Ullman (1996, p. 184) for a recent discussion of related issues.)

In concluding this section let us notice that (complex) recall images may be adjusted on the basis of further experience. If, during a new training session, the memory location contents of the various RAMs are modified, then the grey-intensity level of the associated pixels in the generated example will change accordingly.

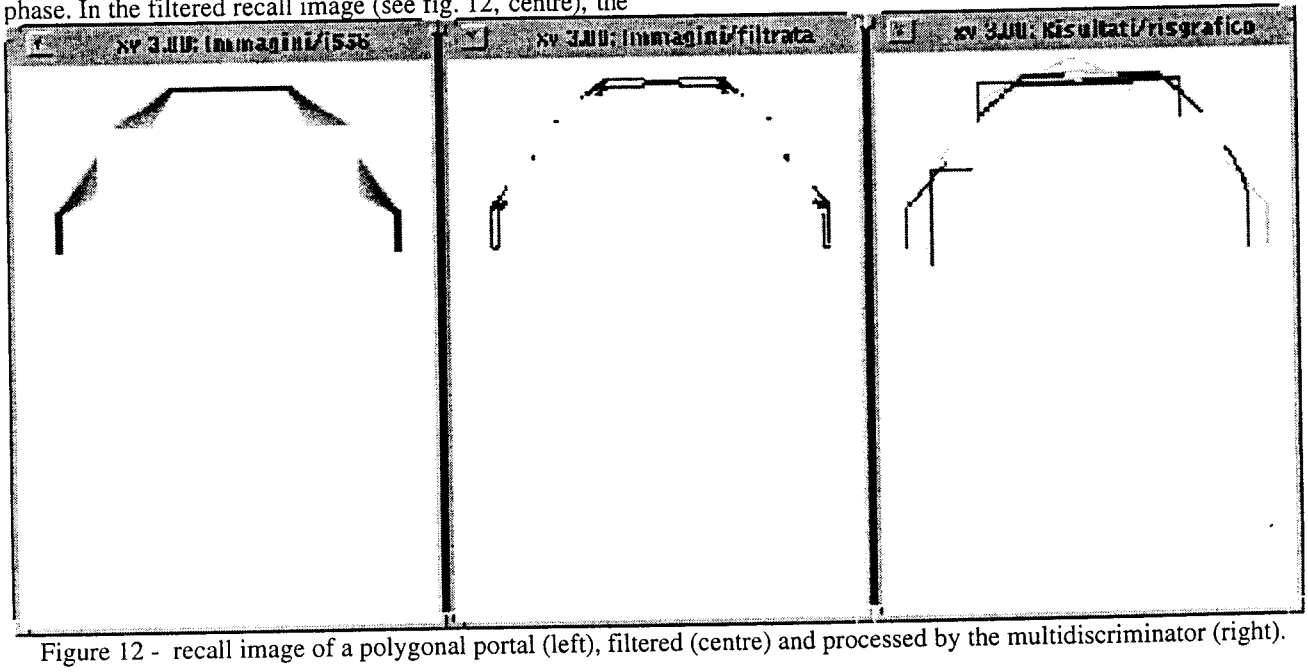


Figure 12 - recall image of a polygonal portal (left), filtered (centre) and processed by the multidiscriminator (right).

Ritter, F. E., & Young, R. M. (Eds.). (1998). Proceedings of the Second European Conference on Cognitive Modelling. Thrumpton (UK): Nottingham University Press. ISBN 1-897676-67-0

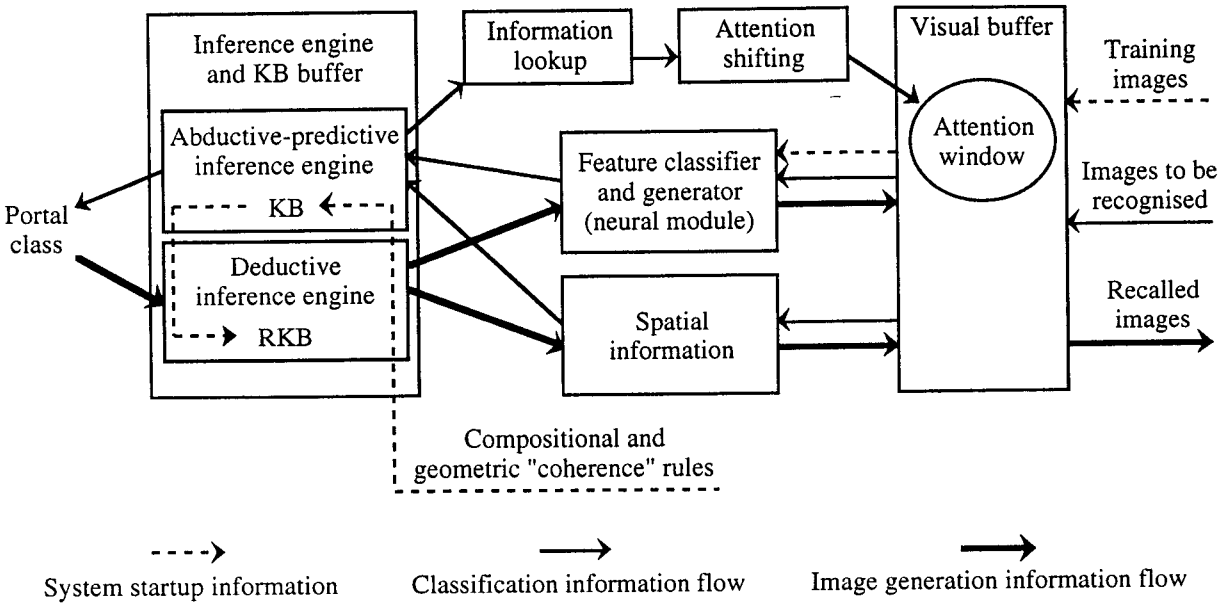



Figure 13 - system layout (RKB stands for Reverse Knowledge Base).

## NEUROSYMBOLIC HYBRID SYSTEMS FOR COGNITIVE MODELLING

The idea that visual perception and imagery share the same underlying mechanisms is supported by various behavioural, neuropsychological and more recently accrued brain imaging data. ARCS reflects this hypothesis, insofar as visual object classification and generation of recall images in the task domain are concerned. Notably, classification of both actual and recall images is performed by exactly the same mechanism, whereas just a reversing of some neural and symbolic mechanisms involved in classification is needed to generate recall images. This reversing might be viewed as contributing to an abstract, purely functional modelling of reentrant neuronal connections (Edelman, 1989, p. 195, Damasio & Damasio, 1994), which combines the two quite different computational techniques of production rules and weightless neural systems, that are usually associated to the symbolic and subsymbolic paradigm, respectively.

Neural and symbolic techniques converge into hybrid neurosymbolic approaches to cognitive modelling (Hilario, 1995). These approaches seem to provide appropriate tools for modelling the interaction between top-down reasoning processes (typically simulated by symbolic computation) and bottom-up perceptual processes (often simulated by neural computation). The hybrid system ARCS is a case in point, since classification is accomplished by the interaction of both types of processes. We were unable to carry out this task by means of the purely neural, one-step classification process that was previously adopted.

Also the generation of recall images in ARCS is not a one-step process. The neural module generates examples of the various classes of features. Categorical spatial information about the position of features is represented in the symbolic module, and enables the system to assemble individually generated features into a complex recall image. Thus, the partition of the system into a neural and a symbolic component corresponds to a decomposition into functionally distinct subsystems, which is postulated in Kosslyn's model of imagery (Kosslyn, 1994 and 1995). A more detailed system layout is shown in fig. 13.

In general, the cognitive models taking the form of hybrid systems may raise a special epistemological problem since, in principle, model and cognitive reality might be compared at the symbolic, subsymbolic, and even neuronal levels. The problem does not arise if the hybrid system is proposed as a coarse model of input/output behaviours, relative to the major components of the cognitive system only. At this level of comparison, it is immaterial whether the various components are implemented as a neural net or as a symbolic system: only the functions computed by each component are relevant. Claims (i)-(iv) — made in the introduction and relating ARCS to the computational modelling of aspects of high-level vision and visual imagery — are to be understood just at this functional level of comparison. Thus, the choice of a hybrid architecture for ARCS is only pragmatically motivated: it allows one to simulate cognitive functions that seem to elude symbolic (respectively, neural) computation techniques in isolation.

These various observations suggest that the hybrid approach pursued here is not in principle incompatible with later developments possibly allowing one to substitute a neural module for a symbolic module, *salva* functional equivalence, and eventually transforming a hybrid system into a *unified* neurosymbolic system. In such unified systems, neurally implemented modules perform symbolic reasoning, too (Hilario, 1995). In principle, ARCS may be transformed into a unified system, by neurally implementing the production rule system performing geometric reasoning (adopting, e.g., the methodology proposed in (Aiello *et al.*, 1997, Aiello *et al.*, 1995)). It is not obvious, however, that every hybrid system can be turned into a unified system. There are forms of reasoning that currently go beyond unified

approaches; these include most forms of classical reasoning in first-order logic, which lack (efficient) neural implementations. For discussion, see (Aiello *et al.*, 1997, Ajjanagadde & Shastri, 1993, Sun, 1994).

Finally, some remarks about future work on recall images in problem solving. One may endow the system ARCS with an explanation module combining words and pictures. If a user wants to know why a given house portal was classified as, say, polygonal, the system may justify this conclusion roughly as follows:

(a) it generates and displays examples of the visual features detected in the input image;

(b) it verbally declares and visually exemplifies the spatial relationships that were recognised to hold between the displayed features;

(c) it displays an example of the overall portal shape, obtained by properly arranging the generated features, next to the input image.

Similar uses of multiple representations have been recently discussed in (Tabachneck-Schijf *et al.*, 1997). Both in ARCS and in other contexts that we are currently exploring (concerning simple 2-D geometric figures), recall images may be used to complete partially occluded pictures, so that the various completions that the system declares as consistent with the occluded image can be shown.

## ACKNOWLEDGMENTS

We wish to thank Carlo Arcelli for helpful comments on an earlier version. One of the authors (MDG) is deeply grateful to Igor Aleksander who introduced him to the fascinating world of weightless systems.

## REFERENCES

Aiello, A., Burattini, E., & Tamburrini, G. (1997). Neural nets and rule-based reasoning. In C.D. Leondes (ed.), *Fuzzy Logic and Expert Systems Applications*, Academic Press, Boston MA.

Aiello, A., Burattini, E., & Tamburrini, G. (1995). Purely neural, rule-based diagnostic systems. Part I. Production rules. *Int. J. of Intelligent Systems 10*, 735-749.

Ajjanagadde, V., & Shastri, L. (1993). From simple associations to systematic reasoning. *Behavioral and Brain Sciences 16*, 417-494.

Aleksander, I. (1988). Logical Connectionist Systems. In Eckmiller R. & von der Malsburg C. (eds.), *Neural Computers*, Springer Verlag, Berlin 189-197.

Aleksander, I., & Morton, E. (1990). *An Introduction to Neural Computing*. Chapman & Hall, London.

Aleksander, I., Thomas, W., & Bowden, P. (1984). WISARD, a radical new step forward in image recognition. *Sensor Review 4*, 29-40.

Burattini, E. (ed.) (1994). *Intelligenza Artificiale e Recupero Edilizio*, Istituto di Cibernetica CNR, I-80072 Arco Felice, Italy.

Burattini E., & De Gregorio, M. (1994). Qualitative Abduction and Prediction. Regularities over Various Expert Domains. *Information and Decision Technologies 19*, 471-481.

Damasio, A.R., & Damasio, H. (1994). Cortical systems for retrieval of concrete knowledge: the convergence zone framework. In C. Koch & J.L. Davis (eds.), *Large-Scale Neuronal Theories of the Brain*, 61-74. MIT Press, Cambridge MA.

De Gregorio, M. (1996). Integrating inference and neural classification in a hybrid system for recognition tasks. *Mathware & Soft Computing 3*, 271-279.

De Gregorio, M. (1997). On the reversibility of multi-discriminator systems. Technical report 125/97, Istituto di Cibernetica CNR, I-80072 Arco Felice, Italy.

Edelman, G.M. (1989). *The Remembered Present*. Basic Books, New York NY.

Farah, M.J., Gazzaniga, M.S., Holtzman, J.D., & Kosslyn, S.M. (1985). A left hemisphere basis for visual imagery? *Neuropsychologia 23*, 115-118.

Finke, R.A. (1985). Theories relating mental imagery to perception. *Psychological Bulletin 98*, 236-259.

Hilario, M. (1995). An overview of strategies for neurosymbolic integration. In *Connectionist-Symbolic integration: from unifies to hybrid approaches*, Working Notes, IJCAI '95, Montréal, 19-20 August, 1995.

Kosko, B. (1988). Bidirectional Associative Memories, *IEEE Transactions on Systems, Man, and Cybernetics 18*, 49-60.

Kosslyn, S.M. (1994). *Image and Brain*. MIT Press, Cambridge MA.

Kosslyn, S.M. (1995). Mental imagery. In D.N. Osherson & S.M. Kosslyn (eds.) *Visual Cognition*, vol. 2 of *An invitation to Cognitive Science*, 267-296. MIT Press, Cambridge MA.

Redgers, A., & Aleksander, I. (1992). Digital neural networks. In Warwick K. *et al.* (eds.), *Neural Networks for Control and Systems*, Peter Peregrinus, London, 13-30.

Rocha, A.F. (1997). The brain as a symbol-processing machine. *Progress in Neurobiology 53*, 121-198.

Sun, R. (1994). *Integrating Rules and Connectionism for Robust Commonsense Reasoning*. John Wiley, New York NY.

Tabachneck-Schijf, H.J.M., Leonardo, A.M., & Simon, H.A. (1997). CaMeRa: a computational model of multiple representations. *Cognitive Science* **21**, 305-350.

Ullman, S. (1996). *High-Level Vision*. MIT Press, Cambridge MA.