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

Influence of Mapping on Analog Access: A Simulation Experiment with AMBR

Alexander A. Petrov*

*New Bulgarian University
Department of Cognitive Science
21, Montevideo Str.
Sofia 1635, Bulgaria
apetrov@cogs.nbu.acad.bg

Boicho N. Kokinov#*

#Institute of Mathematics and Informatics
Bulgarian Academy of Science
Bl. 8 Acad. G. Bonchev Str.
Sofia 1113, Bulgaria
kokinov@cogs.nbu.acad.bg

ABSTRACT

This paper contrasts two views about the relationship between the processes of access and mapping in analogy-making. According to the modular view, analog access and mapping are two separate 'phases' that run sequentially and relatively independently. The interactionist view assumes that they are interdependent subprocesses that run in parallel. The paper argues in favor of the second view and presents a simulation experiment demonstrating its advantages. The experiment is performed with the computational model AMBR and illustrates one particular way in which the subprocess of mapping can influence the subprocess of access.

KEYWORDS

Analogy-making, interactionist approach, access, mapping, simulation experiment, hybrid cognitive architecture.

INTRODUCTION

A crucial point in analogy-making is the retrieval of a base (or source) analog. Accessing an appropriate base from the vast pool of episodes stored in the long-term memory is not only a logical necessity (one cannot make analogies without a source) but apparently is the most difficult and capricious element of analogy-making. Starting with the classical experiments of Gick and Holyoak (1980) it has been repeatedly demonstrated that people have difficulties in spontaneously accessing a base analog, especially when its domain is very different from that of the target problem. In the aforementioned study only about 20% of the subjects were able to solve the so-called radiation problem even though an analogous problem (with solution) was presented shortly before the test phase. When provided by an explicit hint to use this source analog, however, 75% of the subjects achieved the solution. This great difference between the two experimental conditions was attributed to the difficulty of analog access.

On the other hand, we know a lot of stories about great scientists making discoveries by spontaneously using remote analogies. We have also personal experience in everyday usage of remote analogies. A recent study by Wharton, Holyoak, and Lange (1996) has demonstrated that about 35% of their subjects were successfully reminded about a remote analog story studied 7 days earlier when cued by the target story. (They have used a directed reminding task, not a problem solving task, however.)

Researchers of analogical access have become interested in the features of a remote analog that facilitate retrieval. Most data in the field (Holyoak and Koh, 1987, Ross 1989) suggest that analogical access is almost exclusively guided by superficial semantic similarities between base and target—similar objects and relations, similar themes, similar story lines, etc. In contrast, analogical mapping is dominated by the structural similarity between target and base, i.e. having common systems of relations (Gentner, 1983, 1989). This explains why remote analogs are much more difficult to access than to map—they lack the superficial similarities needed for access but do have the (quasi)isomorphic relational structure necessary for mapping.

This clear separation stimulated the researchers in the field to build separate models of mapping and retrieval and even to claim that they are different cognitive modules. Thus Gentner (1989) claims that 'the analogy processor (the mapping machine) is a well-defined separate cognitive module whose results interact with other processes, analogous to the way some natural language models have postulated semi-autonomous interacting subsystems for syntax, semantics, and pragmatics.' Although she explicitly mentions in a footnote that this should not be considered in the Fodorian sense as innate and impenetrable, the actual models built are quite impenetrable. This line of research has generated a number of quite successful models that explained the data and made some new predictions. Typically, a model of mapping is coupled with a (separate) model of retrieval. The best-known examples are SME + MAC/FAC (Falkenhainer, Forbus, and Gentner, 1986; Forbus, Gentner, and Law, 1995) and ACME + ARCS (Holyoak and Thagard, 1989; Thagard, Holyoak, Nelson, and Gochfeld, 1990).

However, the experimental work soon revealed that the pattern is not that clear and straightforward. It has been demonstrated that superficial similarities do play an important role in mapping as well. In particular crossmapping is difficult (Ross, 1989). This led Holyoak and Thagard to include syntactic, semantic, and pragmatic constraints in their model of mapping ACME (Holyoak & Thagard, 1989) and to develop their multi-constraint theory (Holyoak & Thagard, 1995).

There are also some indications that structural similarity might play a role in access as well. Thus Ross (1989) demonstrated that in some cases (when the general story line is similar) structural similarity plays a positive role in retrieval, while in other cases (when the general story line is dissimilar) it does not play any role or can even worsen the results. The results of Wharton, Holyoak, and Lange (1996) also support indirectly the hypothesis that structural correspondences might affect the access. This was reflected in the models being proposed. Both MAC/FAC and ARCS included a submodule of partial mapping in the module of retrieval, thus considering structural similarities at an early stage.

To sum up, the initial separation between retrieval and mapping was founded on their different psychological characteristics—semantic factors govern the retrieval, structural factors govern the mapping. Subsequent more precise experiments, however, cast doubt on this clear separation. These complications were accommodated by making patches to the original models. Finally, it was acknowledged that all kinds of constraints affected all phases of analogy-making, although to different extent (Holyoak & Thagard, 1995).

The experimental data themselves became more and more complex and controversial. These controversies can be explained in terms of more and more sophisticated classifications of the types of similarities involved in access and mapping (Ross, 1989; Ross & Kilbane, 1997). We argue, however, that these problems are resolved more parsimoniously by adopting a principally different view of analogy-making.

This resembles an episode of the history of astronomy. The geocentric system of Ptolemy started as a straightforward theory that described the observable movement of both stars and planets remarkably well[1]. As accuracy of measurement increased, however, discrepancies between theory and data crept in every now and then. It became routine for astronomers to deal with such 'anomalies' by adding more and more epicycles. But as time went on, it became evident that astronomy's complexity was increasing far more rapidly than its accuracy and that a discrepancy corrected in one place was likely to show up in another (Kuhn, 1970).

Back to the domain of analogy-making, most classical models assume sequential processing: *first* the retrieval process finds the base for analogy and *then* the mapping process builds the correspondences between the target and the retrieved base (Figure 1). Thus MAC/FAC+SME and ARCS+ACME are linear models separating retrieval and mapping in time and space. This view underlies most of the experimental work in the field as well. Researchers often contrast hint versus non-hint conditions in problem solving supposing that in the first case only mapping takes place, while in the second retrieval and mapping are running one after the other. However, as Ross (1989) has noted, even when explicitly hinted to use a certain analog subjects still must access the details of its representation. Another common experimental technique uses a memory task (typically recall) for studying access with the assumption that the same processes take place during analogical problem solving.

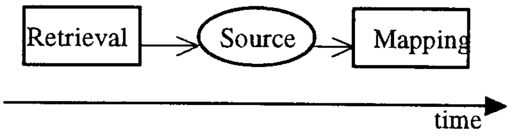

Figure 1. Dominating sequential models of analogy-making.

The limitations of both the models and experimental methods can be overcome by giving up the linearity assumption. This might look strange at first glance—how can you map the source analog onto the base if you have not even accessed it?! If, however, one reconsiders one more assumption—that there are centralized representations of situations/problems in human memory—then it becomes clear that whenever we have partial retrieval of the base (having recalled a few details) we can start looking for corresponding elements in the target. This allows us to conceptualize access and mapping as parallel processes that can interact (Figure 2). In this paradigm, access and mapping refer not to phases or other behavioral steps, but rather to separate mechanisms that both play a role in selecting and activating a base and in finding the correspondences between base and target.

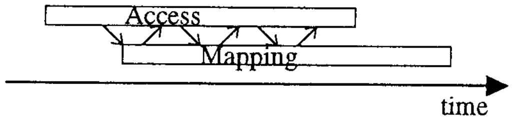

Figure 2. Parallel and interactive models of analogy-making.

The current paper explores the implications of the parallel and interactive view on access and mapping by running simulation experiments with an integrated model of human (analogical) reasoning called AMBR (Kokinov, 1994c, Petrov, 1997). These experiments provide a detailed example of how these two processes can interact and thus open space for new theoretical speculations as well as for new experimental paradigms. AMBR's predictions about the development of the process over time call for appropriate experimental methods capturing the dynamics of human analogy-making—RT studies, think-aloud protocols, etc. Some of the controversies around the role of superficial and structural similarities in access and mapping 'phases' can now be expressed in terms of the interactions between the two mechanisms.

[1] It is still used today as an engineering approximation.

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 very important contribution of the simulation is that it demonstrates how the supposedly later 'phase' of mapping can influence the supposedly earlier 'phase' of access. A detailed example shows how the access process develops over time and how it is influenced by the concurrent mapping process. This is contrasted with the case of isolated access. Different results are obtained in the two cases. These results correspond to the data of Ross and Sofka (unpublished) which main conclusions are summarized in (Ross, 1989) as follows: '... other work (Ross & Sofka, 1986) suggests the possibility that the retrieval may be greatly affected by the use. In particular, we found that subjects, whose task was to recall the details of an earlier example that the current test problem reminded them of, used the test problem not only as an initial reminder but throughout the recall. For instance, the test problem was used to probe for similar objects, and relations and to prompt recall of particular numbers from the earlier example. The retrieval of the earlier example appeared to be interleaved with its use because subjects were setting up correspondences between the earlier example and the test problem during the retrieval.' The simulation data presented in the current paper (obtained absolutely independently and based only on the theoretical assumptions of DUAL and AMBR) exhibit exactly the same pattern of interaction.

We must admit that even in a highly parallel and interactive model such as AMBR the effects of interactions are not predominating. In the majority of cases the independent work of the access mechanism might well yield the same results as the interaction between mapping and access described above. That is why the classical linear models of analogy have been successful and have contributed a lot to our understanding of human analogy-making. However, exactly the few exceptional cases that do provide different results in a parallel model are the more interesting and those who make the interpretation of the experimental data look controversial if analyzed in the spirit of the sequential models.

There are a few other models that advocate a parallel, overlapping, and interactive view on analogy—Copycat (Mitchell, 1993, Hofstadter, 1995), Tabletop (French, 1995, Hofstadter, 1995), and LISA (Hummel and Holyoak, 1997). However, Copycat and Tabletop do not model retrieval at all—they model the parallel work and interaction between perception/representation building and mapping. LISA also integrates access and mapping and performs them in parallel. Thus the mapping mechanism (connectionist learning in this case) influences the access. As a result, LISA could in principle demonstrate effects similar to those reported here.

BRIEF DESCRIPTION OF THE ARCHITECTURE DUAL AND THE MODEL AMBR

The basis for the simulation experiment discussed in this paper is a model called AMBR (Associative Memory-Based Reasoning). It is built on the cognitive architecture DUAL. Space limitations allow only an extremely sketchy description of DUAL and AMBR here. The interested reader is referred to earlier publications (Kokinov, 1988, 1994a,b,c; Petrov, 1997).

DUAL is a multi-agent cognitive architecture that supports dynamic emergent computation (Kokinov,Nikolov, and Petrov, 1996). All knowledge representation and information processing in the architecture is carried out by small entities called *DUAL agents*. Each DUAL-based system consists of a large number of them. There is no central executive in the architecture that controls its global operation. Instead, each individual agent is relatively simple and has access only to local information, interacting with a few neighboring agents. The overall behavior of the system emerges out of the collective activity of the whole population. This 'society of mind' (Minsky, 1986) provides a substrate for concurrent processing, interaction, and emergent computation.

Each DUAL agent is a hybrid entity that has symbolic and connectionist aspects (Kokinov 1994a,b,c). On the symbolic side, each agent 'stands for' something and is able to perform certain simple manipulations on symbols. On the connectionist side, it sends/receives activation to and from its immediate neighbors. Thus, we may adopt an alternative terminology and speak of *nodes* and *links* instead of *agents* and *interactions*. The population of agents may be conceptualized as a network of nodes.

The long-term memory of a DUAL-based system consists of the network of all agents in that system. The size of this network can be very large. Only a small fraction of it, however, may be active at any particular moment. The active subset of the long-term memory together with some temporary agents constitutes the *working memory (WM)* of the architecture. The mechanism of spreading activation plays a key role for controlling the size and the contents of the WM. There is a threshold that sets the minimal level of activation that must be obtained by an agent to enter the WM. There is also a spontaneous decay factor that pushes the activation levels back to zero. As the pattern of activation changes over time, some agents from the working memory fall back to dormancy, others are activated, etc. Only active agents may perform symbolic computation. Moreover, the speed of this computation depends on the level of activation of the respective agent. This makes the computation in DUAL dynamic and context-sensitive (Kokinov et al., 1996; Kokinov, 1994a,b,c). One particular consequence of this dynamic emergent nature of the architecture is that, although all micro-level processing is strictly deterministic, the macroscopic behavior of a DUAL system can be described only probabilistically.

The AMBR model takes advantage of these architectural features to account for some phenomena of human reasoning and in particular reasoning by analogy (Kokinov, 1988, 1994c). Again, due to space limitations we will consider only a small fraction of model's mechanism's.

Analog access in AMBR is done by means of spreading activation by the connectionist aspects of the DUAL

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.

agents. In particular, only few of the many episodes stored in the long-term memory are active during a run and only they are accessible for processing. The episodes or 'situations' have decentralized representations—it is not a single agent but a whole *coalition* that represents the elements of a situation and the relationships among them[2]. Therefore, it is possible that an episode is only partially accessed because only some of the agents have entered the WM.

The process of analogical mapping is done in AMBR by a combination of three mechanisms—marker passing, constraint satisfaction, and structure correspondence (Kokinov, 1994c; Petrov, 1997). The main idea is to build a *constraint satisfaction network (CSN)* to determine the mapping between two situations. This network consists of *hypothesis agents* representing tentative correspondences between two elements. Consistent hypotheses support, and incompatible ones inhibit each other.

This is similar to other models of analogy-making and notably ACME (Holyoak and Thagard, 1989). AMBR differs from the latter model, however, in several ways: (*i*) the CSN is constructed dynamically, (*ii*) only hypotheses that have some justification are created, (*iii*) the CSN is incorporated into the bigger working memory network, and (*iv*) there is no separate relaxation phase so there is a partial mapping at each moment.

The implication of these four points is that, unlike ACME and most other analogy models, the processes of access and mapping run in parallel and influence each other in AMBR. In other words, the model departs from the classical 'pipeline' paradigm and aims at a more interactive account of analogy making.

The influence between the two subprocesses in AMBR goes in both directions. The present paper concentrates on the 'backward' direction—from mapping to access. The next section describes a simulation experiment that sheds light on this kind of influence.

SIMULATION EXPERIMENT METHOD

We performed a simulation experiment to contrast the two ways of combining access and mapping—parallel vs. serial. The experiment also tested whether the AMBR model was capable to access a source analog out of a pool of episodes, and to map it onto a target situation.

Design

The experiment consisted of two conditions. Both conditions involved running the model on a target problem. In the 'parallel condition', AMBR operated in its normal manner with the mechanisms for access and mapping working in parallel. In the 'serial condition', the program was artificially forced to work serially—first to access and only then to map. The target problem and the content of the long-term memory were identical in all runs. The topics of interest fell into two categories—the final mapping constructed by the program and the dynamics of the underlying computation. The latter was monitored by recording a set of variables describing the internal state of the system at regular time intervals throughout each run.

Materials

The domain used in the experiment deals with simple tasks in a kitchen. The long-term memory of the model contains semantic and episodic knowledge about this domain. It has been coded by hand according to the representation scheme used in DUAL and AMBR (Kokinov, 1994c; Petrov, 1997). The total size of the knowledge base is about 250 agents. It states, for example, that water, milk, and tea are all liquids, that bottles are made of glass, and the relation 'on' is a special case of 'in-touch-with'. The LTM also stores the representations of eight situations related to heating and cooling liquids. Two of these eight situations are most important for the experiment and are described below together with the target problem.

Situation A: *There is a cup and some water in it. The cup is made of china. There is an immersion heater in the water. The immersion heater is hot. This state of affairs causes that the water is hot.*

Situation B: *There is a glass and an ice cube in it. The glass is made of [material] glass. The glass is in a refrigerator. The refrigerator is cold. This state of affairs causes that the ice cube is cold.*

Target problem (situation T): *There is a glass and some cola in it. The glass is made of [material] glass. There is an ice cube in the coca cola. The ice cube is cold. What is the consequence of this state of affairs?*

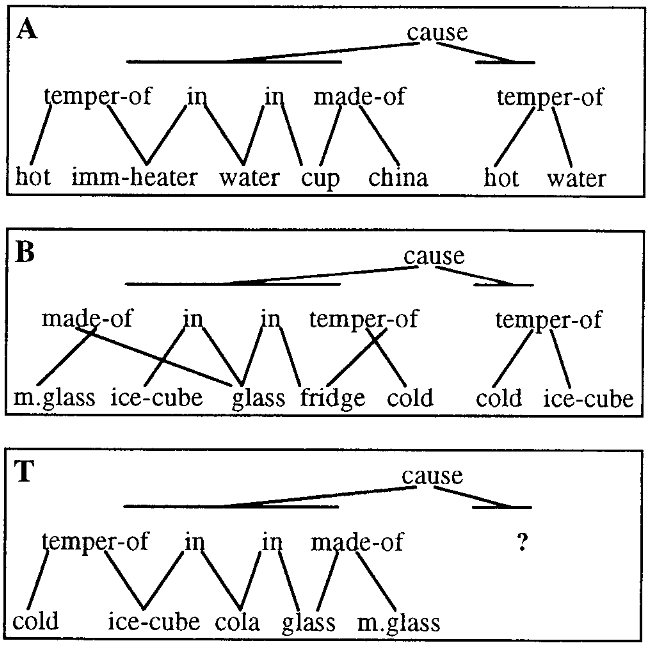

Figure 3. Simplified representations of situations A, B, and T. (The actual AMBR representations are more complex.) See text for details.

[2] This is one of the differences between the current version of the model (AMBR2) and the original proposal (AMBR1) as set forth by (Kokinov, 1994c).

As evident from Figure 3, both situations may be considered similar to the target problem. There are some differences, however. Situation **B** involves the same objects and relations as the target but the structure of the two are different. In contrast, situation **A** involves different objects but its system of relations is completely isomorphic to that of the target. According to Gentner (1989), the pair **A-T** may be classified as analogy while **B-T** as mere appearance. Thus it was expected that situation **B** would be easier to retrieve from the total pool of episodes stored in LTM. On the other hand, **A** would be more problematic to retrieve but once accessed it would support better mapping.

Procedure

The Common Lisp implementation of the AMBR model was run two times on the target problem. The two runs carried out the 'parallel' and the 'serial' conditions of the experiment, respectively. The contents of the long-term memory and the parameters of the model were identical in the two conditions.

Recall that situations have decentralized representations in AMBR. The target problem was represented by a coalition of 13 agents standing for the ice-cube, the glass, two instances of the relation 'in' and so on. 11 of these agents were attached to the special nodes that serve as activation sources in the model. This attachment was the same in the two experimental conditions.

In the parallel condition, the model was allowed to run according to its specification. That is, all AMBR mechanisms ran in parallel, interacting with one another. The program iterated until the system reached a resting state. A number of variables were recorded at regular intervals throughout the run. Out of these many variables, the so-called *retrieval index* is of special interest. It is computed for each situation and is based on the average activation level of the respective coalition. More concretely, the retrieval index is calculated by the formula:

$$\mathrm{RI}\,(t) = \frac{\sum a_i(t)}{0.5 + N}\ ,$$

where N is the total number of agents in the coalition and $a_i(t)$ is the activation level of agent$_i$ at moment t.

In short, at the end of the run we had the final mapping constructed by the program as well as a log file of the retrieval indices of all eight situations from the LTM.

In the serial condition, the target problem was attached to the activation source in the same way and the same data were collected. However, the operation of the program was forcefully modified to separate the processes of access and mapping. To that end, the run was divided in two steps.

During step one, all mapping mechanisms in AMBR were manually switched off. Thus, spreading activation was the only mechanism that remained operational. It was allowed to work until the pattern of activation reached asymptote. The situation with the highest retrieval index was then identified. If we hypothesize a 'retrieval module', this is the situation that it would access from LTM.

After the source analog was picked up in this way, the experiment proceeded with step two. The mapping mechanism was switched back on again but it was allowed to work only on the source situation retrieved at step one. This situation was mapped to the target. Thus, at the end of the second run we had the final mapping constructed at step two, as well as two logs of the retrieval indices.

RESULTS AND DISCUSSION

In both experimental conditions the model settled in less than 150 time units and produced consistent mappings. By 'consistent' we mean that each element of the target problem was unambiguously mapped to an element from LTM and that all these corresponding elements belonged to one and the same base situation. Stated differently, the mappings were one-to-one and there were no blends between situations.

In the parallel condition, the target problem was mapped to situation **A**, revealing the isomorphism illustrated in Figure 4. One element from the source situation remained unmapped—the agent representing that the water becomes hot. This proposition is a good candidate for inference by analogy. *Mutatis mutandis*, it would yield the conclusion that the cola becomes cold. (In the current version of AMBR the mechanisms for analogical transfer are not implemented yet.)

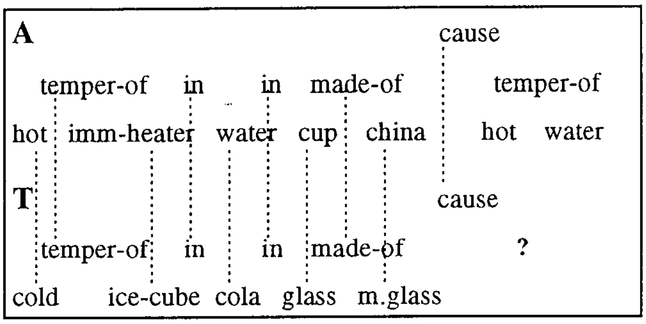

Figure 4. Correspondences constructed by the model in the parallel condition.

In the serial condition, situation **B** won the retrieval stage. This is explained by the high semantic similarity between its elements and those of the target—both deal with ice cubes in glasses, cold temperatures, etc. The asymptotic level of the retrieval index for **B** was more than three times greater than that of any other situation. In particular, situation **A** ended up with only 4 out of 14 agents passing the working memory threshold.

According to the experimental procedure, situation **B** was then mapped to the target during the second stage of the run. The correspondences that emerged during the latter stage mapped consistently the chains of two interlocking relations 'in' and the higher-order relation 'cause' (Figure 5). This structural alignment was achieved, however, at the expense of the semantic similarity between objects—the two glasses did not correspond, which in turn violated the structural constraint regarding the arguments of the relation 'made-of'.

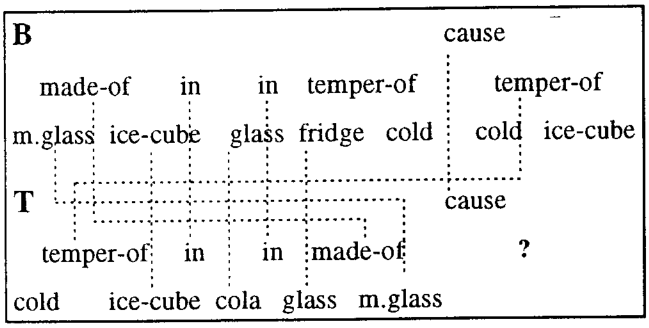

Figure 5. Correspondences constructed by the model in the serial condition.

It might be argued that these flaws of the mapping are not very serious, especially in the light of the structure mapping theory (Gentner, 1983). If we consider the material of the glasses an attribute, it is permissible to give it little attention. There is, however, a more serious flaw in the set of correspondences. The proposition 'temperature-of(ice-cube, cold)', which is a *premise* of the relation 'cause' in the target, is mapped to the proposition 'temperature-of(ice-cube, cold)', which is a *consequence* in the source. Therefore, the whole analogy between the target problem and the situation **B** could hardly generate any useful inference.

To summarize, when the mechanisms for access and mapping worked together, the model constructed an analogy that can potentially solve the problem. On the other hand, when the two mechanisms were separated, the retrieval stage favored a superficially similar but inappropriate base. The mapping stage then worked hard to produce an acceptable set of correspondences. Still, the final result was seriously flawed.

The presentation so far concentrated on the final result produced by the model. We now turn to the dynamics of the computation as revealed by the time course of the retrieval indices. Figure 6 plots the retrieval indices for several LTM episodes during the first run of the program (i.e. when access and mapping worked in parallel).

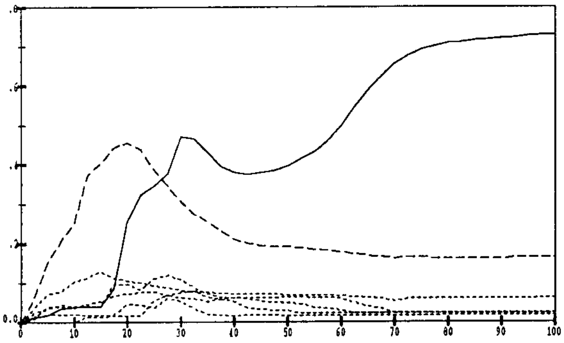

Figure 6. Plot of retrieval indices versus time for the parallel condition. Situation **A** is in solid line, **B** in dashed. The dotted lines at the bottom correspond to other situations from LTM.

This plot tells the following story: At the beginning of the run, several situations were probed tentatively by bringing a few elements from each into the working memory. Of this lot, **B** looked much more promising than any of its rivals as it had so many objects and relations in common with the target. Therefore, all agents representing situation **B** were rapidly activated and they began trying to establish correspondences between themselves and the target agents. The active members of the rival situations were doing the same thing, although with lower intensity. At about 15 time units since the beginning of the simulation, however, situation **A** (with the immersion heater) rapidly gained strength and eventually overtook the original leader. At time 30, it had already emerged as winner[3] and gradually strengthened its dominance.

The final victory of situation **A**, despite its lower semantic similarity compared to situation **B**, is due to the interaction between the mechanisms of access and mapping in AMBR. More precisely, in this particular case it is the mapping that radically changes the course of access. To illustrate the importance of this influence, Figure 7 contrasts the retrieval indices with and without mapping.

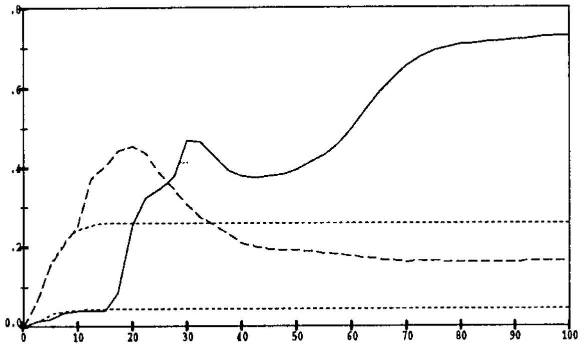

Figure 7. Retrieval indices for situations A and B with and without mapping influence on access. See text for details.

The dotted lines in Figure 7 show the retrieval indices for the two situations when mapping mechanisms are suppressed. Thus, they indicate the 'pure' retrieval index of each situation—the value that is due to the access mechanism alone. The index for situation **B** is much higher than that of **A** and, therefore, **B** was used as source when the mapping was allowed to run only after the access had finished.

In the interactive condition, however, the mapping mechanism boosted the retrieval index via what we call a 'bootstrap cascade'. This cascade operates in AMBR in

[3] The 'hump' in the graph is a side effect of the mapping mechanism which is too complex to be detailed here. In a nutshell, it involves transforming 'embryo hypothesis agents' into 'mature hypothesis agents' (Petrov, 1997).

the following way. First, the access mechanism brings two or three agents of a given situation into the working memory. If the mapping mechanism then detects that these few agents can be plausibly mapped to some target elements, it constructs new correspondence nodes and links in the AMBR network. This creates new paths for the highly active target elements to activate their mates. The latter in turn can then activate their 'coalition partners', thus bringing a few more agents into the working memory and so on.

The bootstrap cascade is possible in AMBR due to two important characteristics of this model. First, situations have decentralized representations which may be accessed piece by piece. Second, AMBR is based on a parallel cognitive architecture which provides for concurrent operation of numerous interacting processes. Taken together, these two factors enable seamless integration of the subprocesses of access and mapping in analogy-making.

CONCLUSION

The simulation experiment reported in this paper provides a clear example of mapping influence on analog access and of the advantages of the parallel interactionist view on analogy-making. Furthermore, the computational model AMBR provides a theoretical framework for explaining the controversies in the psychological data on access and reminding. It is possible to explore in which cases the interaction between access and mapping produces results different from a sequential and independent processing. It provides also a framework for generating more precise hypotheses and new experimental designs for their testing. Thus, for example, the detailed logs of the running model might me used for comparison with protocols of think-aloud experiments.

Analogy-making has certainly no clear cut boundaries. Most literature has concentrated on explicit analogies, i.e. consciously retrieving an analog and noticing the analogy. However, there are other cases which might be called implicit or partial analogies, e.g. subconsciously accessing part of a previously solved problem and mapping it to part of the target description without consciously noticing the analogy. The decentralized representations of situations in AMBR make it possible to model the process of partial access, access with distortions, blending (Turner & Fauconnier, 1995), and interference. A previously solved problem can influence the course of problem solving in an even more subtle way by priming some concepts or situations which then trigger a particular solution (Kokinov, 1990, Schunn and Dunbar, 1996). The AMBR model can be used to analyze such cases. It has already been successfully applied for predicting priming and context effects (Kokinov, 1994c).

Priming effects are an example of the influence of access on mapping which is the opposite direction of the one discussed in the current paper. Order effects are another kind of effect that goes in 'forward' direction. Such effects may be due to non-simultaneous perception of the elements of the target problem (Keane, Ledgeway, & Duff, 1994) and/or non-simultaneous retrieval of relevant pieces of information from LTM. Thus the mutual influence between analog access and mapping offers many opportunities for investigation.

REFERENCES

Falkenhainer, B., Forbus, K., and Gentner, D. (1986). The structure-mapping engine. *Proceedings of the Fifth Annual Conference on Artificial Intelligence.* Los Altos, CA: Morgan Kaufman.

Forbus K., Gentner D., and Law, K (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science, 19,* 141-205.

French, R. (1995). *The subtlety of sameness: A theory and computer model of analogy-making.* Cambridge, MA: MIT Press.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science, 7,* 155-170.

Gentner, D. (1989). The mechanisms of analogical learning. In S. Vosniadou and A. Ortony (Eds.), *Simiarity and analogical reasoning.* New York, NY: Cambridge University Press.

Gick, M.L. and Holyoak, K.J. (1980). Analogical problem solving. *Cognitive Psychology 12* (80), 306-356.

Hofstadter, D. and the Fluid Analogies Research Group (1995). *Fluid concepts and creative analogies: Comuter models of the fundamental mechanisms of thought.* New York: Basic Books.

Holyoak K. and Koh K. (1987). Surface and structural similarity in analogical transfer. *Memory and Cognition, 15* (4), 332-340.

Holyoak K. and Thagard P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science, 13,* 295-355.

Holyoak, K. and Thagard, P. (1995). *Mental leaps: Analogy in creative thought.* Cambridge, MA: MIT Press.

Hummel, J. and Holyoak, K. (1997). Distributed representation of structure: A theory of analogical access and mapping. *Psychological Review, 104,* 427-466.

Keane, M., Ledgeway, K., and Duff, S. (1994). Constraints on analogical mapping: A comparison of three models. *Cognitive Science, 18,* 387-438.

Kokinov, B. (1988). Associative memory-based reasoning: How to represent and retrieve cases. In T. O'Shea and V. Sgurev (Eds.), *Artificial intelligence III: Methodology, systems, applications.* Amsterdam: Elsevier.

Kokinov, B. (1990). Associative memory-based reasoning: Some experimental results. *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Lawrence Erlbaum Associates.

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

Kokinov, B. (1994a). The context-sensitive cognitive architecture DUAL. *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society.* Hillsdale,NJ: Lawrence Erlbaum Associates.

Kokinov, B. (1994b). The DUAL cognitive architecture: A hybrid multi-agent approach. *Proceedings of the Eleventh European Conference of Artificial Intelligence.* London: John Wiley & Sons, Ltd.

Kokinov, B. (1994c). A hybrid model of reasoning by Analogy. In K. Holyoak and J. Barnden (Eds.), *Advances in Connectionist and Neural Computation Theory. Vol. 2: Analogical Connections.* Norwood, NJ: Ablex Publishing Corp.

Kokinov,B., Nikolov,V., and Petrov,A. (1996). Dynamics of emergent computation in DUAL. In A. Ramsay (Ed.), *Artificial Intelligence: Methodology, Systems, Applications.* Amsterdam: IOS Press.

Kuhn, T. S. (1970). *The Structure of Scientific Revolutions* (second ed., enlarged). Chicago: The University of Chicago Press. (First edition published 1962.)

Minsky, M. (1986). *The society of mind.* New York: Simon and Schuster.

Mitchell, M. (1993). *Analogy-making as perception: A computer model.* Cambridge, MA: MIT Press.

Petrov, A. (1997).. *Extensions of DUAL and AMBR.* M.Sc. Thesis. New Bulgarian University, Cognitive Science Department.

Ross, B. (1989). Distinguishing types of superficial similarities: Different effects on the access and use of earlier problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15,* 456-468.

Ross, B. and Kilbane, M. (1997). Effects of principle explanation and superficial similarity on analogical mapping in problem solving. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 23* (2), 427-440.

Ross, B. and Sofka, M. (1986). [Remindings: Noticing, remembering, and using specific knowledge of earlier problems]. Unpublished manuscript.

Schunn, C. and Dunbar, K. (1996). Priming, analogy, and awareness in complex reasoning. *Memory and Cognition, 24,* 271-284.

Thagard, P., Holyoak, K., Nelson, G., and Gochfeld, D. (1990). Analog retrieval by constraint satisfaction. *Artificial Intelligence, 46,* 259-310.

Turner, M. and Fauconnier, G. (1995). Conceptual integration and formal expression. *Metaphor and Symbolic Activity, 10* (3), 183-204.

Wharton, C., Holyoak, K., and Lange, T. (1996). Remote analogical reminding. *Memory and Cognition, 24* (5), 629-643.

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 the interpretation of verbal commands with fuzzy logic and semantic networks

Christophe Brouard
Bernadette Bouchon-Meunier
LIP6
Université Paris 6, Case 169
4 place Jussieu,
75252 Paris cedex 05, FRANCE
+33 1 44 27 70 03
Christophe.Brouard@lip6.fr
Bernadette.Bouchon@lip6.fr

Charles A. Tijus
Laboratoire de Psychologie Cognitive
Université Paris 8
2 rue de la liberté
93526 St Denis Cedex 02, FRANCE
+33 1 49 40 64 84
tijus@univ-paris8.fr

ABSTRACT

This study deals with the interpretation of verbal commands for action. After an experimental study of human interpretation of instructions for drawing geometrical figures, we have devised a model whose computerized version is called SIROCO. This model represents an attempt to simulate category construction for interpretation. The use of fuzzy logic and circumstantial semantic networks allows emphasizing the importance the situation plays in completing and clarifying propositions expressed in natural language. Finally, a simulation shows quite good results for the model.

Keywords

natural language, command, action, fuzzy logic, semantic network, situation, categorisation.

INTRODUCTION

When you look at the content of verbal commands, they appear to be poor, ambiguous and elliptic. Nevertheless, they are in fact efficient as measured by the fit of actions carried out by an operator to the speaker's (the person who formulates the command) intended goal. In summary, a few words are enough to elicit complex and precise actions. How can the power of utterances be explained ?

A partial explanation lies in the fact that the operator has mental models of situations, scenarios and procedures at his disposal. These comprise a general knowledge which allows him to complete the information received, to activate other knowledge in order to understand what is being asked of him and finally, to carry out the action. When, for example, someone is asked to post a letter, he knows that the letter needs a stamp, an address, and that it should be dropped in a mail box or taken to the post office. Modelling the operator, (here, the person asked to mail the letter) calls for describing and representing the kind of general knowledge we have just described. This is what a number of recent systems have attempted to do, including CARAMEL (Sabah & Briffault, 1993) for understanding stories, CAMILLE (Hasting & Lytinen, 1994) for describing scenarios, and KA (Peterson, Mahesh, & Goel, 1994) for technical specifications.

Pragmatic explanations might also be useful in explaining the power of utterances. Sperber and Wilson's communicational implications (1986) and Grice's maxims (1975) come to mind. Thus, in the above example, lacking any indication as to the cost of the stamps, the operator might rightly assume that the letter should be sent at a standard rate; because if it were to be sent express or recommended, this very relevant bit of information would surely have been provided. Modelling the operator thus calls for integrating pragmatic rules as well as general knowledge into the comprehension system. This is what has been done with DIABOLO, a system for analysing and generating dialogue (Vilnat, 1995).

The situated action approach[1] provides a more circumstantial way of explaining the efficiency of speech. The proponents of situated action place less emphasis on the notion of internal representation and more on situational cues and action. For Olson (1970), who rejects the linguistic approach to studying the comprehension of verbal utterances, the meaning of an utterance should not be looked for in the proposition, but in the situation to which the utterance refers. This is the approach we are taking here : the power of language resides in its relation to a given situation. Important clues that allow completing vague and elliptical utterances are provided by (i) the environment, (ii) the information that has already been communicated (what we will call the "background") and (iii) the task (what must be done with the elements provided by the environment).

We thus propose that a system for interpreting verbal commands must be able to cope with the incompleteness and the imprecision of language by analysing situations. The system we have devised to do so is called SIROCO. Though it is currently outfitted to interpret verbal commands for drawing geometrical figures, it could be adapted to interpret other kinds of verbal commands. We have used it to study how operators interpret commands and make decisions. In the case of incompleteness, the system has to identify the instructor's intended categories. In the case of imprecision, it has to define the fuzzy boundaries of the categories. To this end, we used two tools for representing information that is incomplete or imprecise, namely: circumstantial semantic networks and fuzzy subsets.

The study we present here was done in three phases: An experimental phase in which a human subject-operator was asked to interpret and carry out instructions for drawing geometrical figures given in natural language by a subject-instructor. The second phase consisted in designing a

[1] See Norman (1993), for an introduction to this situated action approach .

model of the subject-operator. Finally, a simulation allowed comparing SIROCO's responses to those of the subject-operator.

EXPERIMENT

Objectives

The aim of this experiment was to provide empirical data on the degree of precision with which people interpret verbal commands for drawing geometrical figures. More importantly, it aimed at providing information on how missing information is completed and, more generally, on how concrete situations influence the precision with which a command is carried out. All data relative to instructor commands and operator actions was collected automatically to provide a precise record of input and output for the simulation.

Method

Participants

Thirty five instructors were recruited from the undergraduate population of the University Paris 8, St Denis-Vincennes. A single operator was recruited from the same population, his responses provided the data we analysed.

Materials

A set of 35 drawings (8.2 cm large and 14.8 cm high), one for each instructor, were created with a drawing software. Each drawing was composed of three simple geometrical figures. The set was designed to provide a wide range of property combinations for the geometrical figures. The different figure-properties were: rectangle, circle and square, for the shape; red, green and blue, for the color; small, medium and large, for the size (from 1 cm up to 6.2 cm for width, from 0.6 cm up to 6.46 cm for height); top, center and bottom for the vertical position (from 0.01 cm up to 11.65 cm on the Y coordinate); and, finally, left, middle and right for the horizontal position (from 0.31 cm up to 5.98 cm on the X coordinate). The complexity of these combinations, from the point of view of the corresponding lattice (see next section), was maximal in all cases. In other words, any two geometric figures have both common and distinctive features.

The computer apparatus consisted of two large monitors placed back to back on a long table (fig. 1). Thus, the instructor and the operator, each behind a monitor, were hidden from each other. The instructor could only communicate through verbal commands, the operator could not see the original drawing the instructor had in his hand.

General procedure

Each one of the thirty five drawings was given to an instructor. The instructor was asked to make the operator reproduce this picture through verbal commands only. The operator, who was not allowed to speak to the instructor, typed each verbal command[2] he received into the word processor and then carried it out. The graphic interface on which the operator worked was of the same size as the instructor's picture. On his screen, the instructor saw what the operator was drawing. After the operator had finished carrying out a command, the instructor could correct the drawing with a new verbal command and so on, until the instructor was satisfied with the drawing the operator had produced.

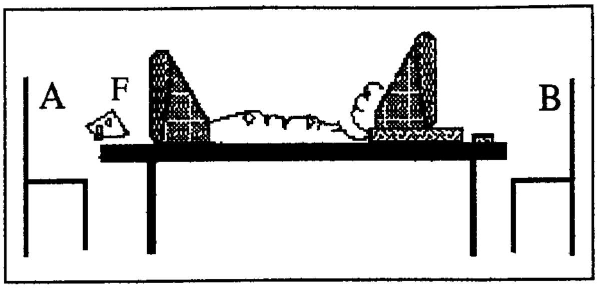

Figure 1. The instructor was placed in A and had in his hands a drawing (F). The operator was placed in B.

Automatic data collection

All action related to writing verbal commands (on the word processor) and drawing figures (on the graphic interface) was recorded with "spy" software.

Results and discussion

The Power of utterances

On average, 9 commands were necessary for a satisfactory reproduction of the original drawing. The minimum was 4, the maximun was 18 for a single drawing. On average, 3 commands were required for reproducing each figure. More precisely, 2 commands were sufficient to correct the first attempt to draw a figure. This may seem very few when one considers that there were four continuous factors which defined each figure (size, shape, vertical and horizontal position).

The precision of commands for discrete properties

There were just a few lateralisation errors ("not on the left, I said on the right"). Though information on size and color was not always given, these were correctly reproduced by the operator. The semantic structure of the properties of the figures already in place (see MODEL section) did allow completing the missing property information in a command. Thus, our hypothesis was globally satisfied. Nevertheless, the operator seemed to hesitate between an average value and a value inferred from properties already in place. The effect of the extracted regularity of the properties already in place would certainly have been greater if the objects were more numerous (in our experiment there are only three figures per drawing).

The precision of commands for continuous properties

From a statistical point of view, there were no significant differences between the figure-values for the continuous properties of the operator's finished drawings and the corrresponding values of the original drawings: for the X coordinates of the figure's top-left corner $p>.96$, for the Y coordinates (of the same point) $p>.17$, for width $p>.94$ and for height $p>.08$. The correlation between the operator's drawings and the originals one was .86 for the X coordinate ($p<.0001$), .93 for the Y coordinate ($p<.0001$), .70 for W (width) and .86 for H (height)

[2] Verbal commands were expressed in French. For the purpose of this article we translated some of them.

(p<.0001).The average correlation for each of the first seven commands is given in table 1.

Table 1: The average correlation for each of the first seven commands

	C1	C2	C3	C4	C5	C6	C7
X:	.75	.95	.70	.80	.91	.91	.91
Y:	.88	.99	.76	.97	.98	.94	.97
W:	.96	.76	.27	.55	.53	.96	.50
H:	.94	.93	.80	.50	.93	.97	.88

It is clear that the operator faithfully reproduced the original values quite rapidly, because from the first try on, the commands were executed with an overall precision of 4% for X, 1% for Y, 3 % for W and 2% for H. When the operator's figures did not fully correspond, a few more verbal commands were all that was needed to correct them. In summary, long-distance geometric figure drawing in this experiment was extremely precise.

MODEL

The results of the experiment show that the situation is indeed an aid in interpreting commands. The present model replicates the way in which the situation provides information by taking advantage of the dynamicity of circumstantial semantic networks and the flexibility of fuzzy subsets.

General Description

For SIROCO, interpreting verbal commands means using the situation to construct the instructor's intended categories. Thus, when one or more of a figure's properties is not explicitly indicated, it is inferred from the property network based on the figures that have already been drawn. Often property-categories are specified with absolute utterances such as "large" or "at the top" but sometimes compound utterances such as "rather square" or "smaller" are used. The meaning of compound utterances must be constructed from the meaning of the absolute ones. Indications and corrections given prior to a new command (the background) must also be taken into account. Finally, all of this information is represented in the form of fuzzy subsets and integrated through a procedure which aims at finding the solution that best satisfies all the constraints including space constraints.

Incompleteness processing with circumstantial semantic networks

The propositional meaning of an instruction is first analysed as to the objects and their associated properties. Subsequently, objects and properties are used in order to construct a semantic network which reflects an understanding of the proposition (Zibetti & Tijus, 1997; Poitrenaud, 1995).

In this network, properties shared by several objects are grouped together in order to constitute categories (figure 2). The underlying mathematical structure of this property network is the Galois lattice (Barbut & Monjardet, 1970). This network allows different logical operations. For example, if among different geometrical coloured figures, all the squares are black, it is possible to predict the black property from the square property because of the inherited properties of the square category in the semantic network. Otherwise, in certain context an object can be designated by a single property such as "the white one" in figure 2 to refer the white circle, or as "the black one" to refer the black circle because given that there are two black objects, the instructor might want to designate a figure that differs from others in the same category by being black.

Finally, the lattice allows some operations which can explain and simulate categorisation processes (Tijus & Moulin, 1997). For example, it is always a problem to categorise an incompletely described new object. A good solution (from the point of view of modelisation) consists in choosing or constructing a category that alter the structure of the network as little as possible. For example, if a white square has to be drawn, without any specification as to its size, in the situation described in figure 2, it will be small. More generally, this semantic network represents the property structure of a given situation which can be very useful for modelling (Richard, J.F., Poitrenaud, S., & Tijus, C.A., 1993).

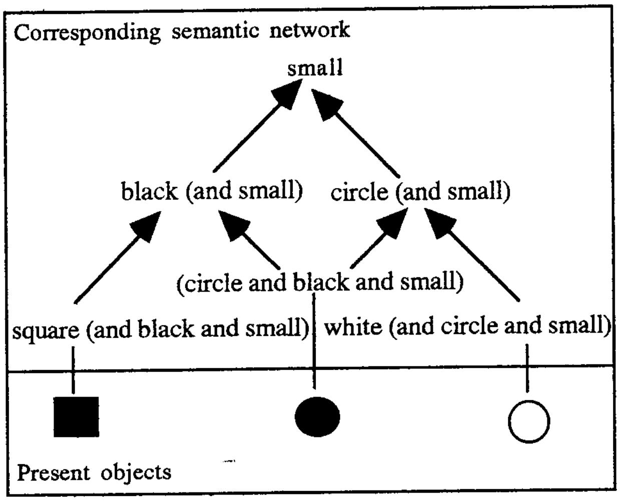
Corresponding semantic network
small
black (and small)
circle (and small)
(circle and black and small)
square (and black and small)
white (and circle and small)
Present objects

Figure 2. Example of a semantic network constructed from the object properties of the situation.

Representing a command with fuzzy subsets

A drawing command specifies size, shape, colour and position categories. Except for colour categories which are precisely defined (there is just one kind of blue, green and red available), the other kinds of categories (for example, large, rectangle) have imprecise boundaries. Thus an element (like a value corresponding to a surface in square centimetres) can have an intermediate degree of membership between 0 and 1 in a category. So we have chosen to represent these categories with a fuzzy subset. The concept of fuzzy subsets (Zadeh, 1965; Bouchon-Meunier, 1995) is a generalisation of the concept of sets. A fuzzy subset is characterised by its membership function (figure 3).

An important issue lies in the choice of reference variables (in figure 3, the choice is surface area as measured in square centimetres). This choice has to be made such that the variable is well suited to determine whether or not an element belongs to the represented category. Ideally, this variable has to correspond to a psychologically relevant perceptive dimension. Psychophysics, which studies the relations between physical and perceptive dimensions,

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

could provide this kind of variable. As we are focusing upon general principles, we chose simple variables (like surface area for size categories, and abscissa for horizontal position) and trapezoidal fuzzy subsets.

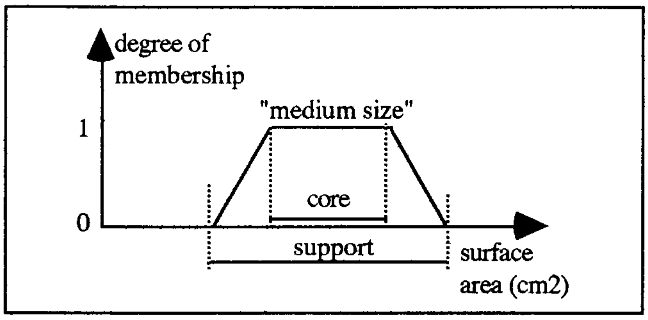

Figure 3. The membership function for a fuzzy subset representing "a medium size" category. Note that the "core" is comprised of elements which belong to the fuzzy subset with a membership degree equal to 1, and that the "support" is comprised of elements which belong to the fuzzy subset with a non-zero degree.

We represent a command by associating a fuzzy set to each dimension of the description. Zadeh (1975) introduced the concept of linguistic variables which consist of a variable, a universe in which the variable is defined (real numbers for example), and a set of fuzzy subsets which represent different characterisations of the variable (for example, small, medium and large for a size variable).

Here we use four linguistic variables to represent a command: (i) the size which is the surface area of the figure and which is characterised by "small", "medium" and "large", (ii) the elongation which is the width/height ratio and which is characterised by "upright" "equal" and "reclining", (iii) the horizontal position on the abscissa which is characterised by "left", "middle" and "right" and, finally, (iv) the vertical position which is on the ordinate and which is characterised by "top", "centre" and "bottom". Two discrete variables complete this representation: colour which can be blue, green or red and shape, which can be rectangular or elliptical.

Because there is an odd number of characterisations for each variable (exactly three), there is always a central category. Moreover the fuzzy subsets that represent these characterisations are such that they constitute a fuzzy partition of the universe. Which means that for each element, the sum of its membership degrees in all the different characterisations for a given variable is 1. Thus the slopes of the trapezia intersect at midpoint (see figure 4).

Applying a linguistic modifier

Our aim here is to represent utterances like "very large" or "toward the left", that is to say modified versions of categories. Zadeh (1972) associates to each linguistic modifier ("very", "rather",...) a mathematical transformation which allows constructing new fuzzy subsets from initial ones. The initial fuzzy subset represents an initial category ("large"). The new fuzzy subset represents a modified version ("very large") of the initial category.

Since Zadeh's pioneering work, numerous new modifiers have been introduced. Here, we use modifiers (Bouchon & Yao, 1992) which exploit the distribution of defined categories in a single universe (size, for example). The mathematical transformation corresponds to a shift whose amplitude and direction can be deduced automatically. We chose them for the way they can readily be applied to all different kinds of properties (see figure 4).

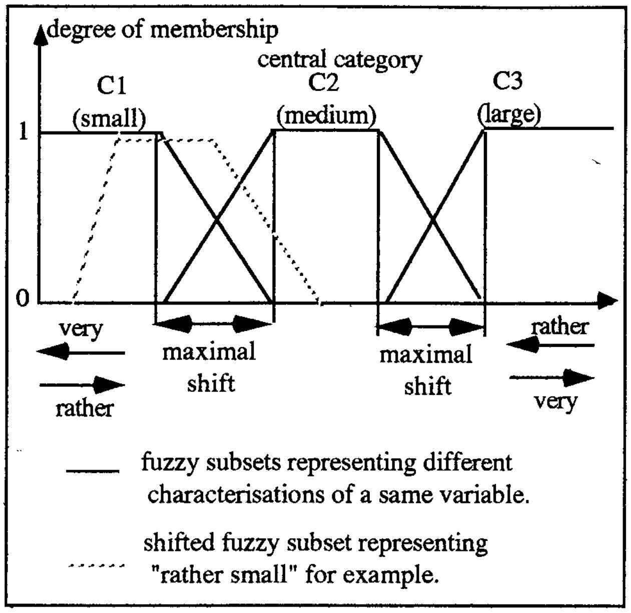
degree of membership
central category
C1
(small)
C2
(medium)
C3
(large)
1
0
very
rather
maximal shift
maximal shift
rather
very
fuzzy subsets representing different characterisations of a same variable.
shifted fuzzy subset representing "rather small" for example.

Figure 4. Illustration of linguistic modifier mechanisms.

From a given characterisation and a given modifier, simple mechanisms yield the shift to be applied. Thus, for modifiers like "very" the direction of the shift is toward an extreme and for modifiers like "rather" the direction is toward the centre (figure 4). The amplitude of the shift is defined as a proportion of the maximal shift which corresponds to the distance between initial category cores. Thus a modified category will never overlap upon a neighbouring category. Moreover, the maximal shift automatically defines a scale regardless of the type of variable. Finally, it is possible to use modifiers of different strengths. Thus "very very" is a modifier of the same kind as "very" but the amplitude of the shift associated to it is larger. To be more precise, the coefficient associated to "very very" is larger than the one associated to "very".

Applying a fuzzy relation

Utterances like "larger" or "a little bit less to the left" can be represented with fuzzy relations. The concept of fuzzy relations (Zadeh, 1971) is a generalisation of the concept of relation as it allows intermediate degrees (between 0 and 1) of relation between elements. It corresponds again to a fuzzy subset. In contrast to the preceding case concerning modifiers, this fuzzy set will not be constructed from a fuzzy set but from a value (the surface area of the figure, if the command is "larger"). We can divide this kind of command into two parts: the relation part which is, for example; "much more", "less" ,or "same" and a category part which is, for instance, "on the left" or "large".

It is possible to define mechanisms such that from a given relation and category, the fuzzy set representing the utterance can be constructed. First, after having defined a sign for a relation ("less" relations will be negative and "more" relations, positive), a category (positive categories are to the right of the central category, negatives are to the left) the direction (increasing, decreasing) indicated by the utterance (for example, "less big") is calculated by multiplying the relation sign and the category sign. When the category is the central one, the direction depends on the position value compared to the middle value of the category (when the command, for example, is "rounder", the question to be asked is whether the figure is an upright or a reclining ellipsis). Like modifiers , different coefficients are associated with each relation expressing a different strength. "Much more" indicates a stronger variation than "more" (figure 5).

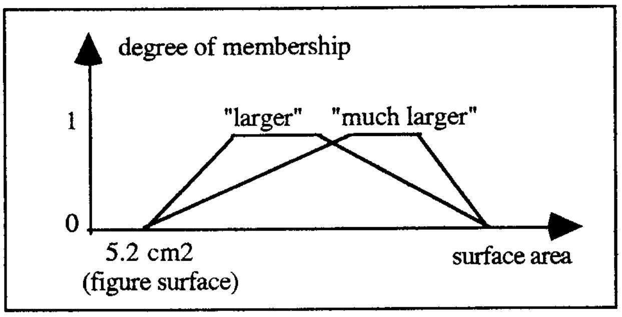

Figure 5. Illustration of fuzzy subset construction for utterances containing "larger" and "much larger" (than 5.2 cm2).

Softening inferred categories

As we mentioned above, when no characterisation is specified for a given variable in a drawing command, it is inferred from the semantic network. This tacit information is not as constraining as explicit information. We therefore chose to represent it by allowing all values, that is to say, by taking a support (for the constructed fuzzy subset) equal to the entire universe of the variable. Moreover, taking the results of experimentation into account we softened the inferred category by applying a modifier.

Background communication

At any point in a verbal exchange involving commands, what has already been said and done constitutes the background communication so decisive for interpretation. For example, what "larger" means can vary according to whether it is an initial correction whose aim is to get the operator to draw a figure of roughly the right size or whether it is a final correction aimed at precision. The background thus allows the commands to be interpreted with increasing precision. Indeed, without background, instructions like "a little bit larger" followed by "a little bit smaller" would consist of nothing more than commands for switching back and forth from an initial value.

Background construction

During communication, various indications and corrections are given. This can be represented by a list of slopes of different constructed trapezia in the prior commands. For each variable, there is one background. Fixing a maximal length for this list allows taking the operator's limited memory into account.

Making background operational

Only two slopes are useful for each variable. They correspond to the more restrictive constraints (right, left constraints could be for instance, respectively, much smaller than 10.3 square centimetres and larger than 5.4 square centimetres) and allow constructing a fuzzy set. So, background is accounted for by intersecting this last fuzzy subset with the current command associated to the fuzzy subset. When this intersection is small (under a given threshold), we can decide to forget the background in order to produce an appropriate response despite contradictory commands.

Choosing an appropriate solution

Choosing a relevant point

According to the specified position in the drawing command, the relevant point varies. For example, if the command calls for drawing a figure at the top left corner, the top left corner of the figure is the relevant point. Which means, that it is the point which will be taken into account for characterising the figure's position (in the example, the more the top-left corner of the figure is at the top and to the left, the more the position of the figure is acceptable). If the command is "To the left of the square, draw a ....", the right and centre (vertically) point is relevant.

From the 3 vertical position characterisations and the 3 horizontal position characterisations, we defined 9 relevant points. Relevant points allows simplifying the decision procedure. We could have chosen a more sophisticated variable that might have been more psychologically relevant, but as we are focusing upon general principles, we did not do so.

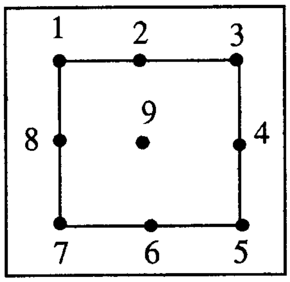

Figure 6. The nine possible relevant points of a figure.

Defining the degree of acceptability for all points of the drawing area.

For a point p of the drawing area, the acceptability degree is computed by aggregation of two intermediate degrees $d_1(p)$ and $d_2(p)$. We chose the min operator for expressing conjunctions:

$$d(p) = \min (d_1(p), d_2(p)),$$

where $d_1(p)$ indicates the degree to which point p (the relevant point) is a good point from which to begin drawing the figure specified in the command ("in the top-left corner" or "near the circle") and where $d_2(p)$ indicates the degree to which it is possible to place at p a figure of the size and shape corresponding respectively to the size and the shape of the characterisations of the command. It is computed as :

$d_2(p) = \sup\{(\min(\mu_{size}(l), \mu_{shape}(l,h)) / 0 < l < lmax, 0 < h < hmax(l)\}$,

where lmax and hmax are respectively the largest possible width and height taking into account the figures already present and μ_{size} and μ_{shape} are respectively the membership functions of the size and shape fuzzy subsets constructed from the command (figure 7).

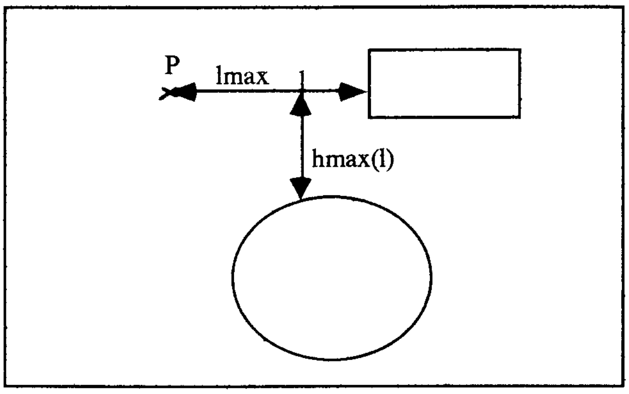

Figure 7. Illustration of the d2(p) calculation when p is the top-left corner of the figure.

Computing d(p) for all drawing-area points allows defining favourable areas (figure 8).

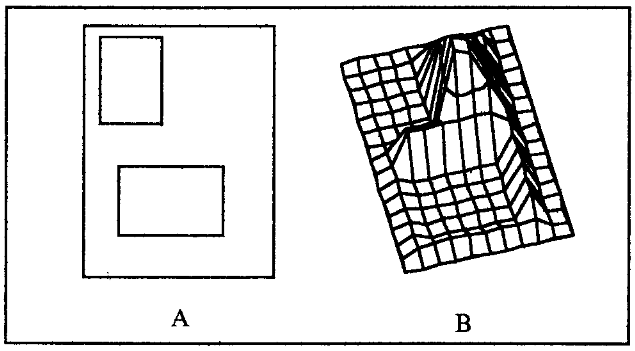

Figure 8. A) Figures already present.B) Visualisation of favourable areas for drawing "a large circle at the centre of the drawing area".

A solution suited to the situation

The general optimisation procedure allows choosing a solution suited to the situation without explicitly describing the situation beforehand (figure 9).

VALIDATION

The above model has been computerised and called SIROCO. This system allowed simulating the operator-subject in order to validate the model by comparing system responses to the operator responses.

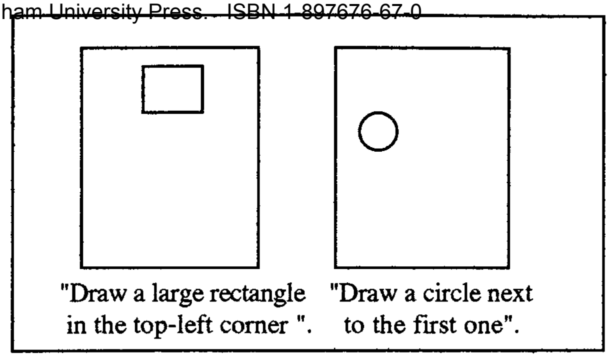
"Draw a large rectangle in the top-left corner ".
"Draw a circle next to the first one".

Figure 9. In these two situations, the optimisation procedures decide respectively to draw an upright rectangle and a circle to the right of the first one.

Model parametrization

The experiment provided thirty five communication records. Ten records were kept in order to test the model. The others were used for teaching the fuzzy subsets of the different characterisations, the modifiers and relation parameters to the system. More precisely, the first drawings for each communication (which correspond to a minimal context) allowed defining the cores for all characterisations. Supports were then defined in order to construct a fuzzy partition for each variable (see above). Analysing experimental results allowed defining modifier and relation coefficients. Relation coefficients express similarity, these similarity relations are not necessarily linear. For example, for an equal difference of surface, the smaller the two compared surfaces are, the more they are perceptively different. However, we considered these relations to be linear, and chose average coefficients because the experimental material did not allow inferring their exact shape[3].

Simulation

A description of SIROCO

Developed in C++, SIROCO includes a graphic interface for visualising system and subject drawing responses. It also allows running a commands file, typing commands interactively and readjusting the system's responses to the subject's responses at will. Finally, it allows visualising favourable drawing areas (by creating a matlab file).

Definition of a minimal language

The commands that were kept in order to test the model were translated into a minimal language with a limited number of words and with a strict structure[3]. Most of these words indicate the linguistic variable characterisations, and also, the modifiers and relations often used in commands. This language aims at representing commands without interpreting them. For example, "nearer the edge" is not translated as "more

[3] We should define, as in FILIP (Zemankova, 1989), these relations from the outside of the system.

[4]The commands which could not accurately be translated into the minimal language were excluded from the results. Variables and objects (like edges) need to be added to the language to make it more expressive. However our translation tables do allow expressing most of the commands.

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.

toward the left" (if the figure is near the left edge) but by "more extreme" ("extreme" is automatically replaced with the object category, in this case, the "left"). Likewise, "make the rectangle longer" is translated by "more extreme". Thus, if the shape category is upright, the height will be increased. If the category is reclining, the width will be increased. We also use the OR connector in order to express utterances like "rectangle" (an upright rectangle OR a reclining rectangle) or "next to" (to the left OR to the right of another figure).

Simulation with readjustment

For this simulation, the operator-subject comparison was made command by command. Each of the system's responses was automatically readjusted to the subject's responses just before the next command was interpreted. The communication background was also readjusted. Thus, for each new command, the system was placed in precisely the same interpretive situation as the subject.

Results and discussion

In order to evaluate model validity, we compared the figures the human operator drew with the ones the system drew. More precisely, we compared the X and Y co-ordinates of the figure's top-left corner, the width W and the height H. We measured the error margin and the correlation for each of these variables.

Table 2: The average error margin in centimetres for the first seven commands.

X:	Y:	W:	H:
0.25	0.35	0.30	0.28

Compared to the figure variance in the initial drawings, there is no significant difference between subject and system drawings for the X coordinate (p>.2) and for width W (p>.32). On the other hand, we found differences for the Y coordinate (p=.02) and for height H (p <.01). Positions and sizes have a very important correlation: .93, .84, .93 and .92, respectively for X, Y, W, H. From the first to the third figure, the correlation is shown in table 3.

Table 3: The correlation from the first to the third figure.

X:	.94	.93	.91
Y:	.93	.98	.30
W:	.91	.90	.97
H:	.83	.90	.97

We can see that simulation becomes more and more precise as communication progresses (table 4).

Table 4: The correlation from the first to the seventh command.

	C1	C2	C3	C4	C5	C6	C7
X:	91	.94	.97	.97	.95	.92	.96
Y:	.86	.79	.8	.99	.99	1	1
W:	.35	.9	.97	.80	.86	.86	.99
H:	.67	.84	.93	.85	.95	.97	.99
av:	.70	.87	.92	.90	.94	.94	.98

The small error margin with which the system operated might be, but is not necessarily, due to the model. The system chose one solution from a set of equally possible solutions and, under the same conditions, a human operator might also give different responses. In order to explain these differences, we should add that the system, as opposed to a human subject, does not make mistakes and does not forget information. Overall, the response given by the system is always acceptable and it is difficult to distinguish it from the human operator's response. Otherwise, softening the category the system had infered from the semantic network also gave good results.

GENERAL DISCUSSION

A small number of combined cues are enough to enable us to define a precise solution. Other more elaborate experiments could reveal other important cues. Even in the particular case of this experiment, we do not pretend to have tackled all the facets of command interpretation. Category learning (Omri, 1994), that is to say the adjustment of interlocutor categories, is not taken into account here. Nonetheless, its affect would probably have been insignificant because communication between the operator and the instructor took place very quickly (the instructor was replaced for each new drawing).

As we mentioned in the introduction, our study is about a particular contextual explanation of the power of language. Thus, some implicits of communication were not taken into account, whereas their effects were not negligible from the point of view of the results. For example, when the command was to draw a figure on the left and there already was a figure on the left, the system chose to place the new figure very near the first one (it placed it as far to the left as it could). The implicit information in this command is that the two figures can not be stuck together, because if they were, this information would be given. To explain this kind of implicit principle of relevance introduced by Sperber and Wilson (1986) seems well suited. It could be implemented with semantic networks and fuzzy sets.

In summary, we have shown here a set of mechanisms for constructing the meaning of utterances from the basic category meanings. We have associated fuzzy subsets with semantic circumstantial networks and it appears that these representational tools are complementary as they cope with two different kinds of knowledge imperfection (imprecision and incompleteness) (Bouchon-Meunier, 1992). We could talk about "fuzzy semantic networks" even if category inclusion is not gradual as in Rossazza's networks (1992).

Unlike Hersh and Caramazza (1976), we are not only interested in representing the meaning utterances, we wanted to make it work, which is much more challenging. The method we follow, first, determination of fuzzy meaning for a set of variables, and second, definition of a solution maximizing the satisfaction degree of all variable constraints and integrating all environnement constraints, seems well adapted to model action. Compared with a rule system where the rules have to cover all situations and have to be explicited, this method appears more adaptative and more simple to implement, the main work consisting in constructing variables.

CONCLUSION

The aim of this interdisciplinary study was double. On the one hand, our goal was to model the processes of command interpretation (through cognitive psychology) and on the other hand, it was to create a system capable of responding consistently to verbal commands, of detecting implicit information and of adapting itself to a given situation (through artificial intelligence). These two aspects of the study are by no means opposed because devising a system that models a human subject has every chance of being a system whose behaviour is adequate. This is all the more true given that verbal communication is a specifically human activity.

There already exist certain mobile remote control apparatuses, equipped with a camera, for inspecting places that humans, for one reason or another, cannot enter. The operator who controls the apparatus must constantly specify the angle and speed at which the apparatus moves. Though the interface may be user friendly and, for instance, allow guiding the apparatus with a joystick rather than explicitly indicating angle and speed, there are still disadvantages. Namely, the constant supervision that the system requires calls for technical mastery as well as taxing levels of alertness and watchfulness on the part of the human operator. These disadvantages could be partially compensated for by redesigning the system to respond to natural language.

AKNOWLEDGEMENTS

This research was funded by DGA/DRET Contract N° 93-166. The authors express their thanks to E. Hamilton for her assistance in research and translating.

REFERENCES

Barbut, M., & Monjardet, B. (1970). Ordre et classification: algèbre et combinatoire. Paris: Hachette.

Bouchon-Meunier, B. & Yao, J. (1992). Linguistic modifiers and gradual membership to a category. International Journal on Intelligent Systems, 7, 25-36.

Bouchon-Meunier, B. (1995). La logique floue et ses applications. Paris: Addison-Wesley France.

Bouchon-Meunier, B. (1992). Représentation et traitement de l'incertitude, Journées Science et Défense, La Villette.

Grice, H.P. (1975). Logic and conversation. In P.Cole and J.L. Morgan (Eds.). Syntax and semantics, Vol 3: Speech acts, New York: Academic Press.

Hasting, P.M., & Lytinen, S.L. (1994). Objects, Actions, Nouns, and Verbs. In Ashwin Ram & Kurt Eiselt (Eds.). Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society.(pp. 397-402). Hillsdale, NJ: LEA

Hersh, H. M. & Caramazza, A. (1976). A Fuzzy Set Approach to Modifiers and Vagueness in Natural Language. Journal of Experimental Psychology: General, Vol 105, 3, 254-276.

Norman, D. A. (1993). Cognition in the head and in the world : an introduction to the special issue on situated action. Cognitive Science, 17, 1-6.

Olson, D. R. (1970). Language and Thought: Aspects Of a Cognitive Theory Of Semantics. Psychological Review, 4, 257-273.

Omri, M-N. (1994). Système interactif flou d'aide à l'utilisation de dispositifs techniques : le système SIFADE. Thèse, Université Paris VI.

Peterson, J, Mahesh, K., & Goel, K. (1994). Situating natural language understanding within experience-based design. International Journal of Human-Computer Studies, 41, 881-913.

Poitrenaud, S. (1995). The Procope Semantic Network : an alternative to action grammars. International Journal of Human-Computer Studies, 42, 31-69.

Richard, J.F., Poitrenaud, S., & Tijus C.A. (1993). Problem-solving restructuration: elimination of implicit constraints. Cognitive Science, 17, 497-529.

Rossazza. J-P. (1992). Une représentation centrée objet. Deuxième Congrès National sur les Applications des Ensembles Flous, Nîmes.

Sabah, G., & Briffault (1993). CARAMEL: step towards reflexion in natural language understanding systems. Actes IEEE International Conference on Tools with Artificial Intelligence, Boston, 258-265.

Sperber, D. & Wilson, D. (1986). Relevance : Communication and cognition. Cambridge, MA: Harvard University Press.

Tijus, C.A., & Moulin, F. (1997). L'assignification de signification à partir de textes d'histoires drôles. L'Année psychologique, 33-75.

Vilnat, A. (1995). STANDIA: A pragmatic-driven Man-Machine System, the structure of multi-modal dialogue. Amsterdam: North-Holland.

Zadeh, L.A. (1965). Fuzzy sets. Information and Control , 8, 338-353.

Zadeh, L.A. (1971). Similarity relations and Fuzzy Orderings, Information Sciences, 3, 177-200.

Zadeh, L.A. (1972). A Fuzzy-Set Theoretic Interpretation of Linguistic Hedges, Journal of Cybernetics, 2, 2, 4-34.

Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning. Information Sciences, 8, 199-249, 301-357.

Zemankova (1989), FILIP : a Fuzzy Intelligent Information System with Learning Capabilities, Information Systems, 14, 6, pp. 473-486.

Zibetti, E., & Tijus, C.A. (1997). l'effet des propriétés d'objet sur l'interprétation de l'action. Actes du Colloque Sciences Cognitives JIOSC 97: La perception: du naturel à l'artificiel. Orsay: 1er et 2 décembre 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

Chess Players' Early Recall of Chess Positions: An Empirical and Simulative Investigation

Pertti Saariluoma
Cognitive Science
University of Helsinki
P.O. BOX 13, Helsinki, Finland
+358 9 191 23458
psa@utu.fi

Tei Laine
Cognitive Science
University of Helsinki
P.O. BOX 13, Helsinki, Finland
tei@iki.fi

ABSTRACT

Novice acquisition of skilled recall of chess positions was studied in an experiment in which two novices studied a series of five hundred chess positions during a period of several months. They spent fifteen minutes to half an hour a day teaching themselves chess positions. As a result their skills in recalling chess positions rose from an average sixteen percent to somewhere between forty to fifty percent. The learning curve proved to be a logarithmic function in which learning is very fast at first but after some 100-150 studied positions the speed of learning decreases substantially.

A computer simulation was used to model the results. Two alternative ways of thinking were tested. In the first model chunk construction was assumed to be based on neighbourhood of associated pieces. The second model assumed a frequency based correlative association process. Although the learning curves of the two models are very similar by shape to that of the subjects, the frequency based associative model gave better explanation for the data. This is why it is natural to suggest that common co-occurrence is one mechanism in associative processes during chess players learning of chess specific chunks.

Keywords

Cognitive modeling, novice skill acquisition, chunking

INTRODUCTION

One can argue that research on chess players' memories is relevant only when the top level skills are considered. When all the basic skills training is focused on people that are very far from having ten years experience in the field, it should be interesting to investigate what are the major properties relevant to early learning in chess. The first hours of chess training are close to any basic course in some symbolic subject matter. Therefore, it would be good to pay more attention to these earliest stages of information processing. An important shift to the direction of early learning was made by Fisk and Lloyd (1988) when they studied the acquisition of skilled visual search in chess with absolute novices.

Fisk and Lloyd's (1988) study showed that a skill develops very rapidly at first, but later the speed of learning decreases substantially. By studying some later stage of skill development, they could not have made this observation (see also Newell and Rosenbloom 1981, and Rosenbloom and Newell 1987 for parallel findings in different task environments). If a similar pattern of skill development to the one concerning the reaction time results of Fisk and Lloyd (1988) could be found in chess recall task, it might explain why the development of skilled memory takes so much time. Though it is easy to achieve one level, each new step takes more and more effort.

To help resolve the problems above, two students with only very elementary knowledge in chess were asked to study hundreds chess positions ten to twenty minutes a day for four to six months in order to recall the positions as well as they can. The development of their recall was tested several times during this period. The aim was to determine the form of the learning curve for a later simulation analysis.

By using computer simulation we wanted to study the nature of the chunking mechanisms in early learning. Chase and Simon (1973) suggested that a number of chess specific relations such as colour, kind, threat, defense, and proximity are important in chunk construction. Here, we are interested in an even simpler factor. This is general associativity, and a good approach to it is to use a simple correlative measure. If the pieces that commonly co-occur are used in building new chunks (the idea that general associativity is important), one should get the best fitting simulation outcome by chunking pieces with high correlation.

In the next section the settings and the results of experiments in which two novice chess players learnt and tried to recall game and random chess positions are described. In the following sections a computer simulation designed to model the experimental conditions and results are presented and analyzed.

METHOD

Subjects

Two graduate psychology students participated in the experiment. One was a woman, NT, who had played only few games of chess in her life. The second subject was a man, MQ, with the same background, who had played chess a little more often, but he had neither chess ambitions nor qualifications. Neither of the subjects had ever visited a chess club or participated in a chess competition. Both were thus absolute novices.

Task and Procedure

NT, the first subject, was asked to study five hundred middle game positions from a book of combinations. She studied four to five positions for approximately fifteen minutes a day. When studying the positions she put the pieces on the board according to the illustration and tried to learn the location of each piece. She concentrated, however, only on the patterns and did not study the moves at all. She was tested five times: before the experiment began, after 110, 250, 365, and 500 positions. Her involvement in the whole experiment lasted about four months.

The experiment involving MQ was made a couple of months after the end of the experiment with NT. This second experiment took six months as MQ wanted to spend more time per a position than NT. He also studied five hundred middle-game positions from a book of middle game combinations and his method of study was the same as NT's. It was possible to test MQ somewhat more frequently than NT. His recall was tested eight times: at the beginning, after 30, 60, 175, 220, 270, 350 and 500 studied positions. The irregularities in testing intervals were due to certain practical problems involved in running this long experiment such as compulsory exams, Christmas leave, etc. Each test consisted of a standard de Groot (1965, 1966) experiment. Subjects were shown ten game and ten random positions with 18 to 28 pieces in each. The presentation time was five seconds per position and the presentation order was random. The positions presented in the various testing sessions were always different.

The test positions were made by using chess print transfers which were then photographed as slides. They were shown with a slide projector. The subjects sat at a distance of 150 cm from a display. The size of chess boards on the display was 40 × 40 cm.

RESULTS AND DISCUSSION

The results of the experiment are presented in figure 1. The x-coordinate represents the number of studied positions and the y-coordinate the percentage of recalled pieces in a test session. The percentages are the mean percentages of correctly placed pieces calculated for each test session.

The effect of learning is clear. The subjects were able to increase their percentage of recalled pieces from roughly fifteen to somewhere between forty and fifty percent, which was a rise of 25-35 percentage points. However, in recalling random positions the effect was substantially smaller averaging about five percentage.

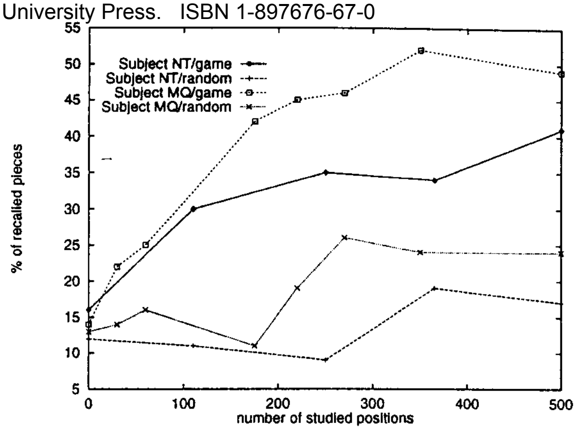

Figure 1: Percentage of recalled pieces when the subjects were tested 5 or 8 times during learning.

The profile of learning curve was very similar for both subjects. When studying the first hundred to one hundred and fifty positions they achieved most of the total increase in recall percentages. The increase was far slower from two hundred positions onwards. There was also some increase in recalling random positions, but the profile is very different from game positions, the increase being more linear throughout the whole learning period.

The learning curves of NT and MQ have a standard form. They are like many other learning curves: at first it is very steep reflecting a sharp improvement in learning. However, after a short period of time the speed decreases and the gain in performance level becomes smaller per training unit (Fisk and Lloyd 1988, Newell and Rosenbloom 1981, Rosenbloom and Newell 1987). This kind of curve can be called negatively accelerating or logarithmic.

SIMULATIVE ANALYSES

The results of the experiment provide very rough information. As such they do not tell very much about chunking as a method of learning. However, by using a computer simulation it would be possible to associate the previous experimental data with some other information about chess players' chunking. Thus the present experiment may be used for a theoretical discussion of some aspects of chess players' information chunking and to estimate the number of chunks needed for very high performance.

Several properties of chunks and chunking in chess which should be taken into account in any attempt to model chess players' recall, have been noted during the last twenty five years. The main function of chunking is to avoid the capacity limitations of human working memory. The more and larger the chunks a chess player has in his long term memory, the greater the probability of him being able to avoid the limits of his working memory and achieve a high level of performance.

This should not, however, be interpreted so that the

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

chess players' working memory is seen as a box with a few slots. The reason for capacity should rather be sought in the integration of information as the chess positions are stored in the long-term memory rather than in working memory (Charness 1976, Frey and Adesman 1976, Lane and Robertson 1979, Lories 1987, Simon 1976).

Chase and Simon (1973) observed that skilled chess players do not necessarily recall more chunks than novices, but the sizes of their chunks are larger. The increase in chunk size is not an all-or-nothing phenomenon but it rather takes place incrementally. Good players do not learn new and longer chunks at one time but their chunks slowly lengthen and their recall improves (Chase and Simon 1973, Newell and Simon 1972). This is also the way chunk learning is assumed to occur by some theories of cognitive skills (Newell and Rosenbloom 1981, Rosenbloom and Newell 1987).

In addition to chunk size the factors behind the coherence of chunks have deemed important. Chase and Simon (1973) found five chess-specific relations (same kind or colour, threat, defense and adjacent locations on board) that increase the probability of successive pieces to belong in the same chunk. Another issue affecting the recall ability of a chunk is its location, not just the chess-specific relations between the pieces in it (Saariluoma 1984). It is easy to find strongly associated piece patterns in random positions, but they are very seldom correctly located. Finally, the speed of information intake must be taken into account. Ellis (1973) and Saariluoma (1984, 1985) have shown with very different procedures that skilled chess players are faster to extract information from chess positions than less skilled (Charness 1988). Chase and Simon (1973) have also noted that the more skilled the subjects the faster they learn chess games.

All these properties of chunks must be built into any model attempting to explain chess players' recall of chess positions. The model must contain an initially almost empty long term memory with a large number of simple chunks and it must be able to incrementally learn larger chunks. The pieces in chunks must have a number of chess specific relations between them and they must also be located in precise positions on a chess board. The speed of learning must also increase. The importance of building this kind of model is in testing the logic of theories. It has been known since the original study by Simon and Gilmartin (1973) that chunking can be studied in this way. Their model, however, was not a learning program and therefore it was not suitable in explaining the early learning curve. It is thus necessary to build a model, which is able to simulate the dynamics of the learning process. In the simulation model described below only the aspects of learning and precise location of pieces when building chunks are addressed, other chess specific heuristics are not considered.

Structure of the Simulation Program

Two versions of computer simulation programs were built to model the chunk construction strategies of novice chess players in the experiments described above. The models were programmed in the object-oriented language Java. Their functionally separate cognitive components are implemented as different object classes, instances of which are created during run time. The main classes are *piece, chunk, long term memory (LTM), short term memory (STM)* and *subject* that controls learning and recalling. The chess board is coded as two dimensional array of strings which present piece type and color. As the size of the array was the same as a real chess board's, the location of every piece was presented explicitly. For the chunking chess pieces (location, color and type) were coded as integers, and memory chunks were lists of these integers. The class hierarchy, and class or object methods were not intended to model human cognitive architecture or algorithms. The system was only to predict the development of the learning curve of an unexperienced chess player due to accumulation of new chess chunks in memory when her/his recall of unfamiliar chess positions is tested regularly during the learning phase.

Learning and Recalling

In the beginning the simulation systems have in LTM 768 chunks, which present every possible one piece chunk that can be formed, i.e. every piece type (12) on each location (64) on the board. So it was assumed that the subjects can trivially recognize single isolated pieces wherever they are situated on the board. After the initial situation the systems form new chunks in LTM from every shown study position. The size of chunks stored in LTM increases due to the systems' experience; in the beginning chunks of size two pieces are built, but later on the systems memorize larger chunks as they notice that possibly all the two (three, four, five, etc.) piece chunks are already known to them. The amount of chunks learnt from one position and used in recalling one position is limited by the capacity of the short term memory. Overlapping chunks are not constructed from a single position.

Unlike Simon's and Gilmartin's (1973) EPAM-based learner, the systems can find a chunk in memory independent of the piece around which it is built, so the chunk is identified by the pieces in it and no duplicates of the chunks are stored. Simon's and Gilmartin's chunks were identified by the focal pieces around which the chunks were built. However, the simulation systems do not examine the chess positions as a whole but process them one chunk at a time, starting with a specific or a random piece on a board. In the test phase the systems first build a proper chunk of the pieces on a test position, and then look for a corresponding chunk in LTM. So they do not reconstruct positions on empty boards like in Gilmartin and Simon (1973), but try to cover the pieces on the board with corresponding chunks in LTM if they are found. If the chunk cannot be found, the systems try a one piece smaller or a totally new chunk, otherwise they add the chunk to STM and mark the corresponding pieces on the board as recalled. Finally the recall score is calculated as a percentage of pieces explained by chunks in STM of all the pieces in the position.

Pieces or chunks that are not seen in learning phase are never memorized or retrieved, so the models make no commission errors. Once they have learnt something they never forget it, nor retrieve any incomplete or wrong chunk from memory. The models did not learn any chunks from the test positions, either.

Chunking Heuristics

The first version of the simulation is a naive one. It uses a **random neighbourhood heuristic**. In the learning phase and in the recall phase it always processes the chess positions in random order. It starts building a chunk from a random piece (focal piece), and when expanding a chunk to its neighbours the system proceeds to a random direction. It should be noted that only the pieces in adjacent location can form a chunk. The pieces that do not have any immediate neighbours can only form a one piece chunk.

The other simulation model uses a **correlation heuristic** when constructing chunks. Its decision about which pieces belong to a chunk is based on the frequency of co-occurrence of those pieces. The system chooses the most commonly seen piece as a focal piece around which it tries to form a chunk. Next the system adds to the chunks the most common neighbour of this focal piece, and then expands the chunk to the most common neighbour of this piece. However, in the learning phase the system starts examining the board and building chunks from random pieces. In this way it is guaranteed that the diversity of learnt chunks is high; not merely the most frequent pieces or chunks around them are exploited. Additionally, the multiplicity of chunks was thought to be of some use in recalling random positions.

The correlation model keeps record of the occurrences of single pieces and two piece combinations in a matrix like table. Note, that the system calculates frequencies of those pieces only that it memorizes in learning phase.

Neither of the models take into account the possibility of building very oddly shaped chunks. Despite of the chunking heuristics their structure and functioning is identical.

Simulation Results

The conditions in which the simulation models were tested were similar to those of the subjects MQ and NT. The models were taught 500 chess positions and the recall of unfamiliar game and random positions was requested within the same intervals as MQ's, in the beginning, after 30, 60, 175, 220, 270, 350 and 500 studied positions. Every test session consisted of ten real game and ten random positions. The random positions were permutations of the real chess positions used in the tests. They included just the same number and type of pieces, only in different locations.

For curiosity, test runs were run with short term memory sizes 4, 7, 10 and 12 chunks, because it was not very clear in the beginning whether it was just the quality of the chunks, not the number of them used in recalling that could improve the performance most. The short term memory size of 4 produced quantitatively the most similar results to the novice human subjects. When the STM size was over 7 chunks, the performance reminded that of experts'. The learning curves with STM size 4 for real game positions and random positions, and for both models are presented in figure 2. The curve is plotted such that for every test session the average recall percents of game and random positions are calculated, and then the whole test sequence is averaged over 20 independent runs. Similarly, the learning curves for all the short term memory sizes and both chunking heuristic in real game conditions are presented in the figure 3.

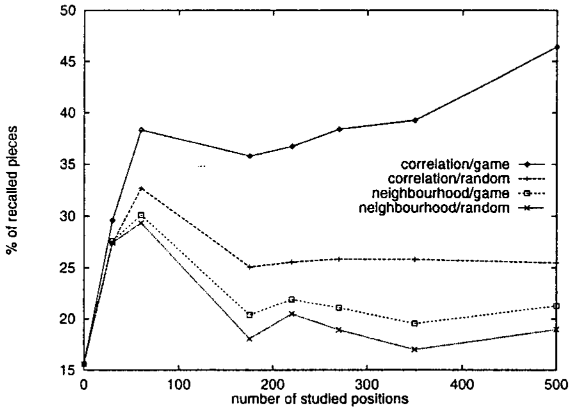

Figure 2: Percentage of recalled pieces when the simulation systems were tested 8 times during learning. Short term memory size was 4 chunks.

When evaluating the simulation versions it is very clear that the correlation model can exploit the regular patterns seen on chess board much better than the random neighbourhood model. Hence it is able to memorize the most useful piece combinations which help it to recall more pieces in the test situations. The other model stores too much redundant information. With the same amount of stored chunks it could recall much less of the game positions. The both simulations performed worse in the random test condition than in real game test condition, but the correlation model performed significantly worse than in the real game condition. With the neighbourhood model the difference was not so big. Still in the random con-

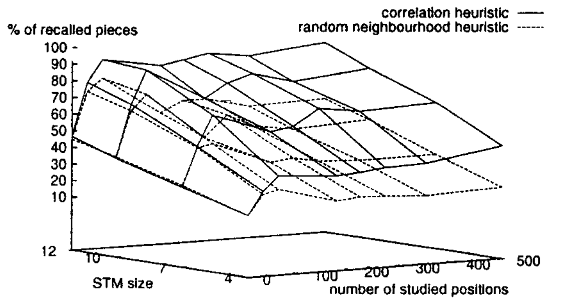

Figure 3: Learning curves for the short term memory sizes 4, 7, 10 and 12.

dition the correlation version reached better results than the neighbourhood version in the real game condition.

The Effect of the Amount of Learnt Chunks vs. Short Term Memory Size

The hypothesis was that the sharp increase at the beginning and the modest increase later on in the learning curve is caused by early accumulation of relevant chunks in LTM. Although the chunk amount goes up at a quite constant rate throughout the learning time the recall score does not seem to reflect it very well. It seems like after some turning point remarkably more chunks would be needed to enhance the performance. Note that the program using a neighbourhood heuristics could produced no more than about 700 chunks while the other one discovered a little over 2000 chunks. The accumulation of LTM chunks for different STM sizes and the two chunking heuristics is plotted in figure 4.

The size of the simulated short term memory played more drastic role in the performance than learning the relevant chunks. With the minor size (4 chunks) it was impossible to reach the results that were quite easy to obtain with STM sizes 10 or 12, as can be seen in figure 3. Otherwise a huge amount more learning would be demanded, say 10000 or 50000 LTM chunks (which may be normal for expert chess players). Our simulation did learn only about 2000 chunks at its best, and the amount of formed chunks did not, somewhat surprisingly, vary with STM size, when only 500 positions were studied. However, the amount kept on increasing linearly when the number of studied positions was doubled to 1000 (the results of these runs are not reported here, because the recall score did not improve at all). If the formation of overlapping chunks had been allowed, the amount of stored chunks may have been remarkably larger, but the recall scores somewhat smaller, because the same pieces might have been included in several chunks when recalling a single position.

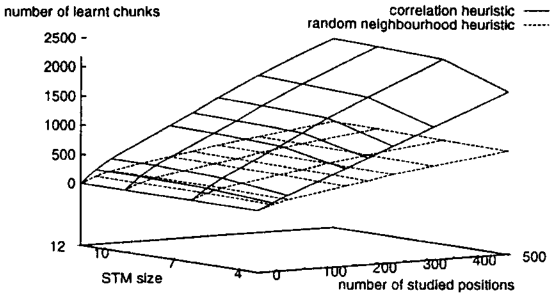

Figure 4: Accumulation rate of long term memory chunks during learning for both heuristics, run over STM sizes 4, 7, 10 and 12.

The Effect of the Chunking Method vs. the Size of Chunks

It was noticed that not merely the size of the chunks was important for the performance but the quality of them, although it was hypothesized that the learning is due to the gradual accumulation of bigger and bigger chunks. In practice the system could not exploit much larger chunks than five pieces on average, because it could learn only a fraction of the combinatorial alternatives, as the amount of them grows exponentially with the size. For this reason the longer chunks were harder to match to the game positions as they were seen so rarely during the learning phase. In the figure 5 the average sizes of the chunks used by the model using correlation heuristic in recalling real game positions are presented. The curve is plotted as an average of the largest chunks used to recall the ten test positions in every test session.

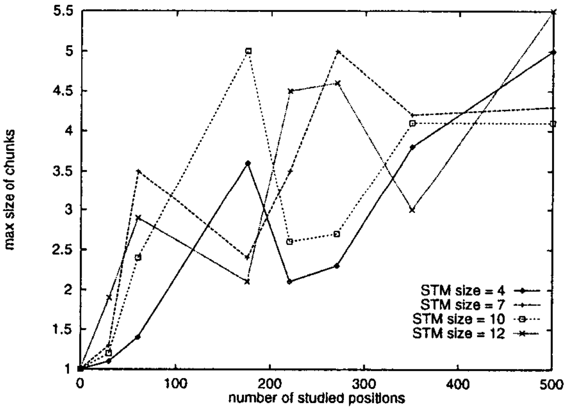

Figure 5: The average sizes of the largest chunks used in recalling real game positions by correlation model, plotted for all STM sizes.

It was also noticed that the overall method used in building chunks produced significant differences in performance. The method that incrementally builds larger chunks adding one adjacent pawn to an existing chunk improved the performance considerably compared to simple accumulation of chunks of different sizes. Hence it is advantageous to add one pawn to the earlier memorized chunk than to memorize two almost separate chunks that do not have many pieces in common. The latter method forms more variable chunks but it does not take into account the nature of real game positions. Although the test positions were not consecutive positions from a single game, especially the model using correlation heuristic was superior in exploiting regular patterns in positions it had seen in learning phase. The incremental method building chunks uses additionally the idea that it does not really matter that one pawn has changed place, it can still retrieve the chunk partially i.e. recall a one pawn smaller chunk that it has possibly built earlier. It may not have seen all the one piece smaller chunks previously, but in practice quite a few of them, anyway.

GENERAL DISCUSSION

These empirical results are very clear. A negatively accelerating learning curve was found. In this aspect chess is similar to many other symbolic and motor skills: The first steps are always the fastest in acquiring any skill. During this period one learns the most basic but also most common aspects of the domain. In chess this means the very familiar chunks such as castling or standard pawn chains. In random positions the absence of similar regularities makes it impossible for them to find equally common piece configurations and chunking is much less effective. Later in skill development the number of pieces in a chunk will increase and the number of combinatorial possibilities also increases exponentially. Consequently, it is very logical that the increase in the number of recalled pieces decreases respectively.

By a computer simulation we investigated various possibilities to interpret the empirical data. Of course, simulation has several weaknesses as a method. It is far from being unambiguous, because it is possible to construct several different types of models to investigate possible interpretations of data. Nevertheless, one should not forget that it is still better than mere intuition. The formal dimension of thinking is better controlled by using modeling than by relying on intuition. Therefore, a simulation, though having undeniably speculative sides, can be beneficial.

In our simulation, we were at first interested in the form of learning curve. Therefore, we let chunks grow incrementally, and indeed, our assumption was correct. There was a clear difference between game and random positions. This means that the incremental growth of chunks is a very good conjecture for the explanation of the negatively accelerating learning curve in chess. The exponential growth in the number of possible chunks as a consequence of the increase in the required chunks' length effectively explains the form of the learning curve.

It was interesting to notice that we got the best fit with the data when the size of working memory was kept at 4. On one hand this piece of evidence fits extremely well with the classic models of chess players' memory suggested by Chase and Simon (1973), for example, in which chunks are stored into short term working memory. However, the empirical research since Charness (1976) has clearly shown that experts do not store information into their short term working memory but in long term working memory (Ericsson and Kintsch 1995). This is apparently very problematic, but we must remember that in this experiment we investigated early learning. Therefore, there is nothing strange in that people would use their short term working memories to store chess specific information. The development of retrieval structures typical to masters takes a decade.

Finally, we were interested in the nature of associative connections between the chunk elements. The two simulation models suggest very evidently that frequency based correlative chunks provide much better a model for the data than random neighbourhood model. Indeed, this fact is also in harmony with the classic theory of associations, which assumes that frequent co-occurrence is sufficient explanation for many associations.

The ultimate point of simulation is the analysis of the interconnections between various phenomena and cognitive mechanisms. In that way simulation allows us to provide global theoretical concepts with more accurate contents than is possible when basically intuitive theoretical notions are used. This is an important point when the foundations of psychological argumentation is considered (see Saariluoma 1997). Here, the main problem is to find, how the learning curve, chunks growth, ST-WM capacity are interrelated and what is the significance of these finding in global psychological terms.

The model suggest that chess skill is essentially based on associative piece configurations and the basic learning mechanism is a gradual construction of them. The problem in improving memory recall is to resolve the combinatorial problem of getting sufficient number of chunks to get full coverage of standard real game piece configurations. The learning curve shape is thus simply a consequence of required number of chunks on each level of length. More chunks of length five need to be stored than chunks of length three. Consequently, model suggests that the shape of early learning curve is a consequence of combinatorial properties of the materials and limited capacity of the system.

In global terms, one can argue that chunking is one form of knowledge construction. As it is well known, the major contemporary global learning theory is called constructivism. It is predominant way of think-

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

ing as well in clinical as in social and educational psychology (Resnick 1987). The crucial theoretical problem in this way of thinking is the notion of construction itself. What does it mean, in concrete terms, that people construct their knowledge bases. The simulation of early learning provides one alternative. It is frequency based construction of associative and prelinguistic patterns.

A problem in this context is the precise role of learning results and chunking mechanism. de Groot and Gobet (1996, p.117) criticise Chase's and Simon's (1973) chunking explanations relying on Ericsson's and Harris's (1990) experiment in which they showed how a novice, by using mnemonic techniques, can improve his/her performance with no improvement in chess skills. The point is that chunks only do not suffice in chess, but knowledge about moves is also required. The authors are naturally correct in their thinking.

Nevertheless, one can argue that the memory mechanism of chunking is not bound to static piece patterns, but moves are sequences of spatio-temporal chunks. Thus chunking in position recall tasks utilizes the same underlying mechanisms that all learning of chess knowledge. Blindfold game recall strongly speaks for this interpretation (Saariluoma 1989). The problem is that one must learn all relevant types of chunks, i.e. piece configurations and moves, to improve one's chess skills. The concept is relevance (de Groot and Gobet 1996, Saariluoma 1995). If people do not learn relevant spatio-temporal chunks their skill construction is biased. Indeed, much of our conceptual knowledge is in these tacit patterns and therefore it is so important to understand these knowledge construction mechanism also in early learning (Saariluoma 1995, 1997).

ACKNOWLEDGMENT

This work was supported by Foundation of the 350th Anniversary of University of Helsinki grant to Tei Laine.

REFERENCES

Charness, N. (1976). *Memory for chess positions: Resistance to interference.* Journal of Experimental Psychology: Human Learning and Memory, 2, pp. 641-653.

Charness, N. (1988). *Expertise in chess, music, and physic: A cognitive perspective.* In L. Obler and D. Fein (Eds.), The exceptional brain. New York: Guilford press.

Chase, W. G. and Simon, H. A. (1973). *The mind's eye in chess.* In W. G. Chase (Ed.), Visual information processing. Academic Press: New York.

Ellis, S. H. (1973) *Structure and experience in the maching and reproduction of chess patterns.* Unpublished doctoral dissertation, Carnegie-Mellon University, Pittsburgh. Citated from Charness (1988).

Ericsson, K.A., Harris, M.S. (1989), cited in de Groot and Gobet (1996).

Ericsson, K.A. and Kintsch W. (1995). *Long term working-memory.* Psychological Review 102, pp. 211-245.

Fisk, A. W. and Lloyd, S. J. (1988). *The role of stimulus-to rule consistency in learning rapid application of spartial rules.* Human Factors, 30, pp. 35-49.

Frey, P. W. and Adesman, P. (1976). *Recall memory for visually presented chess positions.* Memory and Cognition 4, pp. 541-547.

de Groot, A. D. (1965). Thought and choise in chess. The Hague: Mouton.

de Groot, A. D. (1966). *Perception and memory versus thought: Some old ideas and recent findings.* In B. Kleinmuntz (Ed.), Problem solving: Research, methods and theory. New York: Wiley.

de Groot, A. D., Gobet, F. (1996), Perception vs. Memory in Chess. Assen: van Gorcum.

Lane, D. M. and Robertson, L. (1979). *The generality of the levels of processing hypothesis: An application to memory for chess positions.* Memory and Cognition, 7, pp. 253-256.

Lories, G. (1987). *Recall of random and non random chess positions in strong and weak chess players.* Psychologica Belgica, 27, pp. 153-159.

Newell, A. and Rosenbloom, P. (1981). *Mechanisms of skill acquisition and law of practise.* In J. R. Anderson (Ed.), Cognitive skills and their acquisition. Hillsdale, N. J.: Erlbaum.

Newell, A. and Simon, H. A. (1972). Human problem solving. Engelwood Cliffs, N. J.: Prentice-Hall.

Resnick, L.B. (1987), *Learning in school and out.* Educational Researcher, December.

Rosenbloom, P. and Newell, A. (1987). *Learning by chunking: A production system model of practise.* In D. Klahr, P. Langley and R. Neches (Eds.), Production system models of learning and development. Cambridge, Mass.: MIT Press.

Saariluoma, P. (1984). *Coding problem spaces in chess.* Commentationes scientiarum socialium, vol 23. Turku: Societas scientiarum fennica.

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

Saariluoma, P. (1985). *Chess players' search for the task relevant cues.* Memory and Cognition 13, pp. 385-391.

Saariluoma, P. (1989), *Chess players' recall of auditorily presented chess positions.* European Journal of Cognitive Psychology 1, pp. 309-320.

Saariluoma, P. (1995), Chess Players' Thinking: A Cognitive Psychological Approach. Routledge, London.

Saariluoma, P. (1997), Foundational Analysis. London: Routledge.

Simon, H. A. (1976). *The information storage system called human memory.* In M. Rosenzweig and E. Bennett (Eds.), Neural mechanisms in learning and memory. Cambridge, Mass.: MIT-Press.

Simon, H. and Gilmartin, K. (1973). *A Simulation of memory for chess positions.* Cognitive Psychology, 5, pp. 29-46.

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

Problem Solving with Incomplete Information: Experimental Study and Computer Simulation

Nathalie Chaignaud
LIPN - CNRS UPRES-A 7030
Université Paris XIII
Avenue J.B. Clément
93430 Villetaneuse - FRANCE
nat@lipn.univ-paris13.fr

Anh Nguyen-Xuan
Laboratoire de Psychologie Cognitive
Université Paris VIII
2, rue de la liberté
93526 Saint Denis Cedex - FRANCE
anguyen@ext.jussieu.fr

ABSTRACT

The aim of this study was to understand some particular human methods of problem solving in everyday situations. In this aim, we designed an experiment to obtain individual protocols. A cognitive model was based on the notions of *phases* and *states of mind* that evolved during the problem-solving process. The proposed model was then implemented in IGGY, a system which uses a blackboard architecture, and the validity of the model was tested by a Turing-like test and by a statistical analysis.

Keywords

Cognitive modelling, problem solving, incomplete information, model validation, blackboard system.

INTRODUCTION

In everyday life, people frequently encounter incompletely described situations where common sense reasoning and planning are essential. In most of these situations, the complexity of the reasoning process comes both from the fact that the state space in which constraints have to be satisfied is so large that no combinatoric approach can be used, and from the fact that some information is missing and hence must be collected. A simplistic example of such a situation is when someone wants to organise a party with friends. To this end, several constraints must be satisfied and the problem cannot be solved instantaneously because information is missing.

The way the solution is built up depends in particular on how strictly people pay attention to the constraints and how well they gather and use information. In this kind of problem, people may choose to use sophisticated reasoning that resembles planning, by optimally articulating the pieces of information already known, inferring the best way to gather the missing information, and anticipating the different possible outcomes. On the opposite, they may decide to avoid paying an important cognitive cost and adopt a simple behaviour, driven more by reaction than by planning. However, although Agre and Chapman (1987) stated that this reactive behaviour was cognitively plausible, there is, to our knowledge, no experimental evidence of such activity for human subjects.

In this interdisciplinary study, at the intersection of Cognitive Psychology and Artificial Intelligence, our purpose is to understand and simulate the way human beings elaborate plausible conclusions in imperfectly described everyday situations. To this end we have chosen to carry out a psychological experiment dealing with an *ill-structured problem* (Simon, 1973; Voss & Post, 1988; Goel, 1992). In this class of problems, people have to reason on incomplete or uncertain knowledge. Design problems (Guindon, 1990; Visser, 1990; Ball et al., 1997) form a particular subclass of this class.

In our work, a *bottom-up* approach has been adopted: an experiment was conducted in which subjects were to solve individually the so-called "hi-fi system problem". The set of experimental protocols obtained were analysed to extract the different behaviours, and from this analysis a computational model was built and implemented in order to have a better understanding of the human problem-solving process. Finally the output of this simulation was compared with the human protocols in order to validate the proposed model.

THE PROBLEM-SOLVING SITUATION

The task was designed with the following characteristics:

- the set of constraints could be satisfied in a large state space, and a pure combinatoric solution could not be considered;
- information initially available had to be incomplete in order to compel the subjects to reason in an uncertain environment;
- it had to be of sufficient complexity so as to obtain a large range of behaviours; however it had to be simple enough to be manageable.

The problem consisted in configuring a hi-fi system. A complete system comprised five different items: an amplifier, a tuner, a record player, a tape recorder and a compact disc reader. The subject could choose between three models of amplifiers and between four models for each of the other items. The amplifier had a special status insofar as the other items had to be compatible with it[1]. The price of the items, the maximum amount allowed and the compatibility between the amplifiers and the components[2] were given to the subjects at the beginning of the experiment. However, the subjects did not know which items were available at the beginning. This information had to be acquired by making a phone call for each chosen item and the subject knew that s/he would be told the number of calls allowed in due time. Thus, there were four types of constraints: the total price of the system, the compatibility between the amplifier and the components, the availability of the chosen items and the number of phone calls.

In order to record all the actions performed by the subjects, the task was simulated via a computer program.

[1] In the remainder of this article, the amplifier is thus differentiated from the other items called "components".

[2] Some components were compatible with more than one amplifier.

The user interface was designed to serve as an external memory store and a calculator of the total amount spent on the chosen items. In order to instigate a large variety of behaviours, three versions of the problem were built, which differed by price and compatibility table. Forty seven female and male students took part individually in the experiment. Each subject was asked to solve the problems by thinking aloud. The subjects' verbalisation was tape recorded and all their actions were automatically recorded by the simulation program. Therefore, a *protocol* was a list of all the subject's actions (and results of actions) and verbalisations during a problem-solving process.

Fifteen of the 141 individual protocols were eliminated from data corresponding to certain subjects that did not understand the instructions. Therefore the raw data comprised 126 protocols.

BUILDING A FRAMEWORK TO MODEL THE PROTOCOLS

The model had to be realistic, complete and simple. This needed a lot of comings and goings between the analysis of the experimental data and the building of the model. Therefore, our cognitive model was built by successive approximations.

An ideal strategy

An optimal way for handling the problem situation consists in selecting three configurations, based on three different amplifiers, and in choosing components that are as multi-compatible as possible. By doing so, one can make sure that the constraints of price and compatibility are satisfied. Information gathering (i.e. by phoning) will be undertaken only when the configurations are completed. This *ideal strategy* is based on parallel planning and can be characterised as being "opportunistic" when item choice and information gathering exploit the idea of multi-compatibility. Such a strategy is similar to a breadth-first search observed in expert designers (Ball et al., 1997), because it takes into account the three possible alternatives at the same time and leads to a solution with a minimum risk of backtracking.

A rough characterisation of the protocols

The subjects could solve the problem at different levels of reasoning, from the most sophisticated to the simplest mode. We characterised a mode of reasoning as being sophisticated when (i) the subject built in parallel three configurations based on the three amplifiers, then explored in depth the configuration that appeared the most promising at a given time; (ii) the subject explored at the same time several solutions with a single amplifier; (iii) the subject used a strategy similar to the *focusing* one (Bruner et al, 1956) by phoning in order to reduce the set of possible solutions quickly. This mode of reasoning resembled the ideal strategy presented above. The reasoning was characterised as being shallow when the subject tried to build up only one configuration at a time. This mode of reasoning is similar to the depth-first search approach observed in novice designers. It bore two characteristics: (i) the subject abandoned the current solution only when forced to do so (i.e. one or more constraints were violated); (ii) the subject phoned for a component of a given category then shifted to another category as soon as s/he got a positive answer.

To our surprise, after a first superficial analysis of the protocols, we did not find much sophisticated reasoning. Only 13 of the 126 protocols can be characterised as adopting a sophisticated mode of reasoning. In the remaining 113, subjects focused on building up one configuration at a time.

Moreover, the subjects did not respect simultaneously all the constraints of the problem.

Despite the fact that the observed behaviours were simpler than expected, there did not exist two identical protocols. The question that arose was then how different were they? To answer this question we needed to define a framework for analysing the protocols more precisely.

This analysis takes into account the protocols that adopt a shallow mode of reasoning.

From the 113 protocols, three sets of protocols were drawn at random. The first set comprised 30 protocols which have been carefully analysed to determine the main ingredients of the cognitive model and to build up a precise method of analysing the protocols by hand. The second set of 43 protocols was randomly drawn from the remaining protocols to validate the completeness of the hand analysis method. The third comprising the 40 remaining protocols was used to validate the implemented model: they were not used for the setting up of the parameters of the implementation.

The ingredients of the model

Our model was based on the notions of phases, states of mind, strategies and tactics.

The Notion of Phases

The configuration building process rarely developed smoothly and some "obstacles" arose which had to be overcome. Two kinds of obstacles were distinguished: either they were not really bad and the situation needed only a few corrections or they were more serious and constituted deadlocks which needed to be removed. From this point, we differentiated between the situations considered as being *normal* and those considered as *abnormal*, with two degrees of *abnormality*.

In every abnormal situation, the configuration building process was interrupted and the subjects undertook either a correction task or a deadlock-solving task. After the obstacle had been solved, they either returned to their previous task or tested their configuration if they thought they had found a solution. Thus, each protocol could be divided into *phases* characterised by the current task. These phases were *configuration building*, *correction*, *deadlock solving* and *test*. Figure 1 represents all the possible relations between the phases.

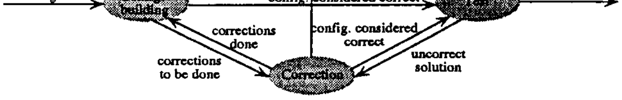

Figure 1: the possible relations between phases

The Criteria Taken into Account by the Subjects: the State of Mind

The decision to perform any particular action depended on the attention paid to the different constraints. Thus, we defined the notion of *criteria* taken into account by the subjects. They were related to the four constraints of the problem, and they were given the same names:

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

- *compatibility criterion*: the subjects focused on the compatibility between the different items,
- *availability criterion*: the subjects tried to check as soon as possible the availability of items,
- *price criterion*: the subjects took into account of the price of the items,
- *phone calls criterion*: the subjects were careful how many phone calls they made.

The criteria were different from the constraints: satisfying a constraint meant making sure that it was not violated, whereas complying with a criterion only meant that the subjects had this constraint in mind while taking decisions, in order to reduce the chances of violating it.

The subjects did not necessarily take into account all criteria at the same time but only a subset of them that varied as new information was acquired. This subset of criteria was called *state of mind.* It evolved according to the problem-solving situation and its changes triggered modifications in the subject's behaviour.

The Possible Strategies

In a configuration building phase, the subjects could perform two different kinds of actions: choose the items that will form a configuration and gather information about the availability of items. We distinguished between *item choice strategies*, and *information acquisition strategies*.

There were two possible item choice strategies:

- *amplifier centred strategy* (Strategy 1): choose the components by focusing on only one amplifier,
- *component centred strategy* (Strategy 2): choose the components without having in mind a predetermined amplifier, so that the determination of a single amplifier was delayed as long as possible.

Three possible information acquisition strategies, probably related to the user interface used in the experimentation, emerged from the protocols:

- *select then phone strategy* (Strategy 4): choose several items and then phone for each of them,
- *phone then select strategy* (Strategy 5): give a sequence of phone calls for a series of items and then build up a solution by selecting only available items,
- *phone and select simultaneously strategy* (Strategy 6): select each item and then phone immediately for it (if available, the item was kept, else it was discarded, and the subject selected another item).

Moreover, the subjects could forget that it was necessary to know the availability of all the elements of a chosen configuration. In this case a *null strategy* (Strategy 7) was attributed.

In a deadlock-solving phase, the subjects had to change the flawed configuration by deciding to focus the deadlock-solving process on either an amplifier or a component. All the strategies, except Strategy 4 and Strategy 7, could be applied.

Finally, since the correction phase concerned the items that violated the constraints, no strategy on item choice was necessary. This is the reason why a *null strategy* on item choice (Strategy 3) was attributed to any error correction phase.

Table 1 summarises all the possible strategies in the different phases.

Item choice strategies	CB	C	DS
1. amplifier centred strategy	X		X
2. components centred strategy	X		X
3. null strategy		X	
Information acquisition strategies	**CB**	**C**	**DS**
4. select then phone strategy	X	X	
5. phone then select strategy	X	X	X
6. phone and select simultaneously strategy	X	X	X
7. null strategy	X	X	

Table 1: strategies for each phase
(CB: configuration-building phase, C: correction phase, DS: deadlock-solving phase)

Instantiating the Items Choice Strategies: the Tactics

In the configuration building and deadlock-solving phases, the same strategy on item choice could be instantiated through different atomic actions. In order to differentiate between these different choices, we introduced the notion of *tactics*.

	CB		DS		C
Tactics on amplifier choice	**St. 1**	**St. 2**	**St. 1**	**St. 2**	**St. 3**
1. cheapest amplifier	X		X		X
2. medium-priced amplifier	X		X		X
3. amplifier compatible with the most components	X		X		X
4. amplifier compatible with the fewest components	X				X
5. amplifier most compatible with the configuration	X		X		X
6. amplifier compatible with available components	X		X		X
7. amplifier compatible with the cheapest components	X		X		X
Tactics on components choice	**St. 1**	**St. 2**	**St. 1**	**St. 2**	**St. 3**
8. cheapest comp. compatible with a given ampl.	X				X
9. medium-priced components compatible with a given ampl.	X				X
10. available comp. compatible with a given ampl.	X				X
11. cheapest comp. compatible with at least 2 amplifiers		X			X
12. components compatible with the most amplifiers		X			X
Tactics on key-component choice	**St. 1**	**St. 2**	**St. 1**	**St. 2**	**St. 3**
13. cheapest key-component		X		X	
14. key-component of blocking category				X	
15. available key-component				X	

Table 2: tactics in terms of phases and strategies

From an informal analysis of the 30 protocols, 15 different tactics were identified, which depended on the current phase, on the state of mind and on the item choice strategy. We distinguished between tactics for choosing an amplifier and tactics for choosing the components. In the latter case, there was an additional distinction between situations where a set of components had to be chosen and situations where only one component, called *key-component*, had to be chosen in order to start or restart a

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

configuration. Table 2 presents the fifteen tactics and their domain applicability.

METHOD TO ANALYSE THE PROTOCOLS

The hand analysis was performed in parallel by two "judges"[3] and the infrequent disagreements between them were easily solved after a discussion with a third judge. The method to analyse the protocols was applied to 73 protocols: the set of 30 that were used to build up the framework of the cognitive model, and the set of 43 that were reserved to verify the completeness of the hand analysis method.

Decomposing the protocols into phases

Identification of the test phase was straightforward: it boiled down to the "test" action.

A deadlock-solving phase was usually a short sequence of actions that eventually led to a change of amplifier. It began when the subjects considered that they would not be able to find a solution with the current amplifier. After possibly checking the availability of some items, the subjects chose either a new amplifier or a component compatible with another amplifier. This choice ended the deadlock-solving phase.

A correction phase, by contrast, was a phase where the subjects checked the availability of items and/or replaced components that violated the constraints. It could be triggered by a negative test or simply by the subjects' noticing one or more errors in the solution.

A configuration building phase was simply defined as a phase that was none of the three phases defined above.

Recognising the strategies and the tactics

Once the phases had been identified, strategies and tactics were rather easy to detect. But the identification of strategies and tactics could not be conducted separately. Most actions, when taken out of context, were compatible with more than one strategy and more than one tactic according to the possibilities given in tables 1 and 2. It was often necessary to take into account a sequence of actions in order to narrow the range of possibilities. As mentioned earlier, the verbal utterances were good clues to help choose among the possible hypotheses. Thus the approach we used to identify strategies and tactics was a hypothesis-and-test approach.

Identifying the state of mind

The last ingredient to be identified in the protocols was the state of mind. To this end, we assumed that any criterion that appeared in the state of mind had a visible effect in the protocol.

The information acquisition strategies depended on the availability criterion . Strategy 5 and Strategy 6 favoured an early discovery of the availability and therefore required the presence of the availability criterion, whereas Strategy 4 and Strategy 7 were inconsistent with it.

Each tactic corresponded to one or two criteria, and some tactics excluded a criterion. The compatibility criterion was also attributed when the subject explicitly referred to it when choosing items.

The subjects could change their states of mind, strategies or tactics during a configuration building phase. This meant that a configuration building phase could be divided into several episodes. An episode was defined as a sequence of actions characterised by the same set of phase, state of mind, strategies and tactics. The end product of the analysis of a protocol is a *skeleton* which partitions the protocols into successive episodes.

GENERATING ARTIFICIAL PROTOCOLS

Our aim was to build a computational model to simulate human reasoning in this particular problem-solving task. Protocol analysis made it possible to identify, for each protocol, the successive phases, strategies, tactics, states of mind and their changes during the problem-solving process. However, the model did not "explain", why different subjects adopted different states of mind, strategies and tactics and why some of them made more careless mistakes than others. In order to introduce this inter-individual variability, individual characteristics had to be taken into consideration. This led us to introduce the notions of *observation* and of *personality.*

Linking the episodes: observations as triggers of the episode changes

During problem solving, any change of ingredient corresponds to a new episode, which depends on the subjects' interpretation of the current situation. From a generative point of view, our aim was to simulate not only behaviours inside episodes but also the inferences the subjects made from the current situation in order to go ahead. For this we used the notion of *observation,* which corresponds to the explanations about the problem-solving process that would be present in the subjects' verbalisations if these were complete. The subjects' verbal utterances are thus considered as a sample of their observations and they play an important role to explain the changes of episode.

Observations may concern the current configuration as a whole (e.g. "the configuration is too expensive") or a particular element (e.g. "tuner 1 is available"). They can have an impact on the phase (e.g. deadlock, correction to be done or configuration to be tested), on the state of mind (e.g. number of phone calls already made too high: take into account the phone calls criterion), on strategies (e.g. an element compatible with several amplifiers: Strategy 2) and/or on the current tactics (e.g. expensive configuration: "cheapest components" tactic).

From the 73 protocols that have been hand analysed, all the verbal utterances were picked out except for the meta-cognitive statements. The list of useful utterances can be grouped into 22 observations. In our model, the observations are represented by predicates with or without arguments. They are not described here due to space limitation.

The personality of the subjects

The second notion we needed in order to simulate the diversity of the observed behaviours was *personality.*

For each of the 73 protocols that have been analysed, we determined the personality of the subject by 5 orthogonal features:

- *careful,* for the frequency of careless mistakes made by the subject,
- *thrifty,* for the importance attached by the subject to the price of the configuration,
- *opportunistic,* for the subject's ability to use information flexibly,

[3] We thank Jean-Marc Meunier for having done a very efficient job as one of these two judges.

- *systematic*, for the subject's choice of elements that followed more or less strictly the order in which the categories were presented in the price and compatibility table,
- *good appraiser*, for the subject's aptitude to estimate a situation correctly.

All of them can take the values *poorly, fairly* and *very*.

In the process of generating artificial protocols, these parameters are given as data of the problem-solving process and observations and choices are made essentially according to them.

IMPLEMENTATION OF THE COGNITIVE MODEL: THE IGGY SYSTEM

From a psychological point of view, the implementation of our cognitive model aims to validate the model and, from an AI point of view, to show the feasibility of such a computational model. IGGY, a system written in Common Lisp, implements the model: it is a protocol generator that takes as input a personality and gives as output a protocol corresponding to this personality.

We need an architecture allowing the specialisation of knowledge and the sequentiality of actions. The *blackboard architecture* (Engelmore & Morgan, 1988, Hayes-Roth, 1993) with a *hierarchical control* is well suited to our needs.

IGGY's components

IGGY is a hybrid system which contains three elements: a blackboard, an *executor* and an *engine* (see Figure 2). The engine co-ordinates the other two elements in a "perception-decision-action" loop in disguise, where the perception and decision tasks are accomplished by the blackboard and the actions are performed by the executor, which generates protocols.

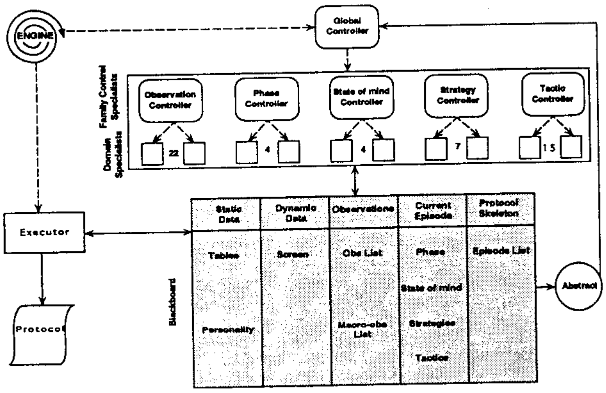

Figure 2: the system IGGY

The Blackboard System

It includes a blackboard, domain specialists grouped in five families, family controllers and a global controller.

The blackboard

It includes five thematic panels: the static parameters of the problem (compatibilities, the price of the items and the personality of the simulated subject), the dynamic data of the problem (the availability of the items and the configuration), the current observation list, the current episode and the history of the different episodes of the problem solving (the skeleton).

An *abstract*, updated at each change of the blackboard, informs the system on the nature of the new information (phase, state of mind, strategies, tactics, observations or action).

The domain specialists

There are fifty two domain specialists grouped into five families representing the ingredients. Each specialist corresponds to a possible choice in its family. Thus, there are twenty two specialists for the observations, four for the phases, four for the states of mind, seven for the strategies and fifteen for the tactics.

They are represented by "condition/action" rules: their condition concerns the state of the blackboard and their action consists in writing a new instantiation of an ingredient on the "current episode" panel, except for the "observation" family that writes on its own panel.

The family controllers

Each control specialist, called *family controller*, concerns one family and knows the list of domain specialists that it supervises. At the beginning of the problem-solving process, the observation controller is triggered to initialise the observation list. Observations are generated according to the personality. From this list of observations, the first episode is calculated by the phases controller, the states of mind controller, the strategies controller and the tactics controller. Then the executor performs just one action according to the episode. After each executed action, the observations controller is triggered and either new observations are written in the observations list, or no observation is made. In the first case, a new episode is calculated. In the second case, the executor continues with its job, and so on.

The family controllers are specified by condition/action rules. The condition concerns the state of the abstract and is defined so that family controllers are triggered cyclically in the following order {observations, phases, states of mind, strategies and tactics}. The action is threefold: send a call for proposals to the domain specialists, choose one of the candidates and trigger it. If there is no candidate then the next family controller is triggered.

The global controller

The global controller supervises all the family controllers. It is reduced to the action part: as a control specialist, it send a call for proposals to the family controllers, then chooses one candidate and triggers it. When a new observation is made that raises a conflict between family controllers, the global controller chooses in priority the "phase" family controller, but if no domain specialist proposes a change, then comes the turn of the "state of mind" family controller, and so on.

The Executor

This module generates the sequence of actions corresponding to the current episode chosen by the blackboard system. It executes only one action at a time and gives the control back to the engine.

VALIDATION OF THE MODEL

Validation consists in comparing a set of real protocols with a set of simulated ones. The first (real) set is the 40 protocols that had been put aside to validate the implemented model. The second (simulated) set of protocols has been provided by IGGY: 73 protocols have been generated, each having the personality of one of the 73 analysed protocols. Then, 40 of the simulated

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

protocols have been randomly selected to constitute the artificial sample.

Turing-like test

The first validation method was based on a Turing-like test (Turing, 1950).

We have adapted this test in the following way: from the two sets of protocols described above, we randomly drew two samples of 15 protocols. The 30 protocols were given to the psychologist[4] who had already analysed the 73 real protocols. We asked him first to hand analyse them (using the same analysis framework to derive 30 skeletons), and second to classify them according to their origin (human or artificial). The results were very good since he misclassified half of the protocols: 8 artificial and 7 real protocols were classified wrongly by the psychologist.

Statistical comparisons

The comparisons were based on a set of observables obtained from the protocols of the two samples (40 protocols for each sample). The significant level we adopted was p=.05.

We counted separately the number of episodes for the four different phases: configuration building, correction, deadlock solving and test.

Table 3 shows the data concerning the configuration building episodes: data in the cells represent the number of protocols in which there were one, two, three, four or five configuration building episodes. By combining the "4 episodes" with the "5 episodes" cells, we obtained $\chi^2(3)=27.75$ (p<.05).

Prot.	1 epis.	2 epis.	3 epis.	4 epis.	5 epis.
IGGY	1	6	15	13	5
Real	4	25	9	2	0

Table 3: Number of protocols by number of configuration building episodes.

Table 4 shows data concerning the test phase: in the cells are the numbers of protocols in which there were one, two, three, or four test episodes. In order to use the χ^2 test, we considered two categories, "one episode" and "two-or-more episodes": $\chi^2(1)=12.29$ (p<.05), the difference was significant.

Prot.	1 epis.	2 epis.	3 epis.	4 epis.
IGGY	36	3	1	0
Real	22	15	2	1

Table 4: Number of protocols by number of test episodes.

Tables 5 and 6 show data concerning, respectively, the deadlock solving and the correction phases. As in tables 3 and 4, data in the cells represent the number of protocols. For these two types of data, the differences between the two groups are not significant: $\chi^2(2)=4.64$ (p=.10) and $\chi^2(2)=1.04$ (p=.59) respectively (the last 3 categories of Table 6 have been combined).

Prot.	0 epis.	1 epis.	2 epis.
IGGY	7	20	13
Real	10	25	5

Table 5: Number of protocols by number of deadlock-solving episodes.

Prot.	0 epis.	1 epis.	2 epis.	3 epis.	4 epis.
IGGY	22	11	6	0	1
Real	20	15	4	1	0

Table 6: Number of protocols by number of correction episodes.

We also counted the number of protocols in which a given strategy or tactic was observed at least once. For Strategy 1 and Tactic 15, no statistical test was needed; for the artificial and real protocols there were, respectively, 40 and 39 Strategy 1, and 1 and 0 Tactic 15. For the other data (six strategies and twelve tactics), Tactic 11 and Tactic 12 being combined, we used the χ^2 test when possible; else the Fisher exact probability test was used. Except for Strategy 6, the obtained p-values were either very small or very large, as shown in tables 7 and 8.

Note that data presented in tables 3 to 8 are obtained from the same two groups; they were a kind of "repeated measures". In this case, we should use the Bonferroni χ^2 statistic (Jensen et al., 1968), instead of the classical χ^2 statistic. It turned out that the Bonferroni statistic gave the same conclusions (except for Strategy 6, where p>.05) as the classical χ^2 statistic and Fisher's test on accepting and rejecting the null hypothesis.

From the twenty possible types of strategies and tactics, artificial and real protocols differed only for two of them (marked by "*"): Strategy 4 and Tactic 5.

Prot.	St. 2	St. 3	St. 4 *	St 5	St. 6	St. 7
IGGY	12	18	25	4	40	5
Real	10	20	9	7	35	4
p-value	.61	.65	.0003	.33	.03	.72

Table 7: Number of protocols (out of 40) in which the strategies were observed at least once

Prot.	Ta. 1	Ta. 2	Ta. 3	Ta. 5 *	Ta. 6	Ta. 7
IGGY	21	3	5	28	4	9
Real	26	9	6	9	5	4
p-value	.26	.06	.75	<.0001	.72	.13

Prot.	Ta. 8	Ta. 9	Ta. 10	Ta. 11/12	Ta. 13	Ta. 14
IGGY	39	5	8	7	8	7
Real	37	7	11	5	4	6
p-value	.37	.53	.43	.53	.21	.09

Table 8: Number of protocols (out of 40) in which the tactics were observed at least once

The last comparison between artificial and real protocols concerns the states of the mind in the first, penultimate and last configuration building episodes. It is clear that the diversity of the states of mind increases during the solving process. Table 9 gives the distribution of the states of mind.

[4] Jean-Marc Meunier

	1st epis.		Penult. epis.		Last epis.	
State of mind	IGGY	Real	IGGY	Real	IGGY	Real
Availability (A)	nil	nil	nil	nil	nil	nil
Compatibility (C)	nil	nil	nil	nil	nil	nil
Price (P)	10	9	1	5	nil	nil
Telephone (T)	nil	nil	1	0	1	0
AC	3	1	0	1	0	1
AP	24	25	24	28	11	19
AT	nil	nil	1	0	2	8
CP	1	0	2	1	1	0
CT	nil	nil	nil	nil	2	1
PT	nil	nil	2	0	15	5
ACP	2	5	6	1	4	5
ACT	nil	nil	nil	nil	2	1
CPT	nil	nil	2	0	2	0

Table 9: Number of states of mind at different episodes

In order to perform statistical comparisons, we combined the rows of Table 9 in the following ways:

- For the first episode, the χ^2 was calculated on a table with 2 columns and 3 rows: P; AP; CP+AC+ACP (the shared criterion was C);
- For the penultimate episode, the only combined table of observed data that allowed the use of a statistic test was a 2X2 table. Row 6 was left alone, and the remaining rows were combined;
- For the last episode, we chose the following combinations of rows: AP; CP+AC+ACP (the shared criterion was C); T+PT+AT+CT+CPT+ACT (the shared criterion was T).

For the 3 episodes, the differences between artificial and real protocols were not significant: $\chi^2(2)$=.073 (p=.96), $\chi^2(1)$=2.32 (p=.13), and $\chi^2(2)$=4.30 (p=.12), respectively.

Discussion

The results we obtained are generally good, since there were only four cases where the difference was significant. The results of statistical tests showed that, the p-value for a test was either very large or very small. It did not seem reasonable to appeal merely to the notion of *sampling error*. The differences concerned the number of episodes in the configuration building phase, the number of test phases, the number of protocols in which there was at least one application of Strategy 4 (select then phone strategy) or Tactic 5 (choice of the amplifier that was the most compatible with the configuration). Reasons must be found to explain some of these differences.

First, the IGGY'S protocols had more episodes in the configuration building phase than the real protocols. This difference can be attributed to an important difference between human subjects and IGGY as far as verbalisation is concerned. Unlike IGGY's reasoning, which is explicitly visible through the evolution of its internal state, the activity of the subjects is only known through their observed behaviour. Consequently there can be changes of episode within a configuration building phase that cannot be detected in the analysis of the real protocols, due to the lack of a proper verbalisation. On the contrary, in the case of IGGY, if all the conditions for an observation to be made are satisfied, then the observation is effectively made. In this respect, IGGY can be considered as a subject who verbalises all her/his actions. This explanation is coherent with the following finding. We counted the number of the verbal utterances in the real protocols that belonged to the 22 verbalisations we used in the model, and the number of observations in the artificial protocols. The mean number of verbal utterances were respectively 6.65 (S.D.=2.38) and 8.32 (S.D.=2.31). The p-value for the Student-t test was .002.

The second difference concerned the finding that human subjects were more likely to test the configuration when it was not yet a satisfying solution. This difference suggests that the simulated subjects were somewhat better appraisers than the human subjects. Thence, being better appraisers, the simulated subjects were more able to establish relationships between the amplifier to be chosen and a set of components already put in a column. Consequently, Tactic 5 was more often observed in the IGGY's protocols. However, IGGY and human subjects were equally efficient in detecting errors. These findings, together with the difference concerning the use of Strategy 4, suggested that the number of personality features we introduced as free parameters of IGGY was somewhat too small, and that these features were probably correlated.

CONCLUSION

We were interested in *complex problems* that belong to a semantically rich domain and which did not give from the outset all the information that was necessary to reach a solution.

The principal characteristics that differentiate our problem from the puzzle problems are: (i) the problem space is very large (more than one million nodes); (ii) the problem is more likely to be an *arrangement*[5] problem than a *transformation* one so that general heuristics such as means-ends analysis cannot be applied; (iii) the subjects must ask for information to find a solution. These characteristics are common to both our problem and design problems, although the latter are much more difficult, take more time, and are usually much more ill-defined.

In our experiment, we deliberately intently built the user interface in order to allow the subjects to use different ways of solving the problem, from a sophisticated reasoning mode similar to that of an expert in the domain of design, to a very reactive mode where the subject tries to build up one solution at a time, without considering the possible alternatives. It turned out that the majority of our subjects adopted an approach rather similar to that of the novice designers.

This result is coherent with studies about human reasoning, which demonstrated a general tendency to depart from *sophisticated* reasoning behaviour. For instance, research work on deductive reasoning and decision making has shown that people do not usually reason following an "apparently appropriate normative system" (Evans & Over, 1997, p.2) such as standard rules of logic or mathematical models. Several factors have been called upon to justify this departure from a normative behaviour, such as Simon's *bounded rationality* and *satisficing principles*, bias introduced by

[5] The goal is a state of the world that satisfies certain requirements. The anagram problem is typically an arrangement problem.

the way the subject builds up a representation of the problem situation, memory load, etc.

However, adult subjects cannot be dubbed as being incapable of sophisticated reasoning, because around 10% of the observed protocols showed an approach that was similar to the ideal strategy. For the remaining protocols, the fact that all of them reached a solution that met the goal requirements suggests that the subjects' behaviour can be considered as based on a *rationality of purpose* rather than a *rationality of process*. Evans and Over (1997) argued that the first kind of rationality is more generally and more spontaneously applied than the second one.

Finally, from the Artificial Intelligence point of view, although we did not find as much sophisticated reasoning and anticipative behaviour as we expected, this study brings out a number of interesting points. We have already shown (Chaignaud & Levy, 1996) that a parallel could be established between our cognitive model and recent trends in Artificial Intelligence such as knowledge compilation or constraint satisfaction.

We think that our model, and particularly the notions of phase, state of mind, strategy, tactic and personality, is general enough to be used in a whole class of problems that we have called *configuration problems*: variables have to be instantiated among a set of values that have to satisfy several constraints. Moreover the data of the problem is incompletely described. Possible examples include timetable problems in a school subjected to the constraint of availability of the classrooms, when the other obligations of the teachers are not known in advance; the travel agency problem where one has to schedule multimodal journeys to go from one point to another, and the availability of the seats is not known in advance. The dichotomy between normal and abnormal situations arises in most problems with incomplete information, and our system is able to manage two degrees of abnormality (simple errors and deadlock) and to react according to the situation.

By using the notions of personality and state of mind, our model accounts for observed individual differences that cannot be explained otherwise. Moreover, it captures the dynamic aspect of the problem-solving process.

REFERENCES

Agre, P. E. & Chapman, D. (1987). Pengi: an implementation of a theory of activity. *American Association for Artificial Intelligence*, Seattle, 268-272.

Ball, L. J., Evans, J. St. B., Dennis, I. & Ormerod, T. C. (1997). Problem-solving strategies and expertise in engineering design. *Thinking and Reasoning*, *3*, 247-270.

Bruner, J. S., Goodnouw, J. J. & Austin, A. (1956). *A study of thinking*. New York: Wiley.

Chaignaud, N. & Lévy, F. (1996). Common sense reasoning: experiments and implementation. *European Conference on Artificial Intelligence*, Budapest, 604-608.

Engelmore, R. & Morgan, T. (1988). *Blackboard systems*, Addison Wesley Publishing Company.

Evans, J. St. B. & Over, D. E. (1997). *Rationality and Reasoning*. Hove, UK: Psychology Press.

Goel, V. (1992). "Ill-structured representations" for ill-structured problems. *14th Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Earlbaum, 130-135.

Guindon, R. (1990). Designing the design process: exploiting opportunistic thoughts. *Human Computer Interaction*, *5*, 305-344.

Hayes-Roth, B. (1993). Intelligent control. *Artificial Intelligence*, *59*, 213-220.

Jensen, D. R., Beus, G. B. & Storm, G. (1968). Simultaneous statistical test on categorical data. *Journal of Experimental Education*, *36*, 46-56.

Simon, H. A. (1973). The structure of ill-structured problems. *Artificial Intelligence*, 4, 181-201.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, *59*, 433-460.

Visser, W. (1990). More or less following a plan during design: opportunistic deviation in specification. *International Journal of Man-Machine Studies*, *33*, 247-278.

Voss, J. F. & Post, T. A. (1988). On the solving of ill-structured problems. In M. T. H. Chi, R. Glasser & M. J. Farr (Eds.). *The nature of expertise*. Hillsdale, NJ: Earlbaum.