Cognitive Modelling for Decision Support in Naval Command & Control

A. Kalus[1]
School of Computer Science and Mathematics
University of Portsmouth
Portsmouth PO1 2EG
Hampshire
England

kalusa@sis.port.ac.uk

A. J. Ashford, C. A. Cook, C. Sheppard
Centre for Human Sciences
Defence Evaluation and Research Agency
Portsdown West
Fareham PO17 6AD
Hampshire
England

Abstract

Traditionally, a decision support system is built to help a broad range of users in their decision making. However, there are some domains such as air traffic control, the nuclear industry, and certain military tasks, where it is critical that particular individuals make correct judgements for the safety of themselves and others. For such people, it is possible that a decision support system tailored to their own individual way of working can help them in their decision making tasks.

Decision makers in such environments can be viewed as using naturalistic decision making methods, and cognitive models of such people performing a simulated electronic warfare task have been implemented using the Soar architecture, for inclusion in a decision support system. This task was chosen because it displays the characteristics most relevant to naturalistic decision making, namely rapid decision making in complex, real-time environments.

For decision making in the unstructured and dynamic real world, this approach of employing cognitive models of the users, embedded in a decision support system, appears to hold much promise to help them in this task.

Introduction

This paper describes the results of a research programme undertaken to produce *customised* intelligent decision support systems (IDSS) for naval personnel working in command and control environments, such as Electronic Warfare (EW) or Anti-Aircraft Warfare.

It can be argued that in such environments, decisions are made in a *naturalistic* manner (Klein *et al*, 1993). This refers to the process of how people actually make decisions in rapidly changing, complex, environments, and is one of the main factors behind this work It is characterised by the following eight factors (Orasanu & Connolly,1993).

[1] The first author wishes to thank DERA for the opportunity to carry out this work under contract CB/CSM1/2198/107

1. Ill-structured problems
2. Uncertain dynamic environments
3. Shifting, ill-defined, or competing goals
4. Action/feedback loops
5. Time stress
6. High stakes
7. Multiple players
8. Organisational goals and norms.

All of these are present to some (usually high) degree for naval personnel working in command and control.

Traditionally the techniques employed in decision support systems have been mathematical in nature, such as Bayesian statistics (Luo & Kay, 1989), multi-attribute utility theory (Riedel & Gordon, 1986), or some form of optimisation (Farrel *et al*, 1986) However, these systems have not always proved to be useful when applied to decision making under dynamic, real world uncertainty. Cohen (1993) states:

> 'Success, however, has been limited; the very features of real-world environments (the features 1-8 mentioned above) typically defeat the kinds of static, bounded models provided [by these technologies].'

and then goes on to say that:

> 'Such an approach may force decision makers to adopt highly unfamiliar modes of reasoning; as a result, aids may not be used, or if used, may be poorly understood; worse yet *they may fail to exploit user knowledge or expertise* that might facilitate adaptation to complex, novel situations.'

[Our emphasis]

This view has been supported by a later study (Sheppard *et al*, 1994), one conclusion of which was:

> 'The most promising developments in the area [of decision support] appear to be based on what have become known as naturalistic decision approaches to decision making. *These concentrate on attempts to understand how human decision makers actually make decisions in the complex real-world settings...*'

[Our emphasis]

While, as previously argued, most existing decision support systems may not be the best in certain situations, the work described here is not an attempt to replace such methods, but to augment these more traditional ways of implementing decision support, where the user's approach to a problem is ignored or replaced, by the introduction of a user-centred, problem-driven approach, which will aid the user's ability to handle problems in ill-structured environments, rather than replace it with inapt technologies.

The System

Figure 1 below is a high-level representation of the architecture upon which the IDSS, to support users in a naturalistic manner, is implemented. Briefly, a cognitive model of the user, (the Soar 'ideal user') which has been built from the user's responses in an ideal situation, is compared to the actions that the user is actually performing in the real-world, real-time situation. When the user starts to take a course of action that is different from how they said they intended to perform in this situation, then support is provided according to how the user said they would ideally handle this situation, thus providing the customised aspect of the IDSS.

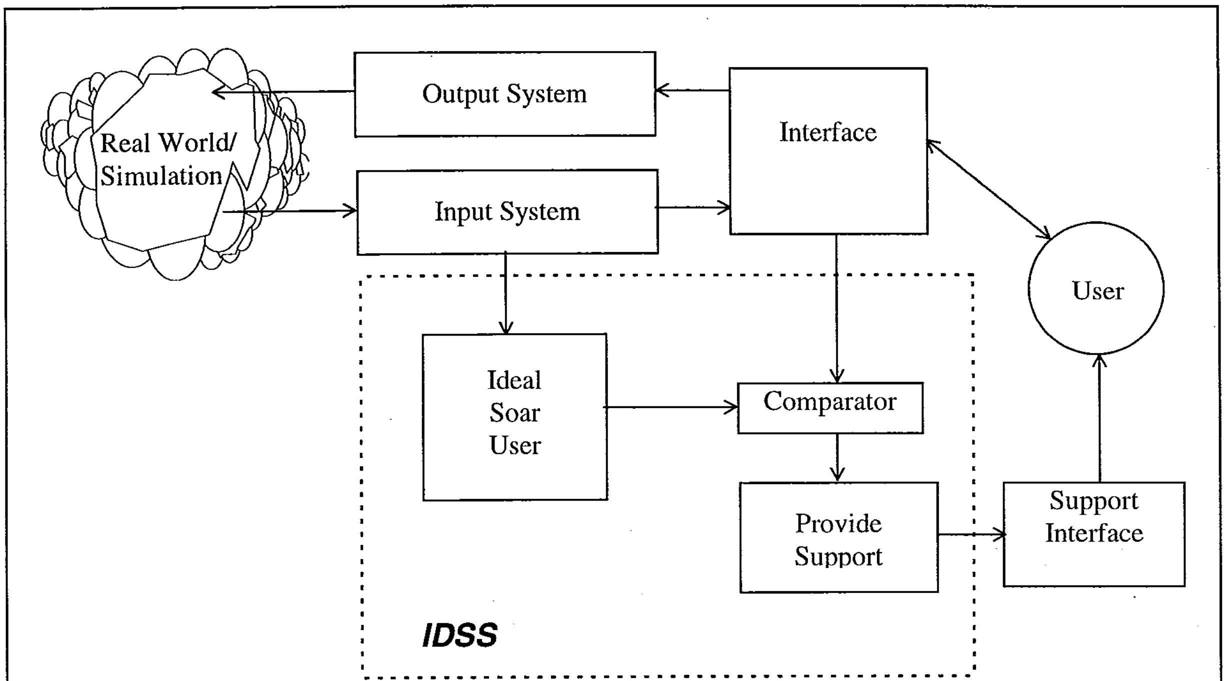

Figure 1 - Architecture of proposed intelligent decision support system.

Ideally, here, means that users have been able to work through the various scenarios they could meet in real-life but without any of the time or other constraints that they would be under if the circumstances under consideration actually occurred in a live command and control situation. They have the time to think about their ideal responses to the various scenarios presented to them, and so arrive at their optimal solution to the problem task.

The essential point here is that this is *not* the case of an expert system giving help to a novice user - the user *is* the expert. It is a system that is designed to help the expert back onto his or her right track if they deviate from it due to the pressures of the situation. The best person to help the expert is the expert themselves.

From the above it can be seen that the Soar model of the user acting under ideal conditions is one of the central features of the proposed system, and the rest of this paper reports on the progress and feasibility of this part of the work.

The Modelling Architecture

The approach in this work is based on the firm belief that the user's knowledge and expertise must play a central part in any IDSS designed to help them. Thus it was necessary to find a suitable architecture to enable this.

Two possible candidates were found that were based on a cognitive, rather than a 'technological' architecture: ACT-R (Anderson,1993) and Soar (Laird *et al*, 1987). (A comparison of these two can be found in Johnson,1996). Soar was eventually chosen for many reasons, both theoretical and practical. On the theoretical side, the underlying philosophy of Soar is of a cognitive architecture based on the hypotheses of the knowledge level (Newell,1981) and problem spaces (Laird *et al*,1987), and on the practical side there is a large and varied body of work already carried out by its research community, and it is well supported by its developers.

Soar provides a framework for problem solving, where all problem solving activity is formulated as the dynamic execution of an operator (or reactive plan) hierarchy. There is a continual selection and application of operators to a state in order to achieve some goal, and one of the central tenets of Soar is that all deliberate goal oriented behaviour can be cast as the selection and application of operators in various task-related problem spaces. The aim of a Soar model is to select and apply operators that will transform the initial state into a desired or goal state

Given this, the use of Soar for implementing user models implies the adoption of a particular perspective concerning the nature of problem solving, which in turn drives the analysis of the user's approach to the task. This will include their prior knowledge, their goals and actions, and the external parameters of the task. The user's knowledge is not, in Soar, pre-compiled into procedures for solving the task. Instead, the knowledge is encoded as a series of 'independent' condition-action type rules, and these themselves are divided into problem spaces. In a situation where there is insufficient knowledge to make progress towards the goal (e.g. an operator cannot be selected), then another problem space is automatically generated by the Soar architecture, in which the search for the required knowledge to continue towards the goal can take place. Soar thus provides a reactive level, a deliberation level, and a reflective level in its problem solving capabilities, which enable it to reason about wider aspects of the task (Rosenbloom *et al*, 1993a, 1993b; Newell, 1990).

Task Description

The task selected upon which to model the user was an EW task, which although perhaps simpler than a real-life command and control task, does display the characteristics of rapid-decision making in a complex, real-time domain, which are of most interest to this research programme.

The program used provides a realistic simulation of a surface ship under attack from one or more missiles of various types, e.g. radar-homing, heat-seeking. The task is for the user to avoid the ship being hit by the missiles by changing the ship's course and/or speed and using chaff and torch deployments to decoy the missiles away from the ship. The user can also zoom the display in or out to view the scene in more or less detail. The program offers different types of ships and missiles, and scenarios can have either predetermined or random starting parameters (such as number and type of missiles, their starting positions, wind speed, etc.) and three different levels of difficulty. The user's success, or otherwise, is recorded by the system.

However, rather than initially attempt to develop a model with a complete set of EW skills, it was decided to start with a simpler problem using only one ship type and one missile type (chaff decoy).

The User

Several options were studied for choosing a user of this system, and these have been described in detail elsewhere (Cook & Ashford,1995). It was finally decided to study a naive user's performance on the easiest of the tasks. The reason for choosing such a user was that the only sources of knowledge available to them could be easily controlled. The user was allowed many attempts at the task, in order to become expert at the task and thoroughly familiar with it, before the knowledge acquisition phase began. Training notes describing the task, operating instructions, and the functionality of the program were given to the user and formed their initial knowledge of the task.

The Modelling Process

For each scenario, the subject was encouraged to articulate thoughts about the task, the strategy used, and any knowledge that was found to be helpful. After each scenario, a debriefing was carried out to obtain more knowledge about the strategy and its success, or otherwise, for that scenario. A video recording was also made of the task display, which showed the results of the subject's actions, thus allowing a match of the verbal protocol data to these actions, which together gave a more complete account of performance on the task. A more detailed account of this process can be found in Cook & Ashford (1995).

As mentioned above, the underlying theory of Soar drives the knowledge acquisition process. There is a need to formulate the task in terms of goals and problem spaces, and the relationships between them. The details of the goals and the user's actions during the simulation were found to yield insight into the cognitive processes required to produce the actions.

In practice, it was found that the verbal protocols were the main source of information in the knowledge acquisition stage. Although there are arguments both for and against using these protocols (Ericsson & Simon,1993), without them, it would be necessary to infer the subject's behaviour entirely from observed problem-solving behaviour (such as the video recordings, mentioned above) (Anderson, 1993), which would provide an even more incomplete model of the processes involved in performing the task.

From these sources of information it was possible to identify the following:

- the initial state from which the problem solving began.
- the various problem spaces in which problem solving took place;
- the main goal, and the goals in these problem spaces;
- the operators which were applied to move towards the goal in each problem space;

The Model

From the results of the knowledge acquisition phase, seven problem spaces were usually identified, as shown in Figure 2. This is self-explanatory except perhaps for the chaff problem spaces. It may happen that a chaff canister is dud, so there is a need to check this and fire another if that is indeed the case.

From these, a cognitive model of the user was constructed, which typically consisted of approximately 150 rules distributed among these problem spaces.

Once the scenario is running, Soar will come to its decision about the next operator to choose, but will *not* apply it instantly. This is because the world could change, e.g. the wind may change direction, between Soar making its decision and the user making theirs. Thus, Soar waits, with its choice of operator 'on hold' whilst monitoring the events taking place. If any changes occur that impact on the current choice of operator, then the proposal conditions for this operator will no longer be true, and Soar can then select another operator based on the 'new' current situation. This again, will be placed 'on hold'.

Eventually, the user will input their choice of action and the 'on hold' operator will then be applied. The system traps the user's input and compares it to Soar's choice. If they agree, the choice is passed through the system; if they do not agree, then the system can flag up that a disagreement has occurred and wait for the user to either reconsider

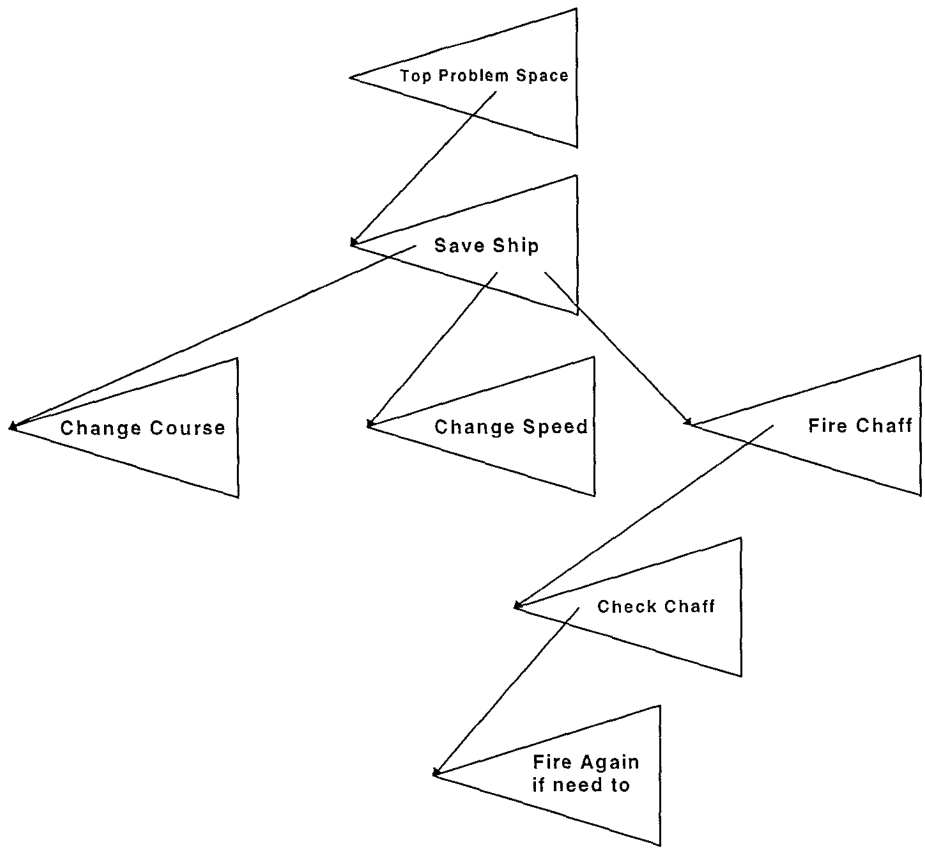

Figure 2

their choice, or override the choice made by Soar.

Testing the Model

Soar comes with Tcl/Tk (Ousterhout, 1994; Welch,1997) which is smoothly integrated in the development package. Thus, for testing purposes, it was possible to implement a simulator in Tcl/Tk which had the same functionality as the one upon which the user had worked. This made it easier to trace the model's actions and produce timings than using the original program. It was thus possible to recreate the scenarios that had been used to gain the information from the subject and so confirm that the model did indeed match the subject's actions and timings in these scenarios.

Also, a prototype IDSS, based on the architecture in Figure 1, and incorporating this Soar model as the 'Soar ideal user' has been developed, and shows a 'proof of concept' of the idea of being able to provide customised decision support. User response has been positive to this aid, even if it only confirms that they are doing what they said they should be! However, as mentioned below, this response has only been informal in nature, due to the lack of more analytical tools to measure feedback from the users.

Future Work

Having shown that the idea of producing user models is practical for IDSS applications, the work is now being extended to handle other, more complex, scenarios, and is being applied to other naval command and control situations that need to provide support for the user.

Work is also being done on incorporating and adapting the Debrief (Johnson,1994a, 1994b) facility which allows the user to query the model for its reasons its actions. This will help in validating the Soar model.

There are also aspects that need further work on the human factors/interface side: for example, should the output from the ideal user model be available continuously to the user (in which case s/he might come to rely on it unthinkingly) or should it be only available when asked for? Also, how should this information be presented?

Another area that needs exploring is how to measure the effectiveness of such systems as the one proposed here. Response to this work, although encouraging, was informal, and thought needs to be given as to how to make this assessment more objective.

Conclusions

The work presented in this paper shows that the idea of producing customised decision support to naval personnel performing command and control is both feasible and practical. The use of Soar combined with a naturalistic approach to such decision making has also been shown to be a promising approach to the development of cognitive models of users who must make rapid decisions in the unstructured and dynamic real world.

References

Anderson J.R. (1993) *Rules of the Mind* Lawrence Erlbaum, New Jersey

Cohen M.S. (1993) The Bottom Line: Naturalistic Decision Aiding. In Klein *et al* (1993)

Cook C.A. & Ashford A.J. (1995) Knowledge Acquisition for Soar Modelling of Rapid Decision Making in CIS: EW Chaff task. (Tech Rep No. *DRA/CHS/HS3/WP95/005/001*) DRA UK

Ericsson K.A. & Simon A.H. (1993) *Protocol Analysis: Verbal Reports as Data* MIT Press Cambridge Mass

Farrel R L, Bonder S, Proegler L D, Miller G, Thompson D E, (1986) Capturing Expertise: Some Approaches to Modeling Command Decisionmaking in Combat Analysis. *IEEE Syst., Man, Cybern., SMC*-16(6).

Johnson T.R. (1996) Control in Act-R and Soar. *Proceedings of the First European Conference on Cognitive Modelling* (pp201-208) Technische Universität Berlin

Johnson, W.L. (1994a) Agents that Explain Their Own Actions. *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioural Representation.* Orlando FL.

Johnson, W.L. (1994b) Agents That Learn to Explain Themselves. *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Seattle WA.

Klein G.A., Orasanu J., Calderwood R., & Zsambok C. (Eds) (1993) *Decision Making in Action: Models and Methods.* Ablex Publishing Corp. Norwood, New Jersey

Laird J.E., Newell A., & Rosenbloom P. S. (1987) Soar: An Architecture for General Intelligence. *Artificial Intelligence*, 33, 1-64

Luo R C & Kay M G (1989) Multisensor Integration and Fusion in Intelligent Systems. *IEEE Syst., Man, Cybern., SMC*-19(5).

Newell A. (1981) The Knowledge Level *AI Magazine* 2(2), 1-20

Newell A. (1990) *Unified Theories of Cognition* Harvard University Press, Cambridge Mass

Orasanu J. & Connolly T. (1993) The Reinvention of Decision Making. In Klein *et al* (1993)

Ousterhout J. (1994) *Tcl and the Tk Toolkit* Addison-Wesley

Riedel S L & Gordon F P (1986) Utilization-Oriented Evaluation of Decision Support Systems. *IEEE Syst., Man, Cybern., SMC-*16(6).

Rosenbloom P. S., Laird J.E., & Newell A. (Eds) (1993a) *The Soar Papers Volume 1* MIT Press, Cambridge Mass

Rosenbloom P. S., Laird J.E., & Newell A. (Eds) (1993b) *The Soar Papers Volume 2* MIT Press, Cambridge Mass

Sheppard C., Campion J. & Scard N. (1994) *Tasks, methods and tools for the study of decision making* (Tech Rep No. *DRA/CIS/CAD5/CR94006/1.0*) DRA UK

Welch B.B. (1997) *Practical Programming in Tcl and Tk.* Prentice-Hall