Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

SIMULATION OF INFORMATION FLOW IN ORGANIZATIONS

Douglas W. Jones

Department of Computer Science
University of Iowa
Iowa City, Iowa 52242, U.S.A.

ABSTRACT

The Icarus simulation system is the result of an experiment in embedding small rule-based expert systems into a discrete event simulation environment. The goal of this work is to allow the use of expert systems to model the intelligent behavior of people that are included in the system being simulated. The major problems we have encountered in building the Icarus system have resulted from unexpected and fundamental interactions between expert systems and our discrete event simulation framework. These interactions are the focus of this paper.

1 BACKGROUND

Quite a bit has been written about the application of artificial intelligence techniques to simulation, but the vast majority of this has revolved around the use of expert systems to either formulate models or aid in their testing or interpretation. Large numbers of examples of such work are documented by Elzas, Oren and Zeigler (1986).

Simulation models that include complex human behavior are rare, but there are some important examples. Perhaps the most impressive examples are battlefield simulations, for example in work by Davis (1986). Another important example is Novick's SASO system (1990).

None of the work we have found has documented the problems we have encountered in combining discrete event simulation with expert systems. We speculate that the reason for this is that most prior efforts have been based on crude models of time, for example, as in SASO, or on a combination of ad-hoc methods, as in many battlefield simulations we have seen.

2 OVERVIEW OF ICARUS

All systems modeled by Icarus are described in terms of a set of global state variables, a set of people, the beliefs those people hold about the state variables, and the geographic constraints governing communication between the people. We use a strict discrete event simulation framework where the state of the model is static between events. Thus, changes in variables, beliefs, and location occur only at discrete instants in time.

Every event in an Icarus model is initiated by some person in the model. A person may move from one place to another, a person may speak what they believe about some variable, a person may observe the actual value of some variable, or a person may change the actual value of a variable. When a person speaks, all of the other people in the same place will hear what is spoken, thus allowing them to change their beliefs. When a person observes, they may change their belief about the variable they observe. Finally, a person may draw a conclusion, changing their belief about a variable.

The behavior of each person in an Icarus model is specified by a small expert system. This describes how a person draws conclusions and decides to act based on their beliefs, observations and what they hear. In addition to modelling the relationship between stimulus and response, the expert system for each simulated person determines the reaction time of that person.

3 THE FUNDAMENTAL PROBLEM

In Icarus, we evaluate the expert system for each person in response to events involving that person. These occur whenever the person moves, whenever some other person arrives in the same place, whenever some other person speaks, and whenever the person in question draws a conclusion.

In general, the reaction to an event takes simulated time, and this causes problems because we must deal with events that take place between the time the expert system modelling a person recommends some action and the time that action takes place. In general, each evaluation of the expert system between the time an action is recommended and the time the action takes place will result in the same action being recommended again.

This causes no problem if the action recommended is idempotent; in which case, applying the action two or more times has the same result as applying it once. Non-idempotent actions, on the other hand, cause severe problems. Unfortunately, such actions are common; for example, consider a person who realizes that they have underestimated some variable. Repeated evaluation of that person's rule base can easily result in repeated increments to the person's estimate.

In general, this problem does not depend on the details of the expert system being used. Our expert system is confined to forward reasoning for the same reasons as Novick's SASO system [1990]. In addition, in order to allow detailed timing of our models, we associate an explicit time delay with each inference rule. With each event involving a person, we evaluate all rules in that person's rule base, and we perform the resulting actions in parallel in simulated time. Each such action may result in new expert system evaluations, thus allowing for inference chains.

4 SOLUTION APPROACHES

Initially, we tried to solve the problem described above with a variety of ad-hoc mechanisms. For example, we included special code to prevent expert system evaluation when an inference does not change any belief, and we included special cases for moves initiated during a pending move.

Only later did we understand that the fundamental problem is a mutual exclusion problem. Each rule-driven person is, in effect, a process, and we allowed these processes to interrupt each other at random times without allowing any control over atomic actions or critical sections.

One solution to this problem is to schedule all actions scheduled at the same time as a result of the same rule as an atomic unit. This prevents other actions or expert system evaluations until the entire co-scheduled set of actions have been completed. Although simple to state, this solution has proven to be surprisingly difficult to implement.

A second solution we have begun to explore involves associating latency with rules. Between the time a rule fires and the end of the associated latency interval, latent rules are removed from the rule bases of the associated people. The latency of a rule is generally the same as the delay, but we have found some cases where making these differ is quite useful.

Finally, we are experimenting with a general prohibition on scheduling more than one change to any one aspect of the simulation state until previous changes to that aspect have either been cancelled or completed.

Icarus has raised other problems, primarily in the area of mixed continuous discrete simulation models. We have found that most of the problems that interest us involve continuous components, and we have begun to investigate adding improved tools to Icarus to support this.

ACKNOWLEDGEMENTS

Phillip Zee and Madhavi Reddy did much of the programming work involved in developing Icarus. This work was supported by the Army Research Institute contract MDA903-90-C-0154.

REFERENCES

Davis, 1986. Applying Artificial Intelligence Techniques to Strategic-Level Gaming and Simulation. *Modelling and Simulation Methodology in the Artificial Intelligence Era,* ed. M. S. Elzas, T. I. Oren and B. P. Zeigler, Chapter VI.2. Amsterdam: North Holland.

M. S. Elzas, T. I. Oren and B. P. Zeigler, 1986. *Modelling and Simulation Methodology in the Artificial Intelligence Era,* Chapters I and II. Amsterdam: North Holland.

D. G. Novick, 1990. Modelling Belief and Action in a Multi-Agent System. *AI, Simulation and Planning in High Autonomy Systems,* ed. B. P. Zeigler and J. Rozenblit, 34-41. Los Alamitos: IEEE Computer Society Press.

AUTHOR BIOGRAPHY

DOUGLAS W. JONES is an Associate Professor of Computer Science at the University of Iowa. He received his BS degree in physics from Carnegie Mellon University in 1973, and his MS and PhD degrees in computer science from the University of Illinois in 1976 and 1980 respectively. His research interests focus on distributed simulation, pending event set data structures, and high performance simulation of computer architectures.