4X−7

# Integration of Expert Systems and Simulation

Rohit Gupta Yasuyuki Tauchi
SECOM Intelligent Systems Laboratory
SECOM Co Ltd

## 1 Introduction

There has been a growing interest concerning integrating expert systems with simulation. In most of the systems proposed so far, the expert system and the simulation are two separate programs. We have developed a language which can be used to create an application which can dynamically switch between running as an expert system and running as a simulation. Using one language for both systems has several advantages such as consistency in data represention and the reduction of communication overheads. We have used this system to implement a prototype training simulator.

## 2 Simulation and Expert Systems

In recent years there has been a growing interest in integrating artificial intelligence with simulation [WL89]. The field which has received the most attention is the integration of expert systems and simulation[O'K86]. Two specific areas which we are interested in are:

- Using an expert system within a simulation to simulate human behavior.
- Using an expert system as an Intelligent Front-End (IFE). An IFE is an interface between the simulation user and the simulation program. It assists the simulation user with configuring the simulation and analyzing the output [O'K86].

Almost all of the methods proposed for integrating expert systems and simulation assume that the simulation and the expert system are two separate programs. While this has the advantage of modularity, it also has several disadvantages. One disadvantage is that there is a storage overhead because common data must be stored in both systems. Another disadvantage is that there is a communication overhead.

Our approach is to use one language for both purposes. We have developed TecSim, a simulation language based upon a real-time expert shell. TecSim can be used to develop an expert system, a simulation program, or both. The unique feature of TecSim is that it can dynamically switch between the simulation program and the expert system.

## 3 TecSim

TecSim is based upon the Tecalbe expert shell. Both systems are written in EA+[1], a real-time superset of the Scheme programming language (see Figure 1). TecSim can be used to write a simulation using only rule-based programming. Two types of rules can be specified: *pattern rules* and *event rules*. A pattern rule will be fired if its left-hand-side matches current working memory elements and if it is selected from the conflict set of matching rules. An event rule will be fired if an event[2] occurs and there is a rule whose left-hand-side matches the name and parameters of the event.

TecSim supports commands to enqueue simulation events, dequeue simulation events, pause the simulation, resume the simulation, collect and calculate statistics, and generate simulation events. A simulation event consists of the event name, the event time, and the changes to the working memory which will be made by this event.

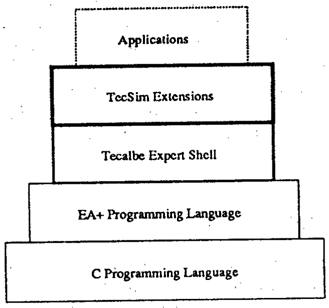

Figure 1: Implementation Hierarchy

The EA+ language supports lightweight processes. TecSim uses four of these processes for program control (see Figure 2). In addition, TecSim supports a boolean variable which can be used to connect or disconnect the **simulation process** from the rest of the system. If the **simulation process** is not attached, then the system functions as an expert system. If the **simulation process** is attached, then control proceeds as follows:

TecSim supports commands to enqueue simulation events, dequeue simulation events, pause the simulation, resume the simulation, collect and calculate statistics, and generate simulation events. A simulation event consists of the event name, the event time, and the changes to the working memory which will be made by this event.

1. The **simulation process** checks to see if the current simulation time (based upon the Unix system clock) is greater than or equal to the time of the next simulation event. If not, then the **simulation process** uses the EA+ event raising facility to activate itself again[3].
2. If it is time for the next event, then the working memory commands associated with this event are executed. Control is then passed to the **pattern rule process**.

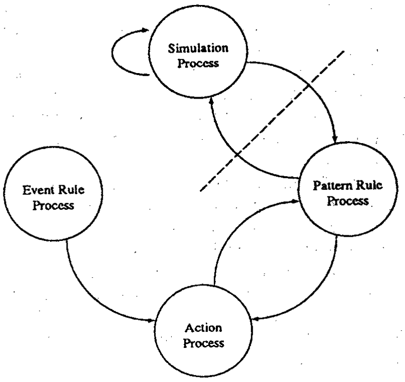

Figure 2: TecSim Control Loop

[1] Tecalbe and EA+ were both developed at Secom Intelligent Systems Laboratory.

[2] *Event* refers to an EA+ event. In EA+ one can specify how to respond to a particular event. An EA+ event can either be raised from within the EA+ program or from an external source.

[3] The granularity of the simulation loop is a user-defined variable which can be modified dynamically.

3. The **pattern rule process** attempts to find a matching rule. If a matching rule is found[4], then control passes to the **action process**. If no matching rule is found then control returns to the **simulation process**.

4. The **action process** executes the right-hand-side of the rule. Control then returns to the **pattern rule process**.

In the case of EA+ events, the **event rule process** finds the appropriate rule and the **action process** executes the right-hand-side of the rule. Control is then passed to the **pattern rule process**.

When using TecSim for simulation, execution will continue until either an *end-simulation* simulation event is found or until a user-specified time period has elapsed. When using TecSim as an expert system, execution will be completed when no matching rules can be found.

## 4 Training Simulator

Secom provides security services to homes and offices. Subscribers are connected to a control center via the Secom network. When an alarm (e.g., intrusion) occurs at a user's premises, an alarm signal is sent from the user's building to the nearest control center. At the control center, an operator monitors these alarm signals and sometimes dispatches a security guard to respond to the alarm. The operator guides the security guard through all steps of the response procedure for that type of alarm.

We are developing a training simulator for the operators. The simulator will simulate all elements of the real-world environment. We have used TecSim to develop a first prototype of the training system. The entire simulation program is written using rules only. Rules are primarily used for the following two purposes:

- To simulate the behavior of a single security guard.
- To assist the user via an IFE.

The training system first runs as an expert system. The user supplies configuration parameters and the expert system checks to make sure that the parameters are correct and within appropriate ranges. Assuming the parameters are acceptable, the system then toggles the boolean variable, and the simulation starts to run. When the simulation is finished, the system becomes an expert system again. It then uses rules to display the output and to answer queries about the output.

## 5 Event-Tracking Simulation

TecSim currently supports **fixed time-step simulation**[5]. Fixed time-step simulation is useful for *training* purposes. However, if we wish to use TecSim for *analysis* purposes then we will need to support **event-tracking simulation**.

Figure 3 shows the design for an alternative version of TecSim. There are now two EA+ processes for simulation. Thus, the system can be used for fixed time-step simulation, event-tracking simulation, and expert system applications. In addition, the system can dynamically switch between the three.

An example of a possible application is the above training simulator. Assume that event-tracking simulation is used in the real-world control center system to determine which security guard to dispatch. We can use fixed time-step simulation for training purposes (as described above). However, when the operator needs to choose a security guard, the system will switch to event-tracking simulation. The event-tracking simulation will determine which guard to dispatch. The system will then switch back to fixed time-step simulation in order to continue with the training simulation program.

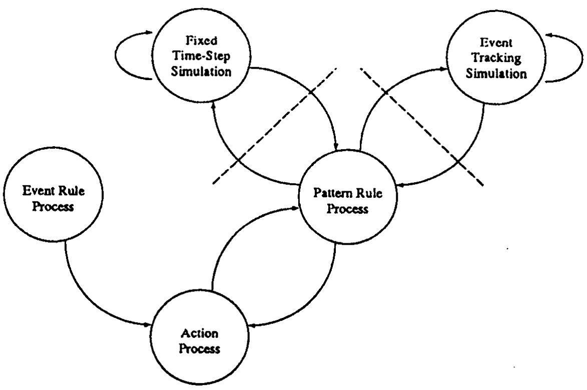

Figure 3: Fixed Time-Step and Event-Tracking

## 6 Conclusion

We have designed and implemented TecSim, a tool which can be used for developing a simulation, an expert system, or both. In addition, we have used TecSim to implement a prototype of a training simulator. The features of TecSim are:

- It can dynamically switch between running as an expert system and running as a simulation. Thus, there is reduction in storage and communication overheads.
- It enables us to use rule-based programming to simulate human behavior within a simulation. Other behavioral simulation methods (e.g., Petri Nets) have the disadvantage that one must specify the overall **system** behavior. Rule-based programming allows one to use a more **modular** representation for behavioral simulation.
- An IFE and the simulation program can be developed with the same language. Traditional IFEs assist the user before and/or after the simulation, but not *during* the simulation run. Our approach allows for assistance during the simulation run also.

## References

[HP89] Stewart V. Hoover and Ronald F. Perry. *Simulation: A Problem-Solving Approach*, chapter 2. Addison-Wesley Publishing Company, 1989.

[O'K86] Robert O'Keefe. Simulation and expert systems - a taxonomy and some examples. *Simulation*, 46(1):10--16, January 1986.

[WL89] Lawrence E. Widman and Kenneth A. Loparo. Artificial intelligence, simulation, and modeling: A critical survey. In *Artificial Intelligence, Simulation, and Modeling*, chapter 1, pages 1--44. John Wiley & Sons, Inc., 1989.

[4] Conflict resolution strategies are used when more than one matching rule is found.

[5] In *fixed time-step simulation* the simulation clock is incremented by one time unit during each iteration of the simulation control loop. In *event-tracking* simulation the simulation clock is updated to the time of the next pending event[HP89].