Proceedings of the 1996 Winter Simulation Conference
ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain

UTILIZING AN EXPERT SYSTEM FOR CONFLICT RESOLUTION IN A "TRAIN SIDING SELECTION" SIMULATION MODEL

Gregory J. Allen
Khaled M. Mabrouk

The Model Builders, Inc.
343 South Main Street
Suite 217
Ann Arbor, MI 48104

Michael L. Weigel

Union Pacific Railroad
116 Dodge Street
Room 406
Omaha, NE 68179

ABSTRACT

Simulation is an excellent tool for evaluating alternative configurations for system flow and/or logic. Occasionally, the complexity of a system presents problems with respect to simulating and improving the performance of that system. When confronted with these complexities, the logic requirements can easily become more difficult to develop than the effort to understand the actual system logic.

Expert systems, on the other hand, are better suited for reflecting complex decision making. Many times this means solving complex problems which can only be stated in written words, therefore extending computing power beyond the mathematical and statistical constructs of simulation and conventional programming. As a result, expert systems can be significantly easier to modify, for experimentation purposes, than a set of complex "IF-THEN" statements.

Through the development of an expert system and its interface to a simulation, the problem of modeling complex systems can be encountered with improved efficiency. The end result is an increased level of simulation performance and customer satisfaction.

INTRODUCTION

The railroad industry is typical of the complexities which often prevent the utilization of computer simulation as an effective decision support tool. As such, it has been selected as the focus industry for testing the feasibility of an interfaced expert and simulated system.

The success of computer simulation as a decision support tool in other industries, coupled with the advances of expert systems in the railroad industry has indicated that it would be advantageous to interface the two respective tools. Previous attempts at combining these two technologies have been performed with relative success. However, none of the previous attempts dynamically connected the expert system and the simulation while the simulation program was operating. Rather, the expert systems were only called before or after the simulations had executed. Specifically, The Model Builders' expert/simulation interface performs real time decision support and analysis for a simulated railway system.

SCOPE AND OBJECTIVES

Conflict resolution is an integral part of most railroad industry simulation efforts. The issue becomes even more important when developing a simulation model for a train corridor where decisions need to be made continuously for each train on whether it should take a siding or stay on the main line. The simulation model that was developed, consisted of a decision support tool capable of reflecting some very basic routing procedures used in a railway system. Furthermore, a small scale expert system was developed and interfaced to provide the simulation's more complex, real time decision logic. It was the interaction between these two systems that was monitored to determine whether the potential value of sharing these applications does indeed exist.

METHODS AND MATERIALS

This project was initiated with the research of information pertaining to simulation, expert systems, software interfaces, and the decision logic behind a simple train conflict resolution module. The collection methodologies consisted of a material search at the University of Michigan libraries, GMI library, and the

Ann Arbor Public Library, as well as conversations with people working in the railroad and expert system industries. Based upon the information gathered through these searches, as well as other economic considerations, CLIPS™ was selected as the type of expert system software to reflect the decision logic needed for the railroad industry. The ProModel™ software package was selected as the tool for simulating the rail system. Visual Basic™ was chosen to provide the interface between CLIPS™ and ProModel™. Upon completing these preliminary stages, the simulated railroad model was constructed first, followed by the conflict resolution logic, and finally the Visual Basic interface for connecting ProModel™ with CLIPS™ was developed.

DESIGN OF THE TRAIN SIDING SYSTEM

A mainline rail and siding system is used to process train meets that occur when two trains traveling on the same main line are moving towards one another. The simulated system is structured so that one train can divert from the main track to a siding and let the other train pass on the main line. A train exits the main by moving into an available siding.

A satisfactory simulation model was built based on the design structures indicated through literature searches and interviews with railroad industry experts. These inquiries helped determine the appropriate level of detail to adequately test the simulation and expert system interface. Ultimately, a physical design was suggested and implemented consisting of the following elements:

Main Line: The main line is the portion of the track upon which trains primarily travel. The trains only leave the main line in the event of a traffic conflict, e.g. another train. The main line selected for the model was 26 miles in length.

Sidings: The length of the main rail allowed enough room to include four sidings. Each siding spanned a distance of 1.5 miles. These sidings were located at equal intervals but on alternating positions along the main line. Siding number one was placed 2.5 miles from the western endpoint on the main rail. The west end of each of the remaining three sidings were placed at 5 mile intervals from the east end of the previous siding, with the east end of the fourth siding located 2.5 miles from the east-most endpoint along the main rail. These intervals between sidings were intentionally given enough capacity for the entire length of a train entity. The multiple sidings modeled created the complexity needed to adequately test the capabilities of the decision logic being utilized.

Switches: At each of the eight points connecting the sidings to the main rail were the switching devices. These switches were where trains could merge with and exit from the main rail and track sidings.

Blocks: Tracking a train's simulated movement through the main rail and sidings, was accomplished by assigning one of 16 possible block numbers that identified a train's current location. The blocks were used to help communicate the status and location of a train at any given time during the model's execution. Essentially the blocks served as points of reference, that were checked by the expert system and implemented by the simulation. All of the information pertaining to trains traveling within a particular block was stored in an array call Block_Array. It was the Block_Array which contained the information communicated between the simulation and expert system.

Intermediate Signals: Intermediate Signals were added to the model at the middle of each 5 mile main rail sections between each siding, creating two block occupancies. Although these signals had no functional capabilities they were necessary to provide a more precise tracking of a train's location within the model.

Measurable Units: Time units were simulated in terms of minutes and distance units were measured in terms of feet.

Entities: In this particular model, the entities being processed were the trains. There were two types of trains for this model: loaded and empty. Every train arriving into the rail system was either assigned an attribute called "empty", or assigned an attribute called "loaded". These attributes were the identifiers used to evaluate the train conflict resolution logic and return a decision based on the current and conflicting trains attributes. The loaded trains were assigned the highest priority and always traveled in a easterly direction. The empty trains were given lower priority and always traveled in a westerly direction. Both the empty trains and loaded trains were given a length of 6,500 feet.

Train Performance: For the purposes of simplification, the model did not account for numerous train performance measures. Other models exist which can duplicate train performance, including velocity and acceleration for a given set of operating conditions such as horse power per trailing ton, track grade, track curvature, track conditions, and locomotive characteristics. However, this level of detail was not necessary to complete the proof of concept for an interfaced expert and simulated system. As a result, the velocities for both empty and loaded trains were maintained at constant speeds of 45 miles per hour on the main line and 10 miles per hour through switches and sidings (until the last car cleared the block).

Arrivals: Any time a new entity was introduced into the system, it was called an arrival. The number of train entities for each arrival was one. The total number of arrivals was 50 per day (25 for loaded trains and 25 for empty trains). The frequency of the arrivals was exponentially distributed over a 24 hour period, e.g., $E^{(24/25)}$. The loaded trains arrived at the western point on the main line and the empty trains arrived at the eastern most point on the main line. The first arrival occurred at time zero of the model execution.

Figure 1 contains a graphical representation of the entire track and siding system. Each of the numbers (1 through 16) represent a block. The track blocks numbered 3, 7, 11, and 15 were the sidings used in train meets and passes. Figure 2 contains the dimension details for a specific section of that simulated rail system. Finally, in Figure 3 the labels for the primary elements of the track and sidings are shown.

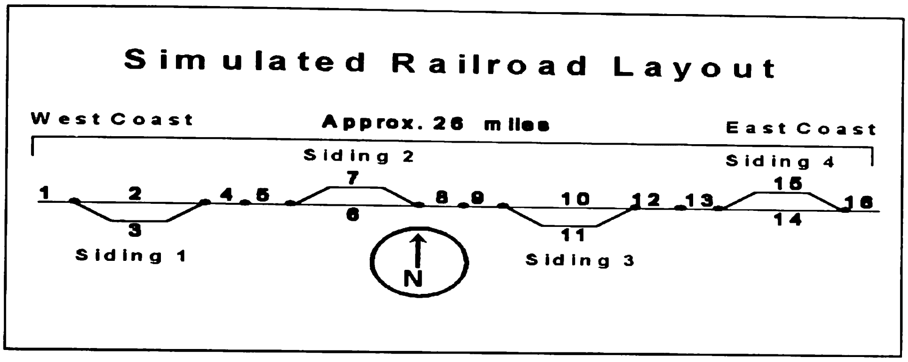

Figure 1: Graphical Representation of the Track and Siding System

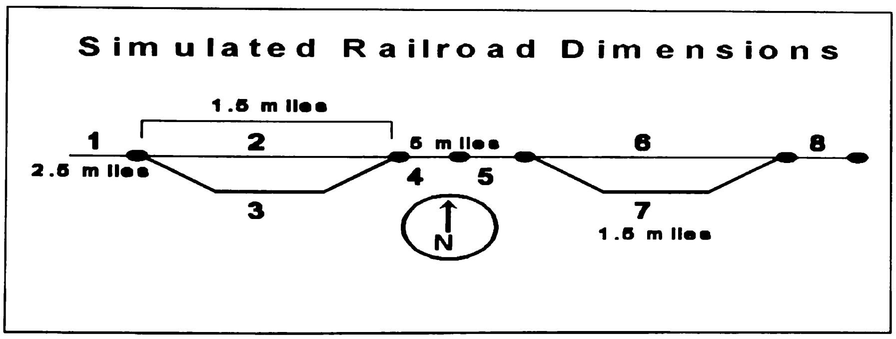

Figure 2: Dimensions of the Track and Siding System

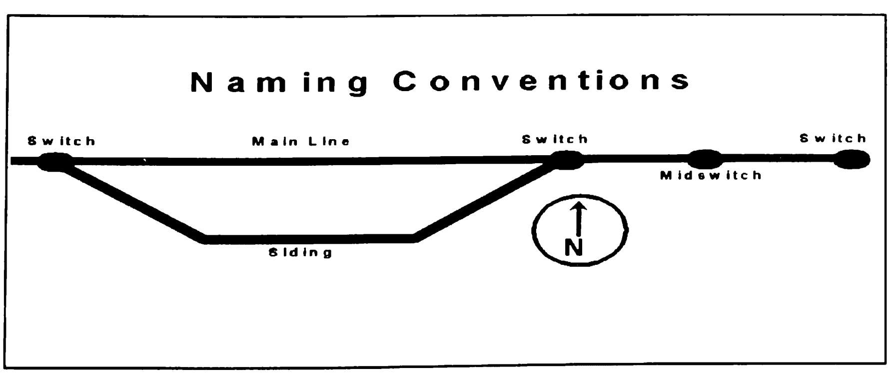

Figure 3: Naming Conventions of the Primary Elements

DESIGN OF THE DECISION LOGIC

The rule sets were developed based on three, isolated facts. Those facts were train type, train location, and train status. Train type indicated whether the train was loaded or empty. Train location indicated the block in which the train was traveling. Train status indicated whether or not the train needed a decision returned from the expert system.

The rule sets were divided into three modules. Each module was referenced at various points throughout the simulation and represented a different problem/solution set. Referencing a particular solution set was based on the facts triggered by the "current" facts for a train's type, location, and status. As a result, the simulation would only refer to a rule set if a train's attributes indicated a rule set was appropriate under the current circumstances. Thus, limiting the event of unnecessarily processing decision attributes.

Module One: Module one pertained to any trains on the main that were approaching a switch exiting a siding and mainrail connection. Its purpose was to keep trains, which were traveling in the same direction, from sharing the same block at any given time during the simulation.

Module Two: Module two pertains to each of the five mile long rail sections located on the main rail between one siding's entrance and the next siding's exit. Each of these sections were divided into two blocks using an intermediate signal. The division was made to create smaller track sections, making the pinpointing of a train's location more precise. Because these sections of rail lacked any siding outlets, trains were instructed to travel the length of the two rail sections without stopping.

Module Three: Module three contained the most complex decision logic to be evaluated. Its purpose was to issue instructions to trains traveling east and west on the main line that were approaching a switch at the entrance of a siding. In these instances the logic would look ahead two sidings to check the approaching blocks for their contents. Anything outside of the two siding cushion was not close enough to cause an eventual conflict, since there was still enough unoccupied main rail remaining given the level of traffic volume modeled. Loaded trains received a higher priority than empty trains, and in the case of a meet, the siding that minimized the delay to the higher priority train was chosen.

HOW THE APPLICATION WORKS

The simulation model defined some of the decision and execution logic for the train entities whenever they would enter a block's location. Essentially, the logic can be separated into operation logic and routing logic. Operation logic specified anything that needed to happen to an entity while in a location. Routing logic defined the outputs for each of the processes defined. Both the operation logic and routing logic were differentiated by location. Those locations were arrival/exit locations, switching locations, siding locations, and intermediate signal locations.

Within the arrival/exit locations, trains would enter and exit the model. The operation logic at these locations specified the appropriate routing logic based on the current train's attributes. If these attributes indicated that the train needed to exit the system, the appropriate Block_Array positions were updated and the train departed from the system immediately. However, if the train was a new arrival, it required that some decision logic to be carried out in the routing logic. The routing logic for new arrivals would wait until the Block_Array indicated that the arrival block was available before moving the new arrival forward into the system. Once the arrival block became available, the arriving train could move forward and update the Block_Array with information specific to its nature.

The switching locations were where the most complex processing logic was carried out. As a result, these locations used the expert system for the more sophisticated decision rules. Depending on the direction of the train traveling through a switch, the operation logic could assume two forms. One was for trains traveling through a switch to a main rail and siding entrance, and the other was for trains traveling through a switch exiting a main rail and siding.

Because the trains traveling through a switching exit were only looking one block ahead, they did not require the assistance of the expert system. These trains merely had to wait until the Block_Array indicated that the next block was available for routing. Once there was a vacancy, the simulation deleted the train's information from the current Block_Array position and updated the new position in the Block_Array with information.

Any trains traveling through a switching entrance had to look ahead several blocks before determining if they could stay on the main rail or if they had to take a siding and allow a on coming train to pass. Because they needed to process several blocks at a single time, their logic would have been difficult to develop with the simulation's typical operation constructs. Rather, the simulation's current status, maintained in the Block_Array, was written to a general text file every time a train arrived at a switching entrance. This text file would later become the information evaluated by the expert system. This process of passing information between systems required using ProModel's™ most powerful feature, the X-Sub command, to call a DLL file allow for communication with the expert system. A DLL is a library of functions that Windows™ reads and executes as a .EXE program requires them.

Once CLIPS™ had finished reporting its routing decisions to a general text file, ProModel™ reset the previous input file, and read-in the new information for execution. Actually, these routing decisions were integer values which were assigned to variable ROUTE statements within ProModel's™ operation logic. If the integer instructed the train to take ROUTE 1, allowing the train to stay on the main rail, ProModel™ waited until the designated route was available before moving forward. For example, if a loaded train was traveling east in Block 2 then it would have to wait until Block 4 AND Block 5 were both empty OR Block 4 was empty and Block 5 contained an east bound train. If the integer instructed the train to take ROUTE 3, telling the train to take a siding, ProModel™ merely updated the Block_Array and moved the train forward.

The procedures and logic mentioned above were used repeatedly at the switching locations throughout the model.

Those trains traveling within a siding location were merely instructed by the operation logic to route to a switch based on the trains' attributes. The resultant routing rule waited until the two blocks prior to the next switch were both empty before moving forward.

The final piece of processing logic dealt with the trains traveling through an intermediate signal. Remember, the intermediate signal's only purpose was to divide a large section of main rail into smaller, more manageable, distances. The signal's decision logic was relatively straight forward and was also performed by the ProModel™ simulation. There was no waiting time incurred by trains passing through an intermediate signal. The simulation simply updated the Block_Array to show the train leaving one block and entering another.

DOES THE APPLICATION MAKE SENSE?

The design and development of a real-time simulation and expert system interface have proven to be both a feasible and practical approach to modeling complex systems. A virtually seamless interface was developed between the two technologies. Embellishing the programming constructs of a simulation package with the functionality and flexibility of an exert system tool will significantly improve the ability to develop decision logic that reflects the random behavior and dynamic interactions of complex systems.

The remainder of the conclusions formulated from this project are based on a specific set of system performance measures. These measures were established prior the design and development phases of this thesis and were used to determine the future of the resultant application. Specifically, the performance criteria applied were:

1. Whether or not the interface could operate with real time capabilities, and
2. The relative effort required to use an interfaced application.

The following section contains each conclusion based on the aforementioned evaluation criteria.

Real Time Capabilities: Real time capabilities refer to the ability of the simulation application to communicate seamlessly, and not distort the simulator's clock, with the expert system during the execution of the model. Additionally, the communication link must occur with relatively little delay in clock time. With this particular simulation interface, a direct line of communication was initiated by the simulation and received by the expert system at any time during the model's execution. However, the interface was not seamless. In other words, there was about a second of delay in the simulator's clock experienced by the entity making a call to the expert system. This delay consisted of the simulated entity calling the expert system, the expert system evaluating the current situation, the entity waiting for a response, the entity receiving back a decision rule from the expert system, and the entity executing the resultant routing rule. Although the interfaced application didn't operate in a purely real time fashion, its delay was insignificant. Especially given the fact that the model was based on a train track and siding system, where a second of delay to the simulator's clock had little affect on the accuracy of such a large system. One conclusion could be that the interface "virtually" performed in real time. This virtual real time capability allowed the expert system to enhance the ability of the simulation to process complicated decision logic with very little affect on the model's performance. Refer to Appendices One and Two for graphical representation of the current and future real time capabilities.

Effort: The majority of the effort expended during this project related to the development of the interface between the simulation and expert system. This effort was the result of having to establish an interfaced line of communication using three different software products: ProModel™, CLIPS™, and Visual Basic™. The major challenge was finding a communication medium that all three packages could use jointly. However, the eventual interface was generic in nature, virtually eliminating the interface development and coding phase for any future projects where this type of technology would be applicable. That fact alone justified the amount of effort required to make the initial communication interfere work.

AUTHOR BIOGRAPHIES

GREGORY J. ALLEN is a systems engineer with The Model Builders, Inc. He is a graduate from the GMI Engineering & Management Institute, where he received his Bachelor of Science in Management with a concentration in information systems. Mr. Allen's current responsibilities include simulation modeling and analysis, the development of graphical user interfaces, and the development of expert systems.

KHALED M. MABROUK is a Simulation Strategist for The Model Builders. He has spent the last ten years assisting many companies in the U.S. & Canada to successfully implement simulation as a productivity improvement tool. His contributions include assisting these companies in choosing the appropriate software tool, training their employees on how to properly implement this tool, and managing simulation projects for them.

MICHAEL L. WEIGEL is a manager in the Network Planning Group at Union Pacific Railroad. he has developed simulation models in the areas of Rail Transportation, Aerospace Logistics and Manufacturing. He is currently involved in the development of Simulation Models used in the analysis of Railroad Terminal Operations. He received a B.S. degree in Industrial Engineering from the University of Nebraska and is a member of IIE.