Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

IMPLEMENTATION OF SPATIAL REPRESENTATION AND AUTONOMY IN MATERIAL HANDLING SYSTEMS

Lobna A. Okashah
Ralph V. Rogers

Industrial Engineering and Management Systems Department
University of Central Florida
P.O. Box 162450
Orlando, Florida 32816-2450, U.S.A.

ABSTRACT

This paper discusses the use of spatial autonomy for detecting and avoiding physical conflicts by using examples from a warehouse environment. Like many other real world environments that are the subject of simulation, a warehouse has numerous autonomous elements, including workers, AGVs and human-operated equipment. Traditional simulations have not fully considered the effects of this autonomy, which can have significant impact upon system performance and result in accidents, delays, and deviation from modeled behavior. In the past, modelers have treated these deviations as inputs to simulations, hidden in travel time distributions and delays based on real world observations.

(Okashah 1994) examined four shortcomings of discrete event simulation: (1) arc and node limitations; (2) predefined entity goals, behaviors and interactions; (3) variable frame of reference; and (4) space as a resource. This paper explains an approach to resolve these issues using spatial representation and autonomy. An autonomous approach to simulation results in deviations from idealized behavior becoming an output of--rather than an input to--simulation. Individual incidents are thus detected and resolved in simulation, and avoided in the real world.

1 INTRODUCTION

Analysts have used discrete event simulation to successfully optimize levels of resources in production systems, and to justify the acquisition of resources with certain characteristics, by accurately predicting long-term, steady state behavior. However, extreme aberrations from steady state behavior, and their causes, are not easily understood (Rothenberg 1989). In many cases, the simulation cannot even identify the specific aberration. As an example, aberrations in a warehouse could include equipment malfunctions, absent workers, an accident requiring repair of equipment or cleanup, or a computer shutdown in the production control system. Modelers do not often attempt to model these phenomena in traditional simulation because of several limitations that are fundamental to existing discrete event simulation modeling methodologies. These limitations include the following:

Arc and node limitations: only preprogrammed outcomes (including next events at the next node) are possible, and competition for resources is centrally controlled.

Predefined entity goals, behaviors and interactions: freedom to change goals, and the reasoning needed to select new goals, are missing.

Variable frame of reference: Time is the frame of reference, with each entity having its own de facto clock. Entities "shut down" during delays representing tasks.

Space as a resource: Space is treated as a resource, much like a machining center, even though it is inherently different from other resources.

(Okashah 1994) discusses each of these limitations in detail. This paper explains how two broad concepts, space and autonomy, must be introduced into simulation methodology to overcome these limitations.

2 THE SPATIAL TEMPLATE

In traditional simulation, a random number generator assigns start and duration times to mechanized stockpickers (Figure 1) picking inventory from shelves in aisles. The usual method of handling two such devices needing to access the same aisle at the same time is to introduce a delay, which indicates a waiting time for one of the stockpickers while the other completes its assignment. In the real warehouse, however, there are several distinct possibilities. First, the operators of the stockpickers can communicate, asking each other's destination and anticipated time in the aisle. Depending upon the response, either operator may decide to reorder his assignment and go to another aisle before returning to the blocked aisle. Alternately, one operator may park his machine (blocking the aisle) and proceed to execute his task on foot. Another possibility is that a collision may result if one or both of the operators do not see that their paths are blocked. The spatial template offers a solution that can adequately simulate each of these possibilities. Conversely, the delay offers only a rough--and possibly inaccurate--representation.

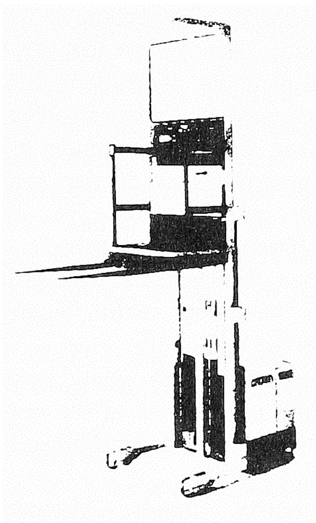

Figure 1: A commercial stockpicker; the operator rides on the lift portion of the machine

In our attempt to represent space in simulation, the focus is to enable all entities to view the space in which they exist and move about. Our goal is to use a frame of reference that does not change or move with time, but rather remains constant for all entities. This space is represented by what is called a spatial template (Rogers 93). Entities can use the spatial template to check the plausibility of future motion and use this information to help set goals.

The spatial template is not a black board as used by other researchers to display facts. Rather, it is a "thinking blackboard" that processes state variables from all of the other entities in the simulation, and returns to each entity a custom display of only the facts that are relevant to that particular entity. As an example, one can think of the departure screen at an airport as a blackboard because it lists all flights. A "thinking blackboard" implementation of a departure screen would display to each traveler only the information for that traveler's flight: person A would see information for the Albuquerque departure, and person B would see information for the Boston departure.

The processing performed by the spatial template, however, is more than simply sorting existing facts to determine which are relevant to particular entities. The spatial template performs reasoning of a spatial nature. It plots the trajectories of all objects based upon their last reported state variables (including position, velocity and acceleration) to project potential conflicts. The spatial template then reports any potential conflicts to their would-be participants, so that those participants can, autonomously, solve the problem.

A general implementation of the spatial template requires position, speeds and acceleration to be reported in global components. Spatial templates for individual applications, however, can be much simpler. In a warehouse, conflict detection between entities not in the same aisle is generally not necessary. For entities in the same aisle, position and speed variables in only one or two dimensions may be of interest. Thus, instead of comparing the trajectories of perhaps dozens of stockpickers, forklifts, AGVs, and people, only the few that are in the same aisle at a given time need to be compared. Figure 2 shows that entities in aisle 1 need to be compared only with other entities in aisle 1, 6 or 7, because they cannot collide with entities in aisles 2 through 5. In such a scheme, it is important to update state variables frequently enough so that the new position of an object is always in either the same aisle or a connecting aisle as its old position. Generally, simulated entities must update their state variables frequently enough so that the spatial template has useful information.

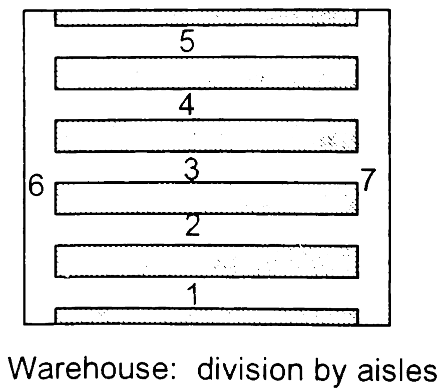

Figure 2: Division of space into logical physical subspaces for ease of collision detection

Conflict detection by the spatial template is best seen by example: Consider two mechanized stockpickers, A and B, each assigned to retrieve or place items on shelves in the same aisle. Each shelf has an identification number as noted in Figure 3.

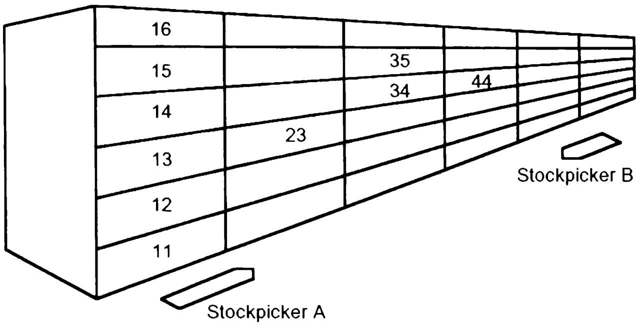

Figure 3: Two mechanized stockpickers accessing shelves in the same aisle

Each physical entity reports its position and intended motion to the spatial template. The spatial template's task is to check for possible interferences and report those to the parties that could be involved. Consider the five cases noted in Table 1.

Table 1: Five cases which must be considered for possible conflicts

Case	Stockpicker A	Stockpicker B
1	23	44
2	34 or 35	44
3	44	23
4	34	35
5	23	23

In Case 1, the spatial template detects no conflict and each mechanized stockpicker is permitted to proceed without warning. In Case 2, a potential conflict exists because the two machines will be very close together. (Modelers can use the guidelines of individual applications to define how close entities may be before a potential conflict is flagged.) Cases 3, 4 and 5 clearly involve conflicts, and in each case the spatial template would notify the two mechanized stockpicker operators that a potential conflict exists, the time and location of collision if they do not take evasive action, and the identity of the other party (or parties) in the conflict. It is up to the operators to resolve the conflict themselves, as they do in the real warehouse. The spatial template merely acts as the eyes of the objects in the simulation. The information it delivers to objects is the most important information that an alert person would see in the real world. The operators also have the option to ignore the spatial template's warnings. This corresponds to non-alert operators in the real world, or to alert operators working in darkness, fog or other poor conditions.

Informing entities of impending collisions has significant consequences. Traditional simulations are generally modeled so that collisions are impossible, but in the real world collisions do occur. This is due to the fact that real world space does not behave like a resource does in traditional simulation. One of the rules of the arc-and-node approach used in traditional simulations is that an entity will know if a resource is already being used, and will wait of its own accord. For two reasons, this rule does not necessarily apply to space. First, entities often act with imperfect information and may not realize that an occupied space is not free. This is especially common in conditions of poor visibility (such as blind corners) or if a sensor malfunctions. Second, space cannot control what entities use it. Whereas a bank teller can refuse to serve a customer who breaks into line, or an already occupied machine may have safeguards to prohibit the loading of a part, occupied space has no way of refusing access to an entity. Thus, while a space resource in traditional simulation may force entities to queue, in the real world those entities may collide.

3 AUTONOMY

Autonomous entities have the ability to set goals, plan their actions ahead of time, execute those plans, and set new goals and replan during execution. In a warehouse environment currently under study for development of our spatial template-based simulation, operators are issued lists of items to be picked from shelves. Each worker has exhibited a different approach to planning for the order, even though the lists are all issued in a preset order by the controlling computer. One worker often chooses to gather all items that he can carry on foot first, and leave those requiring a mechanized stockpicker for later. Another frequently chooses the opposite approach, and start his execution by looking for a mechanized stockpicker. A third operator tends to favor going aisle by aisle to gather all items, regardless of size, with minimal travel. A fourth operator has been observed to go aisle by aisle, but bypass and later return to those that appear congested with mechanized stockpickers. These various approaches demonstrate the need for flexibility in the programming of goal planning. Here the spatial template is useful in providing each autonomous entity with the information needed to carry out their original plans or to help them set new ones.

In traditional simulation, the ability to plan and replan is absent at the entity level. Evaluation and planning are done by a central controller, which is limited to the pursuit of preprogrammed goals that are typically uniform for all entities of a given type (such as stockworkers). Entities can only execute the actions assigned to them. This is clearly at odds with the real world, where entities in motion tend to have a great deal of autonomy. Low technology material handling systems rely on autonomous humans, while the highest technology systems rely on autonomous vehicles. Less numerous are systems that are fully automated but not autonomous.

The spatial template allows autonomous entities to evaluate their environment, both during initial planning and while in motion. The object oriented approach allows modelers to incorporate autonomy into the representation of entities in a system. However, the capabilities of object oriented modeling are limited when used in traditional simulation. We instead combine the object oriented approach with spatial representation to enable autonomous objects to set their own goals and act upon these goals, just as in the real world.

Returning to the example of the two stockpickers in the aisle (Figure 3 and Table 1), consider cases 3, 4 and 5. In each case, the spatial template warns each of the stockpickers that a potential conflict exists with the other stockpicker. The two operators must, autonomously, devise a solution. The various possible solutions together have different effects on overall system performance that traditional simulation cannot duplicate by using a delay. Certainly, one solution is for the later-arriving unit to wait until the earlier one finishes its task. This is the solution simulated by the traditional simulation's delay. Another solution, however, is for the earlier-arriving unit to pause for the later-arriving one. This is likely if the later-arriving unit is on a high priority mission or is closer to its shelf than the earlier-arriving one is to its shelf. Personal habits and preferences of the operators may also result in such a situation.

Yet another possible solution is for the stockpicker that yields to go elsewhere in pursuit of another part of its mission, rather than wait. Most warehouse operations allow operators the freedom to reorder their "pick list." In this case, the yielding unit can still spend all of its time productively filling its order, albeit on a non-optimal path. The solution that may be most efficient of all is for one unit to do the work of two. If the items being processed are not too large or cumbersome, the operators may communicate and agree that one operator, while processing his own order, will also fetch or stow an item for the other operator. In this case, one of the units is delayed but requires no time (or minimal time) to execute that part of its order.

Implementing these solutions in traditional simulation is not impossible, but is very impractical as shown in Table 2. Traditional simulation defaults to Resolution 1. Changing that default to Resolution 2 is feasible, but requires additional logic to be programmed into the model. Resolutions 3 and 4, while common in the real world, require much additional programming with even the replanned goals programmed in advance. Resolutions 5 and 6 are feasible in traditional simulation but require additional programming. In an object oriented simulation, all of these resolutions are relatively easy to implement, provided that spatial representation and goal planning capabilities are included in the model. Implementing autonomy and spatial representation provides the user with an explanation of the reasons behind delays and collisions, and what actions will eliminate the sources of these incidents.

Table 2: Traditional Simulation cannot easily represent many conflict resolutions

Conflict Resolution	Description	Traditional Simulation
1	A executes, B delays	Delay
2	B executes, A delays	Delay with logic
3	A executes, B replans goals	Extensive logic & preprogrammed goals required
4	B executes, A replans goals	Extensive logic & preprogrammed goals required
5	A executes for self and B	Additional logic required
6	B executes for self and A	Additional logic required

4 CONCLUSION

The addition of space and autonomy to discrete event simulation can provide a powerful tool for understanding today's real world systems. Without such concepts, spatial conflicts are very difficult to accurately predict and resolve. Using the spatial template can provide a suitable spatial representation, and the object oriented approach can enable representation of autonomy. In the realm of warehouses and material handling systems, effective implementation of this approach will provide a better understanding of real world system performance and aberrations from steady state.

REFERENCES

Okashah, L. and R. Rogers. 1994. Autonomy in Simulation of Material Handling Systems. *The 1994 Annual International Conference on Industry, Engineering, and Management Systems*, Cocoa Beach, Florida. Department of Industrial Engineering, University of Central Florida, Orlando, Florida.

Rogers, R. 1993. Autonomy: Simulation's Next Events. In *Proceedings of the 1993 Winter Simulation Conference*, ed. G. Evans, M. Mollaghasemi, E. C. Russell, and W. E. Biles, 1378-1379. Institute of Electrical and Electronics Engineers, Los Angeles, California.

Rothenberg, J., S. Narain, R. Steeb, C. Heflry, N. Z. Shapiro. 1989. Knowledge-Based Simulation: An Interim Report. The RAND Corporation Santa Monica, California.

AUTHOR BIOGRAPHIES

LOBNA A. OKASHAH is pursuing her Ph.D. in Industrial Engineering at the University of Central Florida. Her research interests lie in modeling and simulation, artificial intelligence, information systems, manufacturing systems and engineering economics. She holds an M.S. in industrial engineering and a B.S. in chemical engineering. She is a member of IEEE, ORSA/TIMS, SME, IIE, and SCS.

RALPH V. ROGERS is an Associate Professor in the Industrial Engineering and Management Systems Department of the University of Central Florida. His research interests include systems engineering and analysis, computer simulation, artificial intelligence, information systems, production scheduling and control systems. He is a member of IIE, IEEE, ORSA/TIMS, and SCS. He holds a B.S. in electrical engineering and M.S. and Ph.D. in industrial engineering.