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

AUTONOMY: SIMULATION'S NEXT EVENTS

Ralph V. Rogers
Department of Industrial Engineering and Management Systems
University of Central Florida
P.O. Box 25000
Orlando, FL
32816-4450
(407) 823-5912 email: rrogers@ucf1vm.cc.ucf.edu fax: (407)823-3413

KEY WORD: *modeling methodology, autonomous objects, autonomy.*

ABSTRACT

Simulation is facing the challenge to support ever increasingly complex phenomena characterized by autonomous behavior. This paper discusses an approach with which to investigate the wider modeling and analysis issues raised by these demand.

1 INTRODUCTION

Competition for resources by autonomous phenomena in complex, dynamic environments characterizes many of the problems confronting today's system analysts and system designers. In those environments where the resource conflicts are spatially based, problem concerns center on evaluating strategies for achieving the system's objective while avoiding or resolving spatial conflicts. Many systems concerned with spatial competition face the potential of high capital and human costs when conflict strategies fail and collisions between phenomena occur. Consequently, experimentation and analysis with such autonomous phenomena have relied heavily on systems modeling techniques, particularly simulation. Unfortunately, traditional simulation methodologies are reaching the limits of their abilities to deal with the complex of systems phenomena and behavior of interest to the analyst. This paper addresses the need extend these limits and provides a mechanisms for analysis of those important issues associated with autonomous phenomena in complex, spatially based system.

2 AUTONOMY

Autonomous phenomena possess full independence of movement and decision making. Autonomous phenomena may alter their temporal and spatial goal trajectories conditional upon *their* evaluation of their own current state and the state of the rest of the system *as they know it*. In simulation modeling, the functional primitive construct is missing from most discrete-even simulations. The reasons for this absence of autonomy is due to both the underlying network modeling paradigms and the related issues associated with implementing simulation models via computer software.

Typical implementations of simulation modeling use the asynchronous (i.e. event-driven) discrete-event simulation strategy. In the asynchronous simulation implementation strategy, the next scheduled event (i.e. state change) defines the next increment of time that advances the simulation clock (i.e. simulation time). Events may occur at anytime. Thus, phenomena may have their states updated at different times and the resulting increments between time advances may vary widely through the course of a simulation exercise. However, a common reference is still required to identify and resolve resource conflicts and other interactions between phenomena. Such coordination of phenomena interactions and dependencies in an asynchronous based model system requires specific operation decision points.

The network modeling paradigm readily correlates with this implementation strategy. Nodes represent decision points and arcs represent specific distances and/or times. Conflicts are resolved at the nodes. Thus, the frame of reference for asynchronous modeling is the fixed network or similar simulation construct. The specific operational decision points in this frame of reference are network nodes or similar simulation constructs corresponding to fixed points in model space.

To escape the limitations of fixed spatial increments associated with asynchronous simulation and in attempts to obtain more autonomy, discrete-event simulationists have often tried the synchronous (i.e.

time-driven) simulation implementation strategy. In the typical synchronous simulation implementation strategy, time is advanced in fixed increments. At every advance of the clock, the state of the individual phenomena must be updated. Conflicts and resolutions must be identified and actions implemented. Conditional operational decisions are made (and synchronized) at these fixed time intervals.

The reference for synchronization of action is the common time defined by the time increment. Synchronous modeling is advantageous approach when it is desired for a certain event to occur when a particular condition is satisfied (or identified). However, synchronous simulation necessitates the evaluation of each phenomena's relationship to every other phenomena at every time advance of the simulation clock. The resultant computational and modeling complexity severely restricts the simulation modeling domains where synchronous simulation may be use efficiently and effectively.

Thus, while conceptually desirable, implementing autonomy in system models compatible with traditional discrete-event simulation environments has been difficult to achieve. What is desired is allow each autonomous phenomena to schedule its next event anywhere in the model space its operational rules permit, determine if there is a conflict with any other phenomena in model in achieving that next event, and effectively recognize and resolve conflicts when necessary. Clearly, how model phenomena efficiently and effectively recognize and resolve conflicts are the basic issue with incorporating autonomy into discrete-event simulation.

3 AN APPROACH FOR AUTONOMY

The conflict identification strategy pursued uses the concept of a spatial template. Under this concept, all spatially based phenomena in the model space are represented as geometric shapes. To simplify, all object model shapes are polygons. Dynamic phenomena (e.g. airplanes, ships, guided vehicles, weather, etc.) are also represented by polygons but polygon size and shape is based on the phenomena's model space trajectory. The parameters of the associated polygon of the dynamic phenomena in the spatial template are defined by the originating phenomena event and the next scheduled event for that phenomena as well as unique characteristics of the phenomena.

For example, the trajectories of two moving entities could be represented as two polygons in the X-Y plane of their movement. The origins and destinations of each entity would define two points in the X-Y plane. These two points would correspond to events for each entity. The associated polygons on the spatial template are defined by the origin and destination points of each entity and a Δd which defines parallel lines some Δd distance from the line between the origins and destination. Perpendicular lines through the origin and destination complete the definition of the trajectory polygon.

The static representation in two dimensions implies a third dimension (time) by the extent of the polygon from the point of the arrival event. Uncertainty in the proposed model space trajectory may be reflected in the shape and extent of the entity polygon (i.e.Δd's). *Potential* conflicts are identified when the representational polygon of objects intersect. Once intersection is detected, objects may then resolve conflicts. The key issues associated with this approach is how to *efficiently* identify polygon intersections in the spatial template.

How to identify these potential conflicts efficiently is dependent on how to represent the model space information and the entity polygon trajectory information. The spatial template does this by partitioning the model space into equal sectors. The sector partition then overlays the model space Cartesian system and the entity trajectory polygons. The sectors are identified by their coordinate sector numbers. The sectors through which the trajectory polygon overlays and/or intersect are identified. Associated with each sector is a list of entities whose trajectory is scheduled to go through some part or all of the sector. When entity object trajectories are added to the sector list, a check is made to determine if any other entity trajectory are also associated with a sector. If an entity trajectory is already associated with a sector, then a *potential* conflict exists. When conflicts are identified, the model communicates with each entity involved advising each entity of the identity of the other entities involved. Entities then communicate to establish if a true conflict exists. When conflicts do occur, the entities employ their conflict resolution strategies in a attempt to resolve the conflict.