Proceedings of the 1987 Winter Simulation Conference
A. Thesen, H. Grant, W. David Kelton (eds.)

MODELING ASYNCHRONOUS MATERIALS HANDLING IN XCELL+

Richard Conway
William Maxwell
Cornell University
Ithaca, NY 14853, U.S.A.

ABSTRACT

The XCELL+ Factory Modeling System has recently been extended with facilities to model automatic guided vehicle systems and power-and-free conveyor systems. This significantly expands the scope of problems that are susceptible to modeling in a graphical, menu-driven mode by non-simulation professionals. This paper summarizes the issues and solutions represented by these extensions to XCELL+.

1. INTRODUCTION

At WSC 86 we reported on the recent and rapid development of modeling tools for non-simulation-professionals and, in particular, described the XCELL modeling system which we have developed. We noted that we were then working on extensions with which to model certain types of materials handling facilities. The current paper is a summary of the result of that work, as implemented in the XCELL+ system.

As described at WSC 86, there are currently two different development efforts in the field of simulation tools for manufacturing. The first is extension of the power of the general-purpose modeling systems to increase the effectiveness of simulation professionals. The second is the development of tools that are appropriate for use by people concerned with the problem, without forcing them to become simulation professionals. The issue with respect to the latter is no longer one of feasibility, but rather one of scope of the problems that are susceptible to this approach.

We have been single-mindedly concerned with the second task -- development of tools for nonprofessionals -- and have steadfastly resisted the temptation to "enrich" XCELL to the detriment of its intended user population. At the same time, we recognized the existence of an important class of manufacturing problems that could not be effectively modeled with the facilities in XCELL. These are problems with significant dependence on the performance of asynchronous materials handling facilities, such as automatic guided vehicle systems (AGVS), and power-and-free conveyor systems. Such problems could, of course, be modeled using the powerful general-purpose tools such as SLAM II or SIMSCRIPT, but it was not clear whether they were susceptible to tools such as XCELL or WITNESS. We have spent almost two years learning about asynchronous handling systems (with generous help from engineers at SI Handling, Raymond, Manessman Demag and Eaton Kenway) and trying to find a simple underlying paradigm that would allow modeling such systems in XCELL-like terms. The results are implemented in the successor to XCELL, called XCELL+, and are described below. As this is written, this version of XCELL+ is just going into productive service, so it remains for the jury of users to decide how well we have succeeded.

2. THE BASIC FACILITIES OF XCELL+

Since an understanding of the general character of XCELL+ is prerequisite to understanding the materials handling extensions, let us briefly summarize the host system.

XCELL+ is an interactive, menu-driven, graphical system in which a model is constructed by placing and connecting elements selected from a very small set of element-types. The construction as well as the execution of the model is inherently graphical. The system is highly interactive, permitting the user to observe the model as it is running (rather than after the run has actually been completed), and to interact with its execution in a variety of ways. The user can even interrupt execution, alter the structure of the model, and then resume execution (without necessarily restarting).

The element-types, from which the user selects, are only the following:

Receiving Area -- receives material (from outside the model) and releases it, one "unit" at a time, to the model

Shipping Area -- receives material, one "unit" at a time, from the model and "ships" it to the outside world

Buffer -- stores material, within the model, that is not currently being processed

Process -- an activity performed on a certain type of material, requiring varying amounts of time, with varying degrees of success

an **Assembly Process** can merge two types of material

Workcenter -- the primary facility required to perform a Process -- each Process is assigned to a particular Workcenter; each Workcenter can have any number of Processes assigned, but only one can be active at any instant

Workcenters are (optionally) susceptible to either scheduled maintenance or random breakdowns

Auxiliary Resources -- the home site for one or more secondary Resources of a particular type used to perform particular Processes (used primarily to represent "operators")

Maintenance Facilities -- the home site for one or more "maintenance teams" of a particular type used to perform maintenance upon a particular set of Workcenters

Links -- a route for material movement from a Receiving Area, Buffer or Workcenter to a Buffer, Workcenter or Shipping Area.

XCELL+ is intended to model stochastic, multi-product, flow systems. In such systems there are a modest number of recurring products, each with a pre-determined (although possibly varying) routing through certain Processes. (XCELL+ is much less convenient for modeling a "job shop" in which each job has a unique and original routing.) "Stochastic" means that various characteristics (processing time, yield, breakdown, repair, supply, demand) can be specified in terms of probabilities rather than fixed values.

Arbitrarily complicated flow paths can be specified, and XCELL+ is capable of modeling demand-driven ("push") systems as well a supply-driven ("push") systems.

3. THE MATERIALS HANDLING EXTENSIONS

At the element level the only materials handling facilities in XCELL+ (prior to the extensions described below) were Links. Links can be viewed as idealized conveyors:

-- material movement is infinitely fast (consumes no model time)

-- there is no inherent storage of material on a Link

-- a Link can go from any element to any other, regardless of their locations on the factory floor, or the location of any other element

Graphically, a Link is represented by a straight line from its source element to its destination element that overlays any possible element between the two. Consequently, a model can be arbitrarily rearranged by moving any of its elements without having any effect on the behavior of the model.

With only Links available to move material, it would be fair to characterize XCELL+ as favoring the construction of models in which processing considerations dominate those of materials handling. To be sure, the user can construct composite elements from the built-in primitive elements to model real materials handling devices more faithfully, but this requires both ingenuity and work. For example, a less-than-infinitely-fast conveyor with more-than-zero storage capacity can be modeled by using a Workcenter-Buffer pair, and since the display of both these elements can be suppressed, the model can show an apparently normal link with appropriate values of transfer time and storage capacity. In fact, an entire section of the XCELL+ User's Guide is devoted to such techniques. However, there is still some degree of approximation to a real conveyor's behavior, and there is no prospect whatever for modeling an AGV system or an overhead crane by such tricks.

Therefore, if the advantages of XCELL-style modeling were to be applied to problems in which materials handling considerations were significant, or dominant, then some new set of primitive elements had to be provided.

As was the case in the original design of XCELL, it was easy to get the rough idea right -- and then slow and painful to get the details right. We repeated our previous development process and built a series of prototype systems on HP's engineering workstations (now called HP 200's and 300's), using their elegant implementation of an extended Basic ("HP Rocky Mountain Basic") and experimented with many variations before casting the final choices in the concrete of C code for the PC/Vectra family of machines. The double implementation probably takes a little longer, but we believe it results in a much more refined and reliable program product.

The basic idea was obviously to construct a network of Paths over which Carriers can transport material. The difficulties lay in:

a. deciding how to relate distance and travel time to the graphical representation of the network (recall that up to this point, only the logical relationship between elements was significant in the operation of the model; the geometric relationship was strictly for the user's visualization)

b. devising tools with which the user could exercise adequate control over the movement of Carriers (without departing from the menu-driven paradigm of XCELL+)

c. figuring out how best to "connect" the new facilities with the existing XCELL+ elements

There were a bewildering number of possible design choices, and a constant battle between simplicity and flexibility, but we are in general very pleased with the end result.

The extensions involve only two new element-types (or perhaps three, depending on your point-of-view):

Path -- a connected sequence of (orthogonally adjacent) cells over which a Carrier can move; a Path can be directed or dynamically reversible

Control Point -- a cell that serves as the end-point of one or more (up to four) Paths; every path originates and terminates in a Control Point; also the points of connection with other XCELL+ elements

The third element-type is:

Carrier -- the vehicle that travels over a network of Paths, sometimes empty, sometimes carrying material

The carrier is probably best viewed as a new element-type, but it is different in two important respects from all others:

-- it moves from one cell to another (all other types are stationary)

-- it never has a cell of its own, and is always co-resident with a Path cell or Control Point

Actually, the presence of a Carrier is simply an attribute of a Path cell or a Control Point.

The connection between the Path network and other elements of the model is provided by ordinary Links to Control Points. Each Control Point can (optionally) have a "pickup Link" and/or a "dropoff Link." Pickup Links come from the elements that normally supply material: Receiving Areas, Buffers, or Workcenters. Dropoff Links go to elements that normally accept material: Buffers, Workcenters, or Shipping Areas.

Pickup and dropoff Links are ordinary Links (with infinite speed and no storage) so the geometric proximity of a Control Point to the element it serves is immaterial -- it has no effect on the operation of the model. However, within the Path network the geometry is very important since it determines the Carrier capacity of the network. That is, each cell of the network -- Path cell or Control Point -- can hold only a single Carrier, so the length of a particular Path, in terms of the number of cells in XCELL+'s rectangular grid, determines the number of Carriers that Path can accommodate.

However, geometric distance in the network does not represent travel time for the Carriers. Each cell in the network can have a transit time specified that is independent of the time of all other cells -- although the default transit time is, in fact, proportional to distance. (A zero value of transit time is allowed, and this is often useful.) In addition to their transit time, Control Points also have a "pickup time" and a "dropoff time" specifying how long a Carrier at that Control Point is delayed by the act of pickup or dropoff. All of these times are specified as constants -- we were advised that the speed of real Carriers is typically relatively uniform so no provision for probabilistic variability in speed was necessary.

There is only provision for one type of Carrier -- that is, there are no attributes assignable to individual Carriers. Speed, for example, is specified by assigning transit times to Path cells so that any Carrier reaching that cell takes the same time to traverse it. There is provision to have empty Carriers and full Carriers move at different speeds, but the difference is uniform for all Carriers. Also, Carriers all cost the same amount in the built-in cost analysis feature of XCELL+.

At least in the current version of XCELL+ there is no provision for breakdown of Carriers, or for failure of any Path cell. It may turn out to be useful to extend to Control Points the maintenance features that are now applicable only to Workcenters. There is also currently no explicit provision with which to periodically take Carriers out of service for "re-charging," although a reasonable approximation can be constructed using the elements provided.

4. THE CONTROL OF CARRIER ACTION

The complicated aspect of an asynchronous-vehicle materials handling system is, of course, control of the action of the vehicles. There are several different classes of systems, differentiated by the location of the control mechanism -- e.g. "smart car" systems and "smart track" systems. We sought for XCELL+ a single set of facilities flexible enough to approximate any reasonable control strategy. The principal issues were the following:

- -- When a Carrier leaves a Control Point that has more than one outgoing Path, which Path should it enter?
- -- When an empty Carrier arrives at a Control Point with a pickup Link, should a pickup be made? When a loaded Carrier arrives at a Control point with a dropoff Link, should a dropoff take place?
- -- Where should empty Carriers be "parked" to be pre-positioned for future use?
- -- How to allow Carrier action to be "pulled" by the material needs of other elements, rather than be "pushed" by the availability of loads?
- -- How should the current direction of dynamically-reversible Paths be determined?
- -- How should Carrier action be affected or dynamically controlled by traffic congestion in the network.

The XCELL+ "solution" to these issues involves three interacting components:

1. Routed Carriers -- a particular Carrier can be assigned the immediate task of traveling to a particular Control Point
2. Dispatch rules -- each Control Point has a set of dispatch rules for empty Carriers, and another set for loaded Carriers
3. Material delivery requests -- each Buffer in the model has the option of issuing individual requests for material delivery.

These facilities are described in detail in the following paragraphs.

The normal (default) mode assumes smart Control Points, subservient Carriers, and dumb Path cells. That is, each time a Carrier arrives at a Control Point it follows instructions associated with that particular Control Point. Path cells have no provision for instructing Carriers.

However, one of the instructions that a Control Point can give a Carrier is to "travel to Control Point N." (Each Control Point has a unique identification number.) The Carrier is then considered "routed," with Control Point N as its destination. The routed Carrier proceeds to N, following the fastest possible route, and ignoring instructions of Control Points it passes through en route. ("Fastest route," in this context, depends upon the static characteristics of the network, and not on current traffic conditions.) Once the Carrier reaches Control Point N it becomes "unrouted" and is again susceptible to local instructions. Either a loaded or an empty Carrier can be routed in this way.

The default dispatching rules at a control Point are the following:

D1. An empty Carrier will make a pickup if there is a pickup Link and if a unit is currently available at the source end of that Link.

D2. A loaded Carrier will make a drop off if there is a dropoff Link and if the element at the destination end of that Link can currently accept a unit.

D3. A Carrier having completed its transit of the Control Point (after dropoff and/or pickup, if applicable) will exit by the next outgoing Path (in clockwise rotation) relative to the exit Path of the previous (unrouted) Carrier at this Control Point.

The optional rules that can be specified for loaded Carriers are the following. Note that each Control Point has its own set of rules -- independent of those of any other Control Point.

L1. Carriers must WAIT until a dropoff can be made (i.e. -- until the element at the destination end of the dropoff link can accept a unit.

L2. A Carrier having completed transit will become routed, receiving as its destination the next Control Point on an arbitrary user-specified sequence of destinations.

L3. A Carrier having completed transit will become routed, receiving as its destination whichever Control Point in an arbitrary user-specified set of destinations has the fewest loaded Carriers currently headed toward it.

L4. A Carrier having completed transit will become routed, receiving as its destination a particular Control Point specified by the user as the destination for the particular type of load currently on the Carrier.

The optional rules that can be specified for empty Carriers are the following:

E1. Carriers must WAIT until a pickup can be made (i.e. until a unit is available at the source end of the pickup Link).

E2. An integer K can be specified such that K empty Carriers will be passed between pickups, regardless of the availability of units on the pickup Link.

E3. Carriers can be HELD until a specific material request is received.

E4. A Carrier having completed transit will become routed, receiving as its destination the next Control Point on an arbitrary user-specified sequence of designations. (This is a separate sequence from the loaded Carrier destination sequence.

Alternatively, any Control Point can be put in "manual" mode so that, during running of the model, each time a Carrier enters that Control Point, the model pauses for a decision to be made by the user.

A "material request" is a specific request for "one unit of Part P to be picked up at Control Point S and routed to Control Point D." Each Buffer in the model has the option of issuing such requests, depending upon its own stock level (much like the "Process triggers" in XCELL+). Each Control Point S, that is the source of material for such requests, maintains a queue of current requests until they have been satisfied (more precisely, until a Carrier has been loaded with P at S and routed to D). If one or more Control Points have been designated as "empty Carrier HOLDING Points," then each time a material request is issued, a corresponding "empty Carrier release" is issued, routing an empty Carrier to S. (If there is more than one HOLDING Point in the model, each Control Point S is automatically assigned to the closest HOLDING point).

Reversible Paths in XCELL+ follow the normal "one-way bridge algorithm":

a. If the bridge is currently empty, traffic can enter from either end.

b. If the bridge currently contains at least one vehicle, traffic can enter only in the same direction as current traffic. Opposing traffic must wait for the bridge to become empty.

This works very effectively in most low-Carrier-density situations in XCELL+. However, since there is no built-in "passing lane" in a Control Point, problems can arise as the Carrier density increases. For example, if a Carrier arrives at a Control Point with only a single outgoing (reversible) Path -- and there is currently traffic incoming on that Path -- the control Point would be deadlocked. XCELL+ counters this with a "deadlock avoidance algorithm" that seeks second-fastest-paths when the fastest Path is blocked and, in extreme cases, forces reversal of reversible Paths currently bearing traffic (i.e. forces vehicles to back off the bridge when there is no other way to avoid deadlock at the end of the bridge). Unfortunately, this algorithm is not omnipotent and a user can still all too easily construct reversible Path networks that have the potential for deadlock. Our only consolation is that apparently it is equally easy to do so in the real world.

5. CONCLUSIONS

If this sounds complicated, it is, although it seems somewhat easier to use than to describe. It is simply the case that manufacturing systems with asynchronous materials handling components are inherently complicated, and tools with which to model them must necessarily reflect that complexity.

We believe that, with these extensions, we have succeeded in significantly expanding the scope of problems that can be modeled with XCELL+ without making the tool itself harder to use. However, we have not been able to insulate the user from the fact that typical systems in this expanded domain are much more complex than the typical flow capacity problems that were XCELL+'s previous targets. Perhaps the difference can be emphasized in the following way. Prior to these extensions, if you neglect issues of variability and reliability, system capacities could often be estimated by calculation without ever running the model -- in effect, XCELL+'s purpose was often just to measure the impact of variability. In contrast, even without any probabilistic elements, it is nearly impossible to accurately estimate the throughput of even a relatively simple model with asynchronous materials handling features without execution of the model.

We have been astounded at the major differences in model throughput that can result from apparently minor changes in the dispatching rules at key Control Points, or minor changes in the layout of the Paths. Presumably, experienced designers of such systems would be less surprised and produce more effective initial designs, but some have admitted (in private) that there is still considerable artistry (if not witchcraft) in the design of asynchronous vehicle systems. This is, of course, a situation made to order for computer modeling, and the industry is already heavily dependent on conventional modeling tools. All that is really at issue is whether useful models of such systems can be constructed with tools like XCELL+.

It is a challenging proposition, with potentially great benefit in the improvement of contemporary manufacturing systems. We offer this description of our best efforts (so far) as a starting point, or a target, for other modeling tool developers.

REFERENCES

XCELL and XCELL+ are distributed by Pritsker and Associates, P.O. Box 2413 West Lafayette, IN 47906, U.S.A. 317-463-5557

Conway, R., and Maxwell, W., (1986). XCELL: A Cellular, Graphical Factory Modeling System. In: Proceedings of the 1986 Winter Simulation Conference

Conway, R., Maxwell, W., McClain, John O. and Worona, Steven (1987). User's Guide to the XCELL+ Factory Modeling System, Second Edition. Scientific Press, Redwood City, CA

AUTHORS' BIOGRAPHIES

RICHARD CONWAY is a professor in the Johnson Graduate School of Management, and WILLIAM MAXWELL is the Andrew Schultz Professor in the School of Operations Research and Industrial Engineering, both at Cornell Universty. Both received their PhD in operations research from Cornell.

Their collaboration in simulation and research in manufacturing spans a thirty year period. Both were at the RAND Corporation in the early 1960's, working with Dr. Harry Markowitz in the development of the SIMSCRIPT language. Their work in industrial scheduling ("Theory of Scheduling," Addison Wesley 1967) has long been the standard reference for the topic.

They, with Steven Worona, are the developers of the XCELL series of modeling systems, in collaboration with the Manufacturing Research Center of Hewlett Packard Laboratories, SI Handling Systems, and Pritsker & Associates.

Richard Conway
Johnson Graduate School of Management
Cornell University
Ithaca, NY 14853
607-255-7207

William Maxwell
School of Operations Research
and Industrial Engineering
Cornell Unyersity
Ithaca, NY 14853
607-255-9147