Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

HYPER-LINKED APPLICATIONS AND ITS EVOLVING NEEDS FOR FUTURE SIMULATION CAPABILITIES

Wayne J. Davis
Fernando G. Gonzalez
Joseph G. Macro

Department of General Engineering
University of Illinois
Urbana, Illinois 61801, U.S.A.

Andrew C. Sieveking

Proctor and Gamble, Inc.
Cincinnati, Ohio 45202, U.S.A.

Scott V. Vifquain

Kurt Salamon Associates
Atlanta, Georgia 30309, U.S.A.

ABSTRACT

A new object-oriented framework, termed "hyper-linked manufacturing," is introduced for computer-integrated manufacturing. This framework permits complex organizational structures among intelligent agents to address the distributed coordination of the enterprise. Each intelligent agent has an integrated planning and control capability and has the capacity to model the interaction of itself with its subordinates. To this end, a distributed simulation becomes a major functional requirement in assessing the performance of subsystems within the enterprise while operating under a planned course of action. This paper defines the basic concepts associated with hyper-linked manufacturing and then discusses the future requirements for simulation technologies that evolve from its implementation.

1 INTRODUCTION

Today, engineers and managers are attempting to design and coordinate the operation of extremely complex, large-scale systems. Since these systems are typically discrete-event in nature, conventional planning and control algorithms are of limited utility. The application of current planning algorithms necessitates so many simplifying assumptions that realism is sacrificed while most control algorithms address continuous-state systems only.

The complexity of these systems also demands that planning and control be distributed. Often, planning horizons ranging from years to minutes must be considered for the same system. Real-time management necessitates that planning be situated with the same agent that is responsible for implementing the plan. To this end, a new generation of intelligent controllers or agents is being developed.

We must also develop the essential algorithms which will permit these objects to interact in a coordinated fashion. Unfortunately, the existing algorithms for the decomposition of planning and the decentralization of control are simply inadequate for these complex systems.

Another critical element in this analysis is our ability to predict the response of a subsystem within the overall system while it operates under a planned control strategy. We now know that current simulation tools are also deficient in this respect. Mize et al. (1992) raises concerns pertaining to the ability of current simulation to adequately predict the response of a flexible manufacturing system (FMS). Davis et al. (1993, 1995a) further demonstrates that an integrated approach to the modeling, scheduling and control of these systems was needed and set out to define a new object-oriented modeling approach to address these concerns.

In generalizing upon this integrated solution for the modeling and management of FMSs, we have arrived at a framework for the decentralized coordination of the entire manufacturing enterprise, termed "hyper-linked manufacturing" (HLM). HLM employs an object-oriented formulation with intelligent agents interacting to coordinate the enterprise. It further assumes that each agent has the capability to construct and validate a model of its interactions with its subordinate subsystems.

The fact that HLM presumes that a given agent can have subordinate subsystems necessarily implies that agents can be integrated into planning and control hierarchies for the distributed management of the overall system. In fact, we have shown that the resulting hierarchical assembly of objects is wholly consistent with current planning decomposition and decentralized control techniques.

HLM, however, significantly expands the traditional hierarchical modeling approaches. It permits any agent to include any other agent in its planning. Hence, planning subhierarchies may be defined for a given agent which are not contained within the basic control hierarchy. Furthermore, the control function has been decomposed, permitting several distinct control structures to exist simultaneously. HLM's goal is to provide the flexible framework required to model large-scale systems.

A discussion of the modeling capabilities which can be addressed with hyper-linked frameworks is well beyond the scope of this paper. (The reader is referred to Davis et al. (1995b) for an expanded discussion of the hyper-linked architecture.) Rather, this paper provides only the most basic concepts of HLM and then attempts to assert the essential simulation capabilities that must exist for future implementations of the HLM in real-world settings. Our presentation continues with a fundamental definition of the manufacturing problem which HLM seeks to address.

2 THE ENTERPRISE PROBLEM

In most large-scale systems, including manufacturing systems, the managed allocation of available resources is critical. These resources can be divided into two groups: task-capable and passive resources. *Task-capable resources* are essential to the operation of the large-scale system, and these typically reside at the lowest hierarchical levels of the control structure. In fact, coordination architectures for a large-scale system have largely directed toward the definition and the assignment of tasks that these task-capable resources execute. If there are continuous-state subsystems within these large-scale systems, these are almost certainly task-capable resources whose continuous-state response is punctuated by the start and finish events associated with each executed task.

Passive resources, on the other hand, are generated or consumed in the execution of tasks. It follows that when a task-capable resource executes a task, it will change the state of one or more passive resources. The state of the passive resource can be modified in at least two ways: either the physical state (properties) or the physical location of the passive resource.

In view of this, we further divide the task-capable resources into two classes: the unit process and the transport resource. The *unit process* physically modifies the state or properties of one or more passive resources when it executes a processing task. The *transport resource*, on the other hand, changes the location of a passive resource.

Processing instructions provide a "recipe" for the execution of tasks. Since much of the operation of a large-scale, discrete-event system can be classified as the execution of tasks, the majority of the system's knowledge base is defined within the database of processing instructions.

The set of available processing plans is not static. Indeed, one role for a large-scale system is to develop additional processing plans. The ability to generate new process plans permits large-scale systems to adapt to their environments over time. Since the process plans represent an important element of the system's knowledge base, this form of adaptation may be viewed as system-wide learning.

We may further assume that the set of unit processes is not static. Hence, as the system evolves in time as new items and the unit processes which produces them are developed. The overall enterprise problem is thus one of managing currently available resources and assembling new resources to manufacture items in a manner which both maximizes profitability and insures long-term security. The HLM approach seeks to define a distributed object-oriented framework through which the enterprise problem can be addressed using the principles of decentralized control and planning.

3 THE COORDINATION HIERARCHY

In this section, we will specify a coordination hierarchy which provides the foundation for HLM. Section 3.1 will introduce the concept of the coordinated object which we will employ in our decomposition of the enterprise problem. Section 3.2 will then discuss the Recursive Object-Oriented Coordination Hierarchy (ROOCH) which results from this decomposition. Finally, Section 3.3 introduces the concept of the Hierarchical Subsystem Coordinator which is assumed to be an element of each coordinated object in the ROOCH. The Hierarchical Subsystem Coordinator represents our generic framework for an intelligent controller or agent.

3.1 The Coordinated Object

The generalized template in our proposed decomposition is the coordinated object (CO), depicted in Figure 1. It represents the most fundamental hierarchical element within which integrated planning and control are implemented. Each CO contains one or more subordinate systems, P_n (n=1,...,N), which are to be employed in the execution of assigned tasks. To execute these tasks, passive resources enter the CO through its input port and eventually exit through the output port. These passive resources are assumed to be under the control of the CO from the moment they enter the input port until they exit through the output port.

In general, the CO does not perform the tasks itself, but decomposes or disaggregates its assigned tasks into subtasks which are implemented at the subordinate systems. When the CO allocates a given passive resource to one of its subordinate systems P_n, the control of that resource is also relegated to that system. The CO must also insure that the essential resources are provided to the subordinate as dictated by the processing plan.

We also assume that each CO has the essential Interfacing Subsystems to move the resources from one subordinate subsystem to another. These Interfacing Subsystems represent the transport resources and must also be under the control of the CO. The control of the Interfacing Sub-

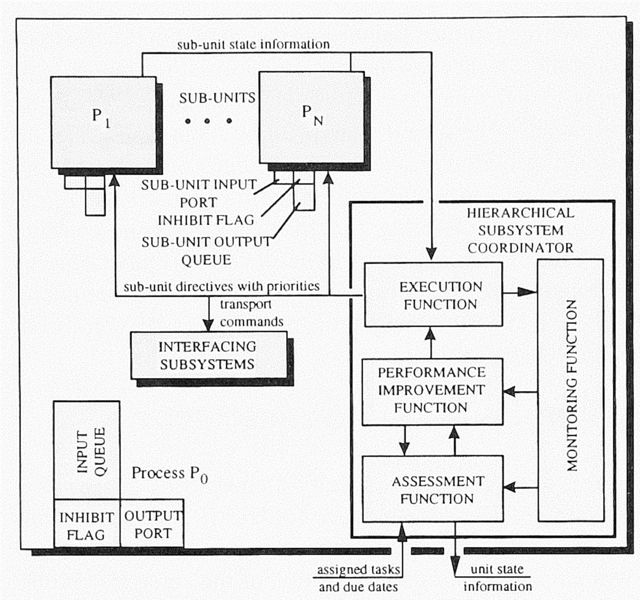

Figure 1: Schematic of the Coordinated Object: The Basic Module for Planning and Control

systems is manifested via the CO's assignment of transport tasks to be executed.

The CO has already been applied by the authors to model several discrete-event systems, particularly those in the manufacturing sector. Perhaps one of the most complex flexible manufacturing systems (FMSs) that has been built is the Rapid Access to Manufactured Parts (RAMP) FMS constructed by the US Department of Defense's ManTech program. After several prior attempts to model this FMS using conventional languages, we adopted a new approach which incorporated a decomposition using COs and modeled the information flows among the controllers which govern the flow of all entities (see Davis et al. 1994).

3.2 The Recursive Object-Oriented Coordination Hierarchy (ROOCH)

The recursive nature of the ROOCH arises from the fact that any subordinate subsystem or object within a given CO can also be represented as a CO. This recursive approach may be applied to construct the ROOCH with the essential number of hierarchical levels needed to model any subsystem. This property is illustrated in Figure 2 where each station within the cell-level CO for the RAMP FMS is also modeled as a subordinate station-level CO.

Each station-level CO also contains one or more unit processes and a material handler. At most stations, the human operator provides the primary material handling, (i.e., the interfacing subsystem), by manually carrying the totes. In many cases, the human operator also functions as the unit processor, (i.e., the execution of the processing steps requires the manual efforts of the human operator as the required processes are not mechanized). It should be noted, however, that the human operator typically functions in only one capacity at a time.

The reader has probably begun to appreciate the complexity of the large-scale system and the obvious difficulty that arises when one attempts to model its operation. Developing the ROOCH for the entire enterprise is outside the scope of this paper, but it has been addressed and is reported in Davis (1995) and Davis et al. (1995b).

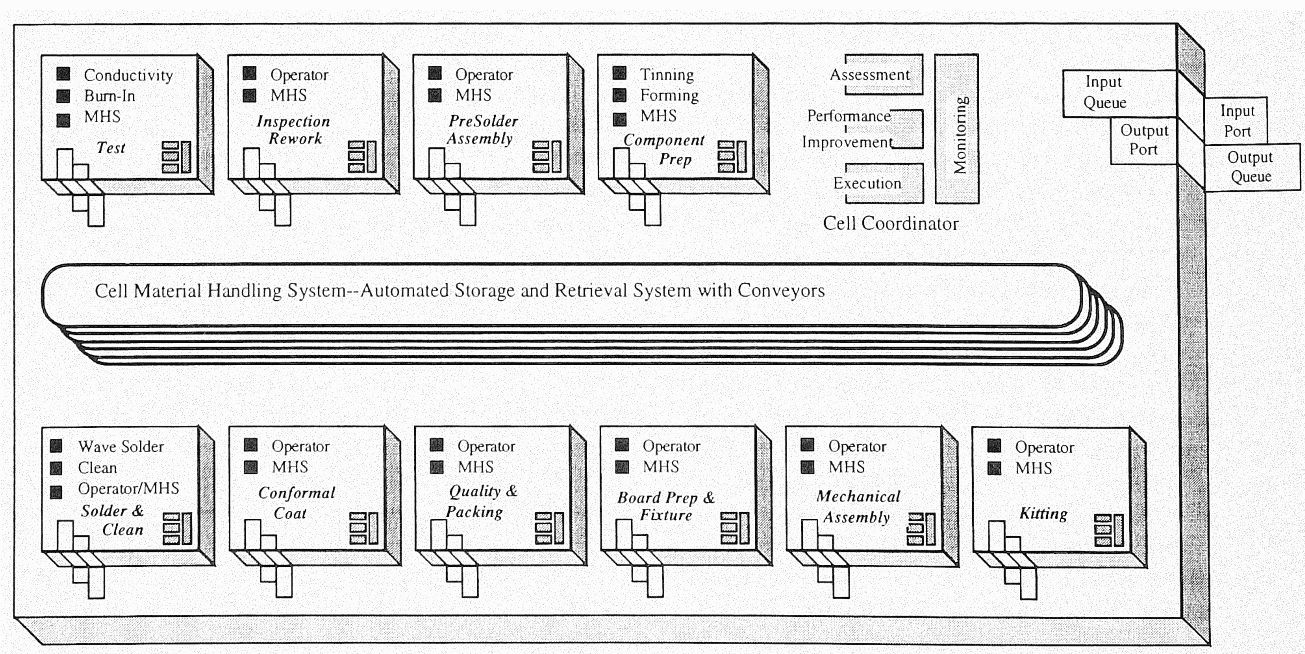

Figure 2: Cell-Level Coordinated Object for the RAMP FMS

3.3 The Hierarchical System Coordinator (HSC)

To implement the integrated planning and control within a CO, the concept of the HSC was introduced and discussed in Davis, Jones and Saleh (1992), Davis (1992) and Davis et al. (1995a). The HSC is explicitly designed to consider the planning and control issues associated with the operation of a discrete-event system in any time domain. The proposed configuration for the HSC is constantly undergoing refinement. As illustrated in Figure 3, the HSC includes four basic functions which are assumed to operate concurrently:

- The Assessment Function,
- The Performance Improvement Function
- The Execution Function, and
- The Monitoring Function.

Given that the CO is managing stochastic subsystems, often in a real-time environment, and that there are typically multiple performance criteria to be considered, it is generally impossible to assert the optimality of the CO's current plan with its enabling control law, C*. Rather C* implements the best plan known to the CO at the moment. The process of seeking an improved plan is the constant responsibility of the Performance Improvement Function.

The Execution Function implements the current plan C*. In this role, the Execution Function interacts directly with the Assessment Function within the HSC of each subordinate subsystem (CO). In a similar fashion, the Assessment Function within the CO interacts with the Execution Function of its supervisor to define new tasks for the CO and their completion dates.

Finally, a CO can never impose an unfeasible request upon a subordinate subsystem. When disruptions occur, the Monitoring Function must intervene and restore consistency among the concurrent operations of the other three functions. In general, the first attempt is to restore feasibility by modifying C*. However, if this is not possible, then the specifications for the assigned tasks must be modified.

4 SPECIFYING HYPER-LINKS

In the previous section, we defined the coordination hierarchy for the manufacturing enterprise problem. Based upon the discussion thus far, there seems to be no need to consider hyper-linked architectures as a pure hierarchies appear to suffice. In this section, we will attempt to improve the overall planning capability of the Coordination Hierarchy by allowing, (for planning purposes only), a given CO to include other objects which are not within its control domain. This capability will be termed "virtual planning." We will also demonstrate a further need to functionally decompose the overall control function into three control subfunctions: Coordination, Execution and Regulation.

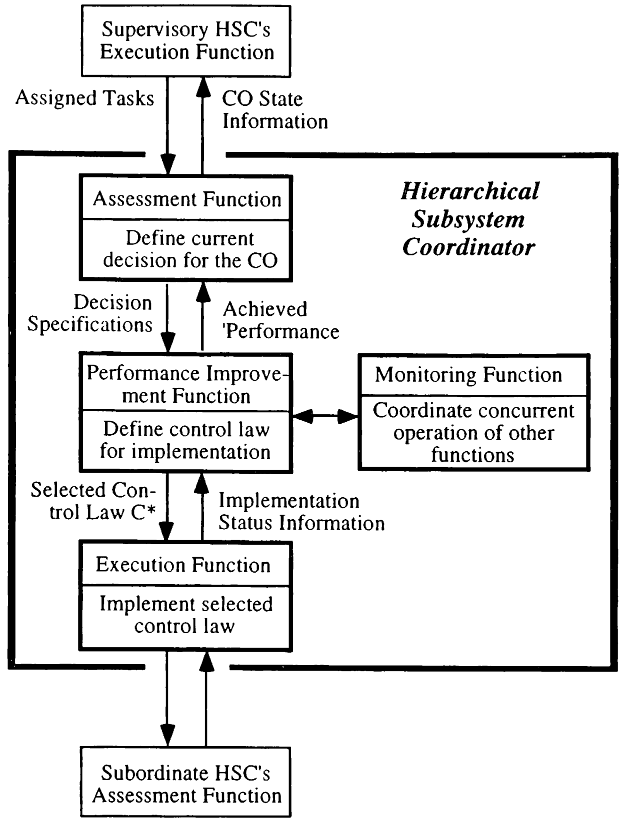

Figure 3: Schematic for the Concurrent Functions Processes Comprising the HSC

4.1 Developing Virtual Planning Hierarchies

In the entire coordination hierarchy for a large-scale system, there are several planning problems to be addressed concurrently. It is often desirable, if not essential, for given COs to cooperate in their planning without the direct supervision of their immediate supervisors. When such planning interactions are permitted, the hierarchical structure from a planning perspective is violated. The result is that each CO has a supervisor which can assign it tasks, a specified set of subordinate COs to which it can assign tasks, and yet another set of COs with which it can interact to solve its planning problems.

To provide for the maximum flexibility, we propose the hyper-linked manufacturing environment. In this environment, each CO has its own individual state description which is protected, (i.e., this state can only be changed by the CO itself). The CO changes its state by executing the tasks assigned by its supervisor. It executes these tasks either by performing predefined (or implicitly known) instructions or by assigning subtasks for its subordinate subsystems to execute.

Each CO is also responsible for validating its own model for the coordinated behavior between itself and its subordinate subsystems, or state transition function, using predefined algorithms that are contained within the object's private functions. This effort is conceptually similar to the system identification function often addressed in control engineering. Without this model of the state transition function, the CO is incapable of predicting its behavior, and thus, its planning and control capabilities are severely diminished or nonexistent.

On the other hand, this CO's model need not be proprietary. In fact, HLM assumes that each CO's model is public and can be exercised by other COs in the coordination hierarchy. Each CO would also be responsible for providing the essential state information needed to exercise (simulate) its model. When a CO employs another CO's model to perform "What If?" analysis, it has no immediate effect upon the simulated CO because the simulated CO can only change its state by executing a task as specified by its supervisor.

In our modeling of the enterprise, we have already discovered several situations where virtual planning hierarchies appear essential. For example, if an accurate model has been developed for the behavior of a given process, the process planner may immediately simulate the process's behavior when it executes a proposed set of processing instructions. If a shop scheduler desires to determine to which cell a rush job should be assigned in order minimize the disruption to the scheduled work flow, the shop CO can execute simulation models for each of the candidate cells to determine the modified work flow patterns which would ensue from the rush being placed in each cell.

The interaction among the objects need not be limited to the COs contained within the enterprise. For example, a given customer may be granted permission to access the system and be able to monitor progress in the completion of the order.

By permitting the inclusion of any other object into the planning hierarchy of another object, it is clear that HLM does provide an essential capability for virtual manufacturing. There are also efforts underway to support the definition of virtual enterprises where several corporations interact to design and manufacture a given product. HLM also provides a framework for this virtual enterprise. A discussion of this capability is outside the scope of this paper, however.

4.2 The Need to Functionally Decompose the Overall Control Function

A recognition of the need for functionally decomposing the overall control function was again derived from our consideration of a real-world manufacturing systems. When we began to investigate the coordinated operation of two or more manufacturing cells, we discovered that on more than one occasion it was essential for a given CO to receive tasks from more than one supervisor. For example, a given automated guided vehicle (AGV) system may serve two cells. In such an instance, it was desirable that both cell controllers be able to make transport requests directly to the AGV controller. However, the inclusion of this capability immediately violated the pure hierarchical structure which does not permit any subsystem to have more than one supervisor.

We then realized that two distinct modes of control were being addressed. The first mode of control was the executive mode which defined which objects could assign tasks to other objects for execution. Based upon real-world examples, it was apparent that the desired executive control structure may not be hierarchical in many cases.

The second control mode was that of coordination which established the priorities that each CO would employ in the execution of its assigned tasks. Based upon the definition of the ROOCH for the enterprise, it is apparent that the structure for coordination is hierarchical. We also observed that both control structures can exist without interfering with each other. Assuming that the two cells which share an AGV system are coordinated by a shop-level controller, then the cell controllers are subordinate to the this shop-level controller. For coordination purposes, the AGV controller is also subordinate to this shop-level controller. However, for executive control purposes, the AGV controller is subordinate to both cell controllers. Under this proposed organization, both cell controllers can make transportation requests to the AGV controller. However, only the shop-level controller can establish the priorities that the AGV controller will apply in the scheduling of its assigned transport requests.

We also discovered a third mode of control which we term "regulation." Under this mode of control, it is assumed that each object's behavior will not be influenced by the regulator as long as the object operates within a prespecified set of constraints. However, when these constraints are violated, the regulator immediately assumes control of the object. For example, a given unit process may operate only a limited number of hours before maintenance. Until this specified limit is reached, the regulator has no influence upon the object's operation. However, when the limit is reached, the regulator prevents the object from processing any further tasks.

Our definition of the regulation control mode also solved another control concern that we experienced in the manufacturing sector when we considered quality control. In the execution of many manufacturing tasks, it is often impossible to make an immediate determination about whether or not the processing steps are being correctly implemented. There is also an inherent delay in obtaining feedback information regarding quality resulting from the completed task until subsequent processing tasks are completed. For example, inspections may not occur after each processing step. The executive and coordination control structures discussed above usually consider the current and future processing tasks only. These cannot act upon quality information because such information pertains to completed tasks. The regulatory control function, however, does act upon this quality information and may inhibit performance of future tasks. Another situation occurs within quality control when we review trends arising from the processing of several items within a given unit process. Here, control is not focused on the processing of a single item, but rather, on a stream of processed items. Our regulatory control permits us to look at an ensemble of processed items rather than just the single item that is currently being processed.

We are still uncertain about whether or not other control functions may be defined. We are certain, however, that the aforementioned control functions are necessary for the operation of most large-scale systems. It is possible that all the control functions can be addressed by each CO in the ROOCH for the modeled system. Hyper-linked architectures, however, allow the overall control functionality to be distributed across multiple control structures. Thus, a given object may be subordinate to several different objects when all modes of control are to be addressed. Apparently, the only time that a true hierarchy is needed is when the basic coordination function is distributed; this distribution determines how each object will plan to execute its assigned tasks.

5 FUTURE SIMULATION NEEDS

HLM cannot currently be implemented with the available planning, control and simulation technologies. Obviously new planning and control algorithms must be defined for these complex systems. Given the wide range of planning and control problems to be considered within a given enterprise, it is unlikely that any proposed planning or control algorithm will be sufficiently robust to apply to every CO. Hence, a suite of algorithms will need to be defined. Similarly, the current technologies for distributed planning and control must also be expanded to consider these complex discrete-event systems. The available algorithms have only been defined for the simplest of the planning and control problems. We believe that the fundamental concepts of the hyper-linked framework are totally consistent with the existing algorithms and will provide guidance for their future embellishments. Still, much research remains.

The complexity of these systems necessitates that simulation remains the primary tool for evaluating the performance of a given object under a planned course of action. With respect to simulation, there are several immediate research concerns that must be addressed.

Concern One: Development of New Simulation Tools. As stated in the introduction, current simulation tools are simply inadequate for modeling most flexible manufacturing systems (FMSs). In the past two decades, these manufacturing systems have become highly automated with numerous controllers managing the workflow. Current simulation tools do not permit us to model the impact that these controllers' have upon the performance of systems. These focus upon modeling the flow of the job entities only. Existing simulation paradigms also fail to consider the flow of supporting resources such as tooling, fixtures and processing plans. Flanders and Davis (1995) documents the concerns which arise when we ignore these auxiliary flows. In fact, in our modeling of FMSs, we have consistently demonstrated that the auxiliary flows are the primary limiting factor upon the systems' operations. Davis et al. (1993, 1995a) discusses these concerns in detail.

We also need a suite of simulation tools. It is impossible for a single simulation tool to address all modeling situations that arise in the manufacturing enterprise. There are already virtual processing tools which demonstrate the behavior of a given unit process when it executes a specified set of processing instructions. Simulation tools for modeling communication networks also exist and such networks are indeed critical to communication among the objects. We note that information flows are seldom considered in the current modeling of manufacturing systems, yet we have already encountered FMSs where information flow constraints may be reducing the overall performance by ten percent or more. Sieveking and Davis (1995) recently defined another new, object-oriented simulation tool to support master production scheduling with integrated materials and capacity requirements planning. This tool provides for a detailed accounting of all inventories and permits multiple processing plans to exist for each manufactured item.

These are only a few of the many tools that will be needed. We simply do not believe that a single simulation tool, (especially the current tools), can address all modeling concerns which arise in the enterprise or any other large-scale system. New simulation tools must be developed.

Concern Two: The Models Must Integrate. Presently, reported modeling efforts have focused upon singular subsystems such as a FMS. In general, it is impossible for the simulation models for two FMSs to be integrated to ascertain the performance of their coordinated operation. On the other hand, we will never be able to construct a monolithic simulation model for an entire enterprise, nor is this a goal desirable. We know now that there are a multitude of reasons, (not only complexity concerns), which necessitate that planning and control be distributed. A monolithic model will not support this distribution. Each subsystem must have its own model of its interactions with its subordinate subsystems. Under the HLM framework, a given object's model must be accessible by more than one object. It is clear that standardization among the modeling approaches is needed such that integration will be possible. The agency which will set these standards is, as of yet, undetermined.

Concern Three: We Need Improved Validation Procedures. Most simulation studies are currently performed off-line to project the performance of a proposed system. Currently, few simulation models are being used on-line to support planning and control. Most of the subsystems for these large-scale systems are time-varying. This results as new products and processing plans are introduced into the systems. It is also a reflection of the introduction of new supporting resources, such as tooling. The time-varying nature of these subsystems implies that the need for validation of the subsystem models is constant. It is difficult to conceive that the validation process for all the included subsystems could be addressed manually. There must be a movement toward the develop of auto-validation procedures such that the modeled objects can collect real-time processing data and update critical parameters in their models as subsystems evolve in time.

Concern Four: We Need Distributed Simulation Techniques. There has been much research in the simulation literature about executing simulations on concurrent processors. We believe that a distributed capability will be essential to the management of complex systems. As stated previously, we do not believe that it is feasible or desirable to construct monolithic models for these large-scale systems. We believe that the models will be distributed among the objects that are managing the distributed planning and control of these systems. In order to ascertain the response of a selected ensemble of these objects, distributed simulation is essential. Other fundamental issues also need to be addressed. For example, can we define the objects in a manner, such that their distributed simulation can be coordinated without techniques such as "time-warping." If these techniques are essential, what are their consequences upon the distributed planning and control of these systems?

Concern Five: We Need Improved Real-Time Simulation Capabilities. Real-time simulation has been discussed by Harmonosky (1994), Davis, Wang and Hsieh (1991), Tirpak, Davis and Deligiannis (1992) and others. Still, our capabilities in this area are embryonic. We do not have the essential statistical techniques that can summarize the real-time performance data and assist in comparing one alternative planned course of action against another. Most of the statistical analysis for simulation has been dedicated to off-line simulations. This is a major research area. The problem is becoming even more critical because our research to date has shown that these systems are not simply stochastic. We have experienced more than one instance in which the existence of multiple attractors for the response occurred (see Davis et al. 1995a). Complexity theorists have observed that the simplest nonlinear systems can have multiple attractors. There is no reason for us to assume that complex discrete-event systems should be immune from this phenomena. If this phenomena does exist (and we believe that it does), then there is a need to radically rethink the approaches applied to the statistical analysis of these systems.

6 CONCLUSIONS

Given the extensive list of research needs defined above, the immediate question arises, "Is HLM a viable or essential approach to the CIM problem?" We believe that the answer to this question is "yes." First, HLM does support object-oriented design principles which have become a standard modeling approach in simulation and other engineering arenas. Second, it does permit the inclusion of intelligent agents capable of planning the tasks which these will execute. Third, it permits these intelligent agents to be assimilated into planning and control structures which are not limited to pure hierarchical arrangements. In general, these large-scale systems are too complex to expect that a simple hierarchy can be defined to support all the essential interactions needed to plan and control these systems. Finally, HLM provides an integral role for simulation, a technology which we feel is essential to predict the behavior of subsystem objects operating under a planned course of action.

To date, the authors have not found another modeling framework which supports the modeling flexibility that is associated with the hyper-linked architecture. Furthermore, few, if any, of the available frameworks have demonstrated that they are truly extensions of the conventional algorithms for distributed planning and control. This fact has also been demonstrated with hyper-linked architectures. Finally, there is a simplicity that has evolved from the ability to define a single object that is capable of simultaneously supporting temporal, spatial and functional decompositions as well as aggregation/disaggregation. The definition of the coordinated object and its included hierarchical subsystem coordinator has certainly increased our fundamental understanding of these decomposition strategies. We further feel that it will eventually permit us to extend the decomposition principles to consider far more complex systems.

REFERENCES

Davis, W. J., Wang, H., and Hsieh, C. 1991. Experimental Studies in Real-Time, Monte Carlo Simulation. *IEEE Trans. on Systems, Man and Cybernetics* 21:802-814.

Davis, W. J. 1992. A Concurrent Computing Algorithm for Real-time Decision Making. In *Proc. of the ORSA Computer Science and Operations Research: New Developments in their Interfaces Conference*, eds. O. Balci, R. Sharda and S. Zenios, 247-266. Pergamon Press, London.

Davis, W. J., Jones, A. T., and Saleh, A. 1992. A Generic Architecture for Intelligent Control Systems. *Computer Integrated Manufacturing Systems* 5:105-113.

Davis, W. J., Setterdahl, D. L., Macro, J. G., Izokaitis, V., and Bauman, B. C. 1993. Recent Advances in the Modeling, Scheduling and Control of Flexible Automation. In *Proc. of the 1993 Winter Simulation Conference*, eds. G. W. Evans, M. Mollaghasemi, E. C. Russell, and W. E. Biles, 143-155. The Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Davis, W. J., Macro, J. G., and Setterdahl, D. L. 1994. An Object-Oriented, Coordination-Based Simulation Model for the RAMP Flexible Manufacturing System. In *Proc. of the Flexible Automation and Integrated Manufacturing Conference*, eds. M. M. Ahmad and W. G. Sullivan, 138-147. Begell House, Inc., New York.

Davis, W. J. 1995. Hyper-Linked Architectures and Their Application to Computer-Integrated Manufacturing. Submitted to the *IEEE Trans. on Systems, Man and Cyberbetics*.

Davis, W. J., Macro, J. G. and Setterdahl, D. L. 1995a. An Integrated Methodology for the Modeling, Scheduling and Control of Flexible Automation. To appear in *IEEE Trans. on Robotics and Automation*.

Davis, W. J., Gonzalez, F. G., Macro, J. G., Sieveking, A. C., and Vifquain, S. V. 1995b. Hyper-Linked Architectures for the Coordination of Large-Scale Systems. To appear in *Proc. of the IEEE Workshop on Architectures for Semiotic Modeling and Situation Analysis in Large Complex Systems*. The Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Flanders, S. W. and Davis, W. J. 1995. Scheduling a Flexible Manufacturing System with Tooling Constraints: An Actual Case Study. *Interfaces* 25:42-54.

Harmonosky, C. M. and Robohn, S. F. 1991. Real-time Scheduling in Computer Integrated Manufacturing. *International Journal of Computer Integrated Manufacturing* 4:331-240.

Mize, J. H., Bhuskute, H. C., and Kamath, M. 1992. Modeling of Integrated Manufacturing Systems. *IIE Trans.* 24:14-26.

Sieveking, A. C., and Davis, W. J. 1995. An Object-Oriented Simulation Language for Master Production Scheduling in a Flexible Manufacturing Environment. To appear in the *Proc. of the 1995 IEEE International Conference on Systems, Man and Cybernetics*. The Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Tirpak, T. M., Daniel, S. M., LaLonde, J. D., and Davis, W. J. 1992. A Fractal Architecture for Modeling and Controlling Flexible Manufacturing Systems. *IEEE Trans. on Systems, Man and Cybernetics* 22:564-567.

Tirpak, T. M., Deligiannis, S. J., and Davis, W. J. 1992b. Real-Time Scheduling of Flexible Manufacturing. *Manufacturing Review* (ASME) 5:193-212.

AUTHOR BIOGRAPHIES

WAYNE J. DAVIS is a professor of General Engineering at the University of Illinois. His active research areas include simulation, computer-integrated manufacturing, and real-time planning and control of discrete-event systems. He collaborates with the Automated Manufacturing Research Facility and the Electronics Manufacturing Productivity Facility. He is a member of ASME and INFORMS.

JOSEPH G. MACRO is a doctoral student in the Department of Mechanical and Industrial Engineering at the University of Illinois. He is a recipient of the US Department of Energy Integrated Manufacturing Predoctoral Fellowship.

FERNANDO G. GONZALES is a doctoral student in the Department of Electrical and Computer Engineering at the University of Illinois. He is a recipient of an NSF Fellowship for Under Represented Minorities in Engineering.

ANDREW C. SIEVEKING recieved an MS in General Engineering and a MBA from the University of Illinios. He is currently employed as a production engineer at Proctor and Gamble in Cincinnati, Ohio.

SCOTT V. VIFQUAIN received an MS in General Engineering at the University of Illinois. He is currently employed as a consulting engineer at Solomon and Associates in Atlanta, Georgia.