*Proceedings of the 2002 Winter Simulation Conference*
*E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.*

# IMPLEMENTATION OF A SIMULATION-BASED CONTROL ARCHITECTURE FOR SUPPLY CHAIN INTERACTIONS

Sreeram Ramakrishnan
Seungyub Lee
Richard A. Wysk

Department of Industrial and Manufacturing Engineering
Pennsylvania State University
University Park, PA 16802, U.S.A.

## ABSTRACT

Techniques based on discrete-event simulation have been widely used for network analysis and policy optimization in the domain of supply chain management. Previous researchers have developed and implemented architectures for simulation-based control for shop floor. A more detailed and high-fidelity simulation model is used for control purposes as opposed to that used for analytical purposes alone. This paper discusses the issues related to implementing a simulation based control architecture for actively controlling supply chain interactions.

## 1 INTRODUCTION

The different decisions associated with supply chain management have been classified as strategic (long-term), tactical, and operational (short-term), and can be treated as location, production, distribution, or transportations problems (Beamon, 1998). Different criteria have been used to classify multi-stage models that deal with Supply Chain Management (SCM) problems. They have been classified as network design, 'rough cut' methods, and simulation models or as deterministic stochastic analytical models, economic models, and simulation models, based on the modeling technique used. Classifications based on solution techniques (Thomas and Griffin, 1996), level of decisions made in the model (Ganeshan, 1997), and problem formulation techniques also exist.

Discrete event simulation (DES) has been used extensively for network optimization, policy optimization, identification of the causes of the uncertainties and their impact, and in the development of methods to reduce/eliminate the uncertainties. The use of DES in strategic SCM planning has been discussed in a "four-step methodology" in Schunk and Plott (2000). The role of simulation is not confined to the optimization of network structures and policy decisions, but also lends itself in analyzing the effects of variability – thereby providing critical input for "design for robustness".

In this paper, a fifth step – simulation-based control, is proposed. This step is added to aid the implementation of the control policies derived from the previous four steps discussed in Schunk and Plott (2000). As can be seen from the objectives of the different steps, the different levels also correspond to different levels of data abstraction. In the highest level which deals with network optimization, the objective is to identify a few alternative structures of the supply chain. This could be in terms of the number and location of distribution centers, for example. The data requirement at this stage is "more aggregate" than that at subsequent levels. In the next level, the objective is to obtain an initial understanding of the networks' behavior under different known demand patterns in order to select a fewer number of alternative networks that need to be analyzed further. Once a particular network structure has been decided upon, the "optimum" control policies for that network need to be identified. However, the simulation models used for policy optimization will be more detailed than those used to predict network behavior. The most data-intensive use of simulation in the context of SCM occurs when simulation models are used to actively control the supply chain interactions. In this scenario, the system needs to keep track of even the "smallest" event – such as loading a part in a machine, in order to effectively control the entire supply chain, even if such actions need not necessarily have immediate implications for the chain's behavior. Figure 1 illustrates the use of simulation models in SCM.

The concept of simulation-based control was successfully demonstrated in the RapidCIM architecture and its associated tools developed at The Pennsylvania State University and Texas A&M. Figure 2 shows the architecture for simulation-based control as used in the shop floor environment. The RapidCIM architecture and the associated tools are capable of automatically generating much of the

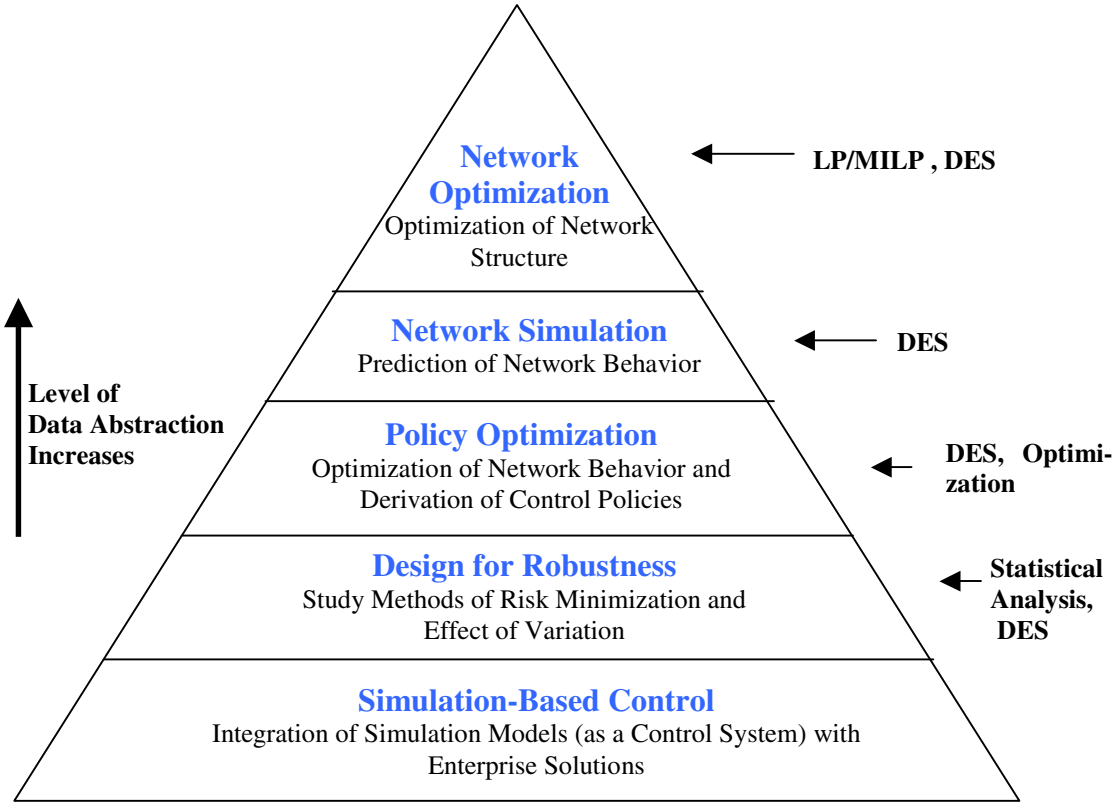


Figure 1: The Use of Simulation Techniques in Supply Chain Management

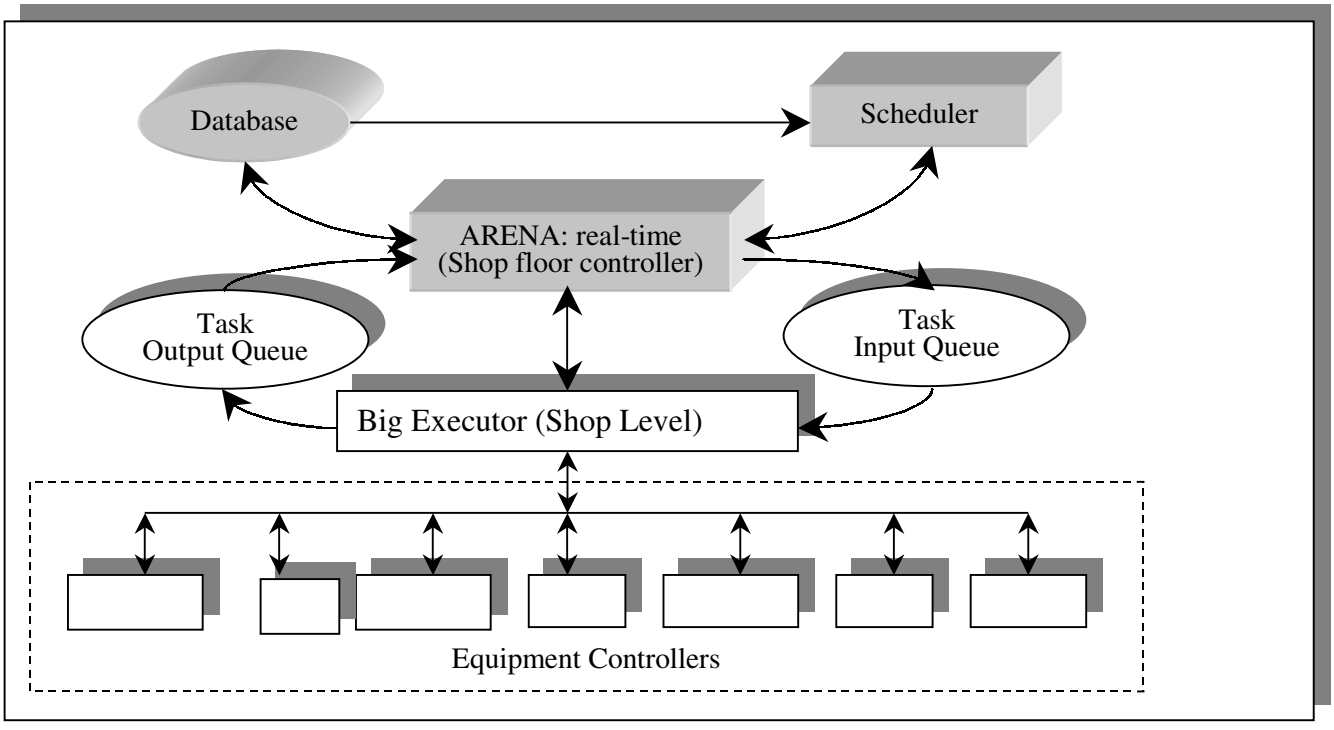


Figure 2: RapidCIM Architecture for Shop Floor Control (Single Simulation Model)

software for the equipment level controllers necessary for automating discrete manufacturing systems (Smith, et al., 1994, Wysk, et al., 1994, Wysk and Smith, 1994, Wu and Wysk, 1989, Son, et al., 1999). In general, the architecture has software tools developed based from finite state automata models that link the simulation model directly to controllers of the physical equipment. Detailed discussions on the RapidCIM concepts and implementation specifics can also be found at the new alpha site at Penn State's Factory for Advanced Manufacturing Education (FAME) laboratory's website at <http://www.engr.psu.edu/cim/FAME/index.html>

## 2 RESEARCH BACKGROUND AND OBJECTIVE

Most SCM analytical models treat the variables as deterministic or as well characterized probabilistic distributions. However, that assumption of deterministic data is not a feasible one in a dynamic domain when deriving active control policies. It is therefore imperative to have access to real time information on the critical variables. A simulation-based control approach for distributed application such as that seen in supply chains is discussed here. In a purely deterministic domain, an approach involving high fidelity models is not expected to yield any gains over traditional techniques.

However, the amount of details that needs to be modeled is relatively high if supply chain interactions need to be control using a simulation-based control approach. The complex nature of the interactions necessitate the need to include significant level of details in the modeling effort. Moreover, since the objective is to implement simulation-based control., it is necessary to have high fidelity models, thereby using a single model as in RapidCIM is not feasible. Issues of modularity and maintainability also require that distributed simulation models be used when a simulation-based control architecture is to be implemented for actively controlling supply chain interactions. The issue of coordination of the models become a critical issue when these models are distributed.

An architecture for real-time control of supply chain interactions was presented in Ramakrishnan and Wysk (2002). A modeling technique wherein a federation of real time simulation models (and fast mode models) serving as active controllers of each supply chain entity, fit within a common architecture can be used for descriptive and prescriptive objectives and for active control of the value chain interactions was discussed. In this paper, the interactions among simulation models is discussed. The proposed architecture also serves as a framework for integrating databases and tools such as MRP systems with simulation models being used for active control.

## 3 SIMULATION-BASED CONTROL FOR SUPPLY CHAIN

An object-oriented (appropriate to model distributed applications such as those found in a value chain), scalable, simulation-based control architecture is proposed in this paper. In this architecture each supply chain entity constantly evaluates its performance with respect to its assigned roles and triggers a re-assessment of the roles if necessitated by any changes observed by it through its simulation models. The proposed architecture can be used to analyze the behavior of the supply chain in "fast mode" as well as in its active control. Also, the architecture demonstrates the use of DES as a tool for software integration. real-time simulation model (by definition, which runs at wall-clock speed).

In this architecture, each entity in the supply chain has two simulation models associated with it – one running at real-time and the other a "look-ahead" simulation. This model can, at appropriate conditions (discussed later), invoke the look-ahead simulation model associated with the same entity. The look-ahead model is capable of predicting the impact of a "disturbance" observed by the real-time model with respect to certain pre- determined performance measures for that entity which are implementation-specific. A Federation Object Coordinator (FOC) coordinates the real-time simulation models of all the entities in the modeled supply chain. The real-time simulation model associated with each of the entities can invoke the federation of simulations to obtain the current information from each of the entity when it perceives that a deviation from its assigned role is imminent (change in shipping schedule, production quantity, etc.). This information can be then used to re-solve the problem (production-distribution, for example) through traditional optimizing tools as and when required. A messaging system based on the scheme provided in Lee (2002) is used to achieve the required coordination and information transfer among the simulation models, between the FOC and the simulation models, and between the FOC and the optimization tool.

Such an approach involving active control is not feasible using purely stochastic or deterministic analytical models. "*Control"* as used in this context, refers to automatic triggering of value chain interactions such as request for quotes (RFQs), purchase orders (POs), transshipment and resource allocation decisions in the ERP/MRP systems, real-time, based on the conditions perceived in any partner in a value chain. Figure 3 shows the architecture for real time control application while Figure 4 shows the architecture when the models are used for analysis purposes.

### 3.1 Example Scenario

In this section, an example based on a supply chain segment for Printed Circuit Board (PCB) assembly houses, is

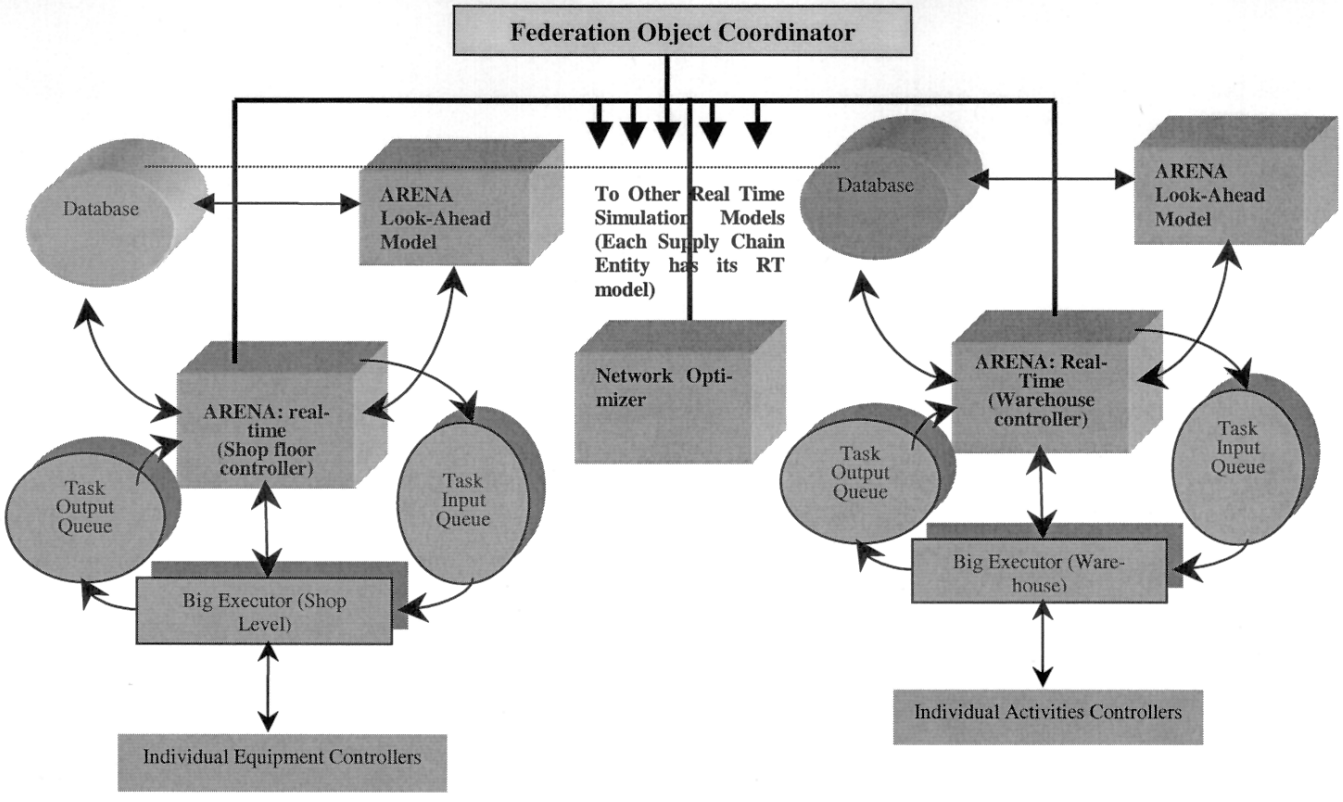


Figure 3: Simulation-Based Control Architecture for Supply Chains

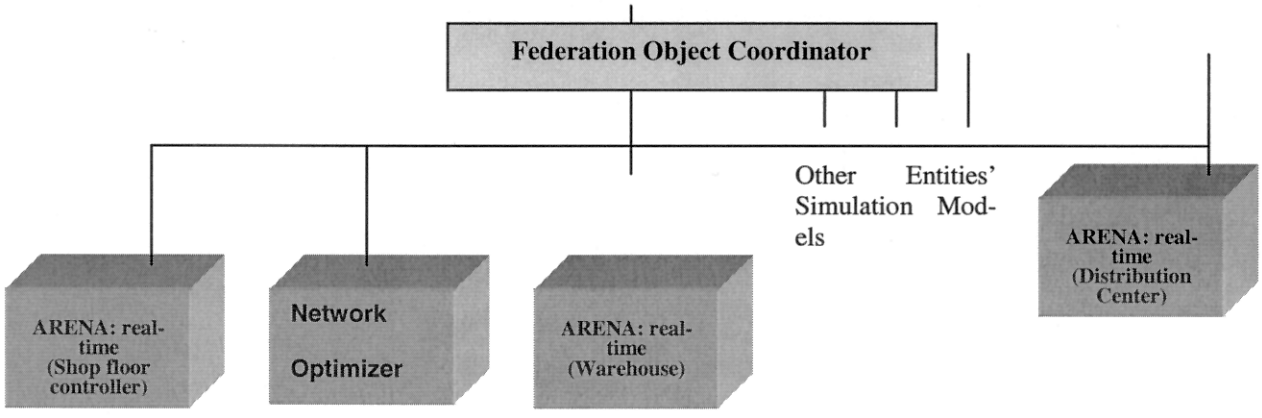


Figure 4: Models in Analysis Mode (Fast Mode) Database Connections Not Shown

used to illustrate the implementation of the architecture discussed in Ramakrishnan and Wysk (2002). The example being implemented as part of a simulation test bed at Penn State, will be subsequently used to compare the performance of simulation-based control with analytical models. Consider a scenario as shown in Figure 5.

Several heuristics exist for process planning and batch scheduling for PCB assembly lines. The importance of obtaining real-time data from the PCB assembly line in order to make decisions on scheduling, part routing, and potential changes in process plans has already been established (Srihari, et al., 1994; Cala and Srihari, 1995; Ku, et al., 1996; Wu and Srihari, 1996). The test bed will enable to test the implementation requirements of the discussed architecture for a segment of a electronics-manufacturing supply chain. Each entity in the segment will have two simulation models – a real-time model and a fast mode model. The real-time models of all the entities are coordinated using a FOC as mentioned before. Each of the real time models is linked to its corresponding look-ahead (fast mode) model which can be invoked by the former as and when required. (In the cur-

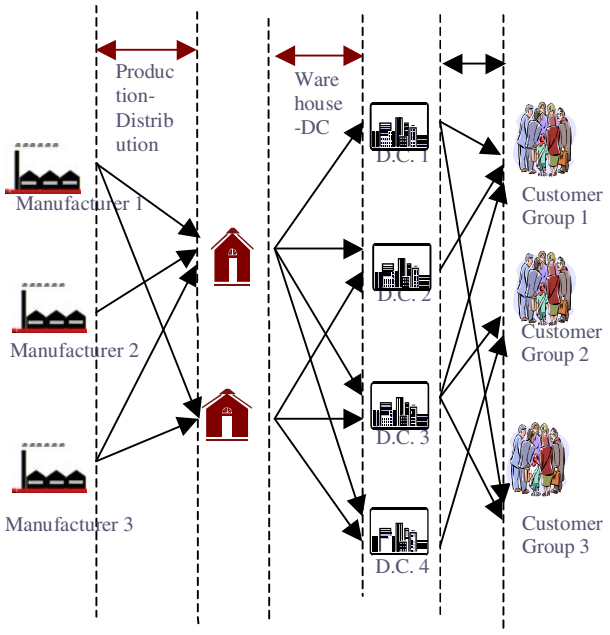


Figure 5: Example Scenario

rent implementation, the distribution centers have been modeled as database objects only.)

### 3.2 Structure of Simulation Models

As discussed earlier, each supply chain entity has two models associated with it. The real-time models are linked to the actual physical system and controls the tasks of the various resources as in the RapidCIM architecture mentioned in section 2. The look-ahead model associated with each supply chain entity is exactly same as the real time model, except that it can be run in fast mode, and has no links to the controllers in the physical system. The real-time simulation model associated with a PCB assembly line, for example, can incorporate different conditions (implementation-specific) at which the status of the systems needs to be verified.

All simulation models have been built in Arena 4.0. In the PCB assembly line models, the following processes have been modeled: kitting, component placement (chip-shooter, Tape-automated bonding, through-hole and surface mount components, etc.), solder deposition (stencil printing, dispensing), solder formation (wave soldering, reflow soldering), cleaning, in-circuit and functional testing, and packaging. The database also stores information on the process routings for particular "builds", process plans, and production schedule. These models have various conditions incorporated in them which correspond to scenarios where that line's performance deviates from its assigned role. These include delay in processing, delays in shipping, high rate of defects detected in testing, machine break-downs, scheduled maintenances, and shortage of material.

For example, one of the conditions incorporated in the real-time simulation model is to verify if specific batches have been assembled at scheduled times. In order to continuously monitor this condition, the simulation model has access to the database which contains updated information on the current scenario in the assembly line. The simulation model verifies for both lateness and earliness penalties. Since it is not sufficient to know whether a batch has been completed on time, but if it is early or late (within a pre-defined "tolerance" interval), the model verifies when each batch is completed. This information is used to ensure that the required production rate is being maintained. This can be achieved by monitoring the number of batches being processed and crosschecking it with the production schedule for that entity.

The interaction of the simulation model with the database is achieved using SQL queries embedded as a .DLL file as discussed in Lee (2002). A "SCAN" block is used to keep track of the identification "key" of the batches and their current location. Using the database, the simulation model can determine if the production is on time. The triggering of database interaction is achieved by using "EVENT" blocks. An event block can be implemented in either Visual C++ or Visual Basic Application (VBA) embedded in Arena software. Whenever an entity in the simulation model reaches one of these blocks, the function "cevent" in the interface code is executed. As noted earlier, different conditions can be monitored in the models. Different EVENT blocks are used to achieve a particular task. For example, 3 different events are used to achieve a "SELECT", "UPDATE", and "DELETE" queries, depending upon the logic in the model. A fourth EVENT is used to verify the condition being checked, in conjunction with the SCAN block for that condition. In scenarios where the condition evaluated using EVENT (4) indicates that the supply chain entity's performance is deviating from its assigned role, the next step is to evaluate if the deviation can have any impact on its role during the decision horizon.

In order to analyze the impact of the discrepancy (delayed shipment, shortage of material, earlier production, etc.) a look-ahead model is invoked. Before invoking the fast mode model, the current information of the shop floor is saved to a database using user-written VBA code. The SIMAN functions SAVE and RESTORE cannot be employed in Arena (Son, et al., 1999). The look-ahead model imports the current status of the shop floor (in this case), and then evaluates the impact of the disturbance on its assigned roles (production schedule, shipment, delaying RFQs, inventory replenishment, etc.). The result is communicated to the real time model, which determines if any deviation for the decision horizon is imminent. If no deviation is noted for the horizon, the simulation model does not take any action and continues controlling the physical system. If a deviation is imminent, the real time model decides to invoke the FOC.

Based on the situation faced by the value chain (reduction in inventory in certain echelons, increase production in some facilities, change warehouse assignment, etc.) the FOC requests requisite information from all the real-time simulation models. Each of the simulation models obtain the current information from their respective MRP/ERP systems and communicates it to the FOC. This information is aggregated and then input to the optimization tool. The aggregation of the information is dependent upon the optimization tool used and the problem formulation, and is hence case-specific. The new 'solution' is then made available to the FOC, which communicates it to all the real time models in the system. Even though the FOC aggregates the information, it needs to be emphasized that the FOC can be invoked by any entity in the supply chain and that it serves as a coordination manager and not as a centralized decision maker for the value chain. The proposed architecture assumes that all modeled entities in the supply chain are willing to share information with the FOC and a centralized decision making algorithm or model is acceptable.

### 3.3 Communication Between Fast Mode and Real Time Models

Figure 6 shows a sequence diagram for the messaging that occurs between the simulation models of each entity and the FOC. The messaging of the FOC has not been discussed in detail in this paper. The FOC acts as a facilitator for the entities in the value chain by acting as a "router" for the information required from each entity when the situations necessitates re-solving the value chain problem. The system is flexible since any entity can invoke the FOM as and when required.

The communication between the real-time simulation and the look-ahead manager is implemented using Remote Procedure Call (RPC). It needs to be noted that since two modes (real-time mode and fast mode) cannot run simultaneously on one computer, the implementation of the architecture requires that the two models be run on different computers. Moreover, multiple models, even under the same mode, cannot run simultaneously on the computer.

Each look-ahead model has a look-ahead manager, based on the Arena RT console, which acts as a server under the client/server environment constructed by the RPC. The manager keeps "listening" to a predetermined port in that machine for any messages that come from its real time model. In the current implementation, the real time model sends a message containing two parameters: the first parameter is a signal to start, while the second parameter specifies the output required by the real time model – cost, inventory level, machine status, etc. A windows-based router is used for exchanging messages between the two models. Each entity is "logged" on to the router with an identification name, defined in a "default.map" for that implementation. In the current implementation, a direct address system is used.

When the real time model perceives the need for verifying the impact of an observation, it invokes the look ahead model by sending a message to a particular address which is pre-defined. The port to which the message is sent is the one the look ahead manager is "listening" to. Upon receiving a message from the real time model, the look ahead manager invokes the fast mode model. The RT console-based look-ahead manager is written in Visual Basic. Appropriate code was added to ensure that relevant action for the messages received from the real time model was taken. For example, a "SHELL" command was used to launch the look ahead model.

The look ahead model, upon being opened, creates a dummy entity to retrieve the "saved" information from the database. This manager is executed in a different machine. First, a dummy entity is used to initialize the system using the information from the database (condition of each machine, batch number and member number processed) is used to initialize the model using user-written code in the "Document Open" method. The fast mode model is "run" for a pre-determined interval (equal to the decision horizon in this implementation, though the look-ahead period can be easily changed as required, provided there is sufficient information regarding production schedules in the database). In order to fasten the simulation runs of the look-ahead model, components such as the "DELAY" blocks were removed. Moreover, some of the process sequences were "hard-coded" instead of using "EVENT" blocks to interact with a database to determine the next processing step as was the case in the real time model. After the model is run for the specified time period, the result from the run – lateness or earliness penalties, estimated completion time, inventory level, etc., is communicated to the real time model. The real time model evaluates the response and decides if the FOC needs to be invoked to help in the re-optimization of the supply chain interactions.

## 4 CONCLUSIONS

Simulation-based control techniques have been demonstrated for shop floor control by previous researchers. The need for distributed simulation models for controlling supply chain interactions was discussed in this paper. A related paper discussed a distributed real time simulation based control architecture for supply chain interactions. This paper discussed some of the implementation issues of the proposed architecture. The architecture allows for the active control of the interactions and fast mode analysis of the entire system. In the analysis mode, the architecture involves the use of high fidelity real time models which identify any disturbances and fast mode models which analyze the impact of that disturbance. The mode of the FOC determines the use of the architecture as an analytical tool or

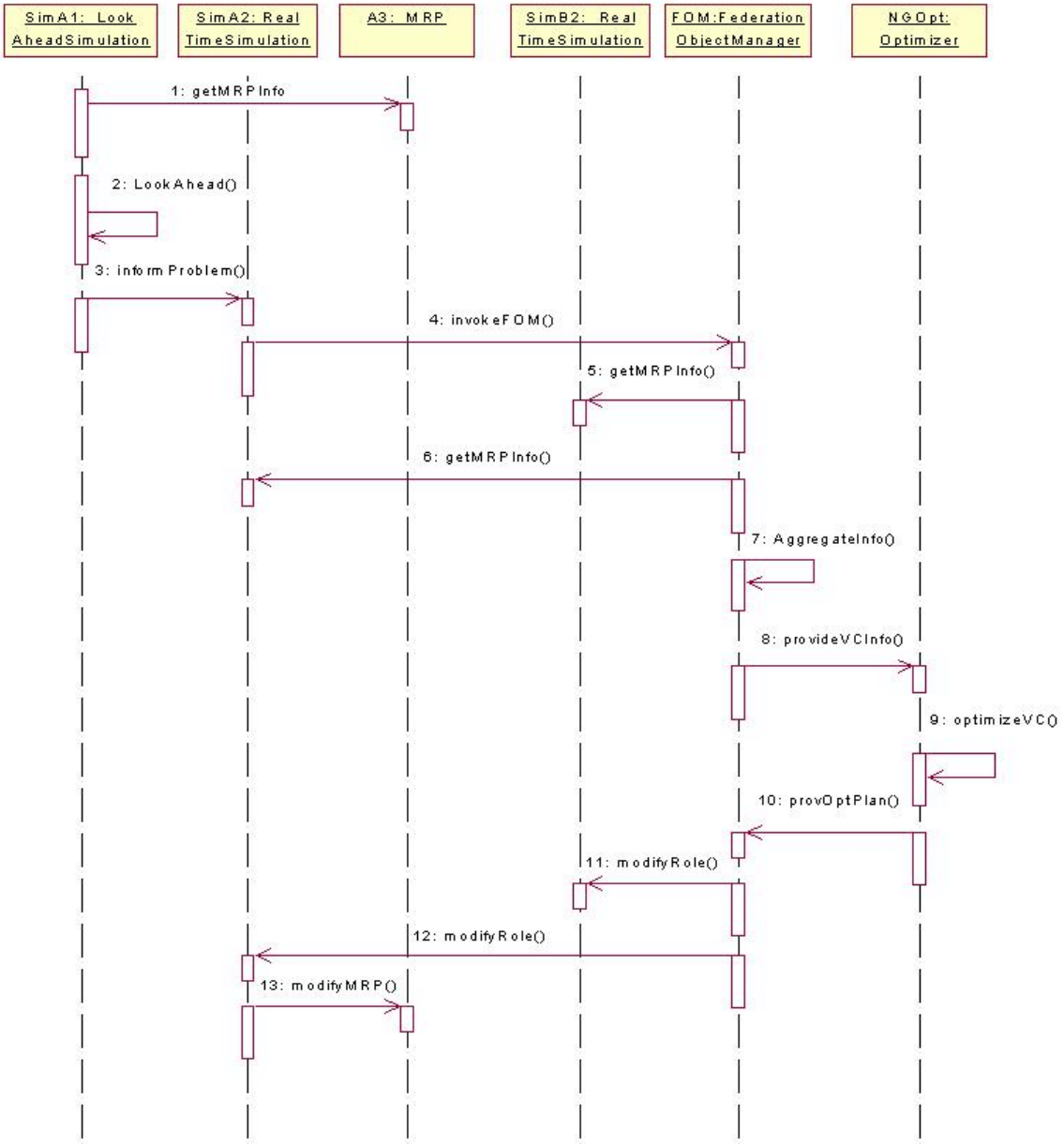


Figure 6: Sequence Diagram for Architecture

as a control execution system. A case study based on electronics manufacturing is being implemented at Penn State and will be used to compare the performance of this methodology with traditional analytical models.

## REFERENCES

Beamon, B. 1998. Supply Chain Design and Analysis: Models and Methods. *International Journal of Production Economics* 55 (3): 281-294.

Cala, M., K. Srihari, and C.R. Emerson,. 1995. Process and Production Planning for Circuit Board Assembly, *IIE Transactions*, 27: 90-95

Ganeshan, R. 1997. Analytical Essays in Supply Chain Management, *Ph.D. dissertation*, Department of Management Science and Information Systems Pennsylvania State University, University Park, PA

Ku, A., K. Srihari, and J. DiLella. 1996. Process Planning for Manual PCB Assembly. *Computers and Industrial Engineering* 31(1): 181-184.

Lee, S. 2002. An Architecture and Implementation of a Distributed Computer-Controlled Shop Floor System Using a Federation Object Manager. *M.S. Thesis,* Department of Industrial and Manufacturing Engineering, Pennsylvania State University, University Park, PA

Object Management Group. UML, Website Document, 2000, <http://www.uml.org/> [accessed April 10, 2001]

Ramakrishnan, S., and R. A. Wysk. 2002. A Real Time Simulation-based Control Architecture for Supply chain Interactions, Industrial Engineering Research Conference (on CD-ROM) Available at <http://fie.engrng.pitt.edu/iie2002/proceedings/ierc/papers/2030.pdf> [accessed July 15, 2002]

Schunk, D., and B. Plott. 2000. Using Simulation to Analyze Supply Chains, *Proceedings of the 2000 Winter Simulation Conference*: 1095-1100

Smith, J. S., R. A. Wysk, , D. T. Sturrok, S. E. Ramaswamy, G. D. Smith and S. B. Joshi. 1994. Discrete Event Simulation for Shop Floor Control, *Proceedings of the 1994 Winter Simulation Conference*: 962-969.

Son, Y., H. Rodríguez-Rivera, and R. A. Wysk. 1999. A Multi-pass Simulation-based, Real-time Scheduling and Shop Floor Control System, *Transactions*, The Quarterly Journal of the Society for Computer Simulation International 16 (4): 159-172.

Srihari, K., J. A. Cecil, and C.R. Emerson. 1994. A Blackboard Based Process Planning for the Surface Mount Manufacture of PCBs. *International Journal of Advanced Manufacturing Technology* 9 (3): 188-194.

Thomas, D., and P. Griffin. 1996. Coordinated Supply Chain Management. *European Journal of Operational Research* 94: 1-15

Wu, C.H., and K Srihari. 1996. Automated Knowledge Acquisition for the Surface Mount PWB Assembly Domain, *Journal of Intelligent Manufacturing* 7(3): 393-398.

Wu, S.D., and R.A. Wysk. 1989. An Application of Discrete-event Simulation to On-line Control and Scheduling in Flexible Manufacturing", *International Journal of Production Research* 27 (9): 1603-1623.

Wysk, R. A., B. A. Peters, and J. S. Smith. 1994. A Formal Process Planning Schema for Shop Floor Control, *Engineering Design and Automation Journal*, 1(1): 3-19.

Wysk, R., A. and J. Smith. 1994. A Formal Functional Characterization of Shop Floor Control, *Computers and Industrial Engineering – Special Issue of Manufacturing Systems* 28 (3): 631-644.

## AUTHOR BIOGRAPHIES

**SREERAM RAMAKRISHNAN** is a doctoral candidate in the Department of Industrial and Manufacturing Engineering at Pennsylvania State University, University Park. He received his B.Tech (Mechanical) from College of Engineering, Trivandrum, India and his M.S. (Industrial Engineering) from Binghamton University where he won the department Award for Academic Excellence. His research interests are simulation-based control and supply chain management. Upon graduation, he will be an Assistant Professor in the Engineering Management Department at University of Missouri - Rolla. His email address is <sreeram@psu.edu>.

**SEUNGYUB LEE** is a doctoral candidate in the Department of Industrial and Manufacturing Engineering at Pennsylvania State University, University Park. He received his B.S. (Industrial) from Yonsei University, Seoul, Korea. His research interests are distributed simulation and simulation-based control for manufacturing applications. His email address is <sxl287@psu.edu>.

**RICHARD A. WYSK** is the Leonhard Chair in Engineering and a Professor of Industrial Engineering at Pennsylvania State University, University Park. Dr. Wysk has co authored six books including *Computer-Aided Manufacturing*, with T.C. Chang and H.P. Wang -- the 1991 IIE Book of the Year and the 1991 SME Eugene Merchant Book of the Year. He has also published more than a hundred and fifty technical papers in the open-literature in journals including the *Transactions of ASME*, the *Transactions of IEEE* and the *IIE Transactions*. He is an Associate Editor and/or a member of the Editorial Board for five technical journals. Dr. Wysk is an IIE Fellow, an SME Fellow, a member of Sigma Xi, and a member of Alpha Pi Mu and Tau Beta Pi. He is the recipient of the IIE Region III Engineer Award and the IIE David F. Baker Distinguised Research Award. He has held engineering positions with Award for Excellence, the SME Outstanding Young Manufacturing General Electric and Caterpillar Tractor Company. He received his Ph.D. from Purdue University in 1977. He has also served on the faculties of Virginia Polytechnic Institute and State University and Texas A& University where he held the Royce Wisenbaker Chair in Innovation. His email and web addresses are <rwysk@psu.edu> and <http://www.engr.psu.edu/cim/wysk.html>