Proceedings of the 2003 Winter Simulation Conference
S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds.

A SIMULATION-BASED TOOL FOR INVENTORY ANALYSIS IN A SERVER COMPUTER MANUFACTURING ENVIRONMENT

Heng Cao
Feng Cheng
Haifeng Xi
Markus Ettl
Steve Buckley

IBM T.J. Watson Research Center
Yorktown Heights, NY 10598, U.S.A

Carlos Rodriguez

IBM Enterprise Server Group
Poughkeepsie, NY 12601, U.S.A.

ABSTRACT

In this paper, we describe a simulation-based inventory management tool developed for the IBM Enterprise Server Group. Through the Web interface of the tool, an inventory manager is able to visualize Days of Supply (DOS) levels – current and projected, and to carry out what-if scenario analysis to identify potential opportunities for improvement. The highly complicated server manufacturing environment poses simulation modeling challenges such as two-stage fabrication/fulfillment process, multi-echelon bills-of-materials, complex server box configurations, part tests with random yields, stochastic lead times and so on. In the following sections, we will introduce the common characteristics of the server manufacturing environment, present the overall architecture of our tool, and describe the simulation model design and how we addressed those challenges. At the end of the paper, we will show some results collected from the tool and point out our future research directions.

1 INTRODUCTION

High-end enterprise servers are products built with leading edge technology featuring: expensive components with high inventory carrying cost; extensive tests for components for high quality requirements; multi-tier suppliers (both internal and external) with long supply lead time; high customer service levels requiring complex product configuration and quick order response time. Today's server manufacturers are typically operating in an environment with the following characteristics: aggressive new product introduction cycles; significant engineering change (EC) activity connected with continuous quality improvement driving component yield issues; extreme demand skews; significant forecast liability with suppliers due to the unique nature of the components and long lead times; currently the cost of missing a shipment far exceeds the cost of carrying inventory; in the future, market pressures may force margins to contract making inventory cost reduction paramount in order to insure continued product profitability.

IBM as a leading enterprise server provider strives to provide high quality services to customers through continuously improving its extended supply chain management. A combination of the traditional build-to-plan with make-to-order operations, which we refer to as the fabrication/fulfillment process in this paper, has been adopted to provide responsive order fulfillment and to lower the expensive inventory holding cost. The fabrication stage is a build-to-plan process, in which components (subassemblies) are procured, tested and assembled according to a predefined build plan. The component inventory is then kept in stock ready for the final assembly of the end products to customer order configurations. The fulfillment stage is a make-to-order process, which means that no finished goods inventory is kept for end products and the final assembly starts after the customer order is received. Through this process, the company can enjoy both the flexibility of mass customization and the speed and efficiency of mass production.

However, this hybrid process structure combined with inherent randomness in the process pose tremendous challenges to inventory management, particularly in terms of financial and operational impacts. To enable better analysis of these issues we developed a simulation tool to model the impact of randomness in parameters like lead times, yields and component usage rates. Simulation outputs include inventory costs, and Days-of-Supply (DOS) profiles.

As illustrated in Figure 1, this tool has two primary usages. One is to enable optimal inventory control. The DOS profile can be used by inventory managers to monitor the daily production, identify optimization opportunities, and develop action plans if the actual inventory level strays from the optimized DOS profile. The second usage is to enable "What–if" analysis to identify opportunities for per-

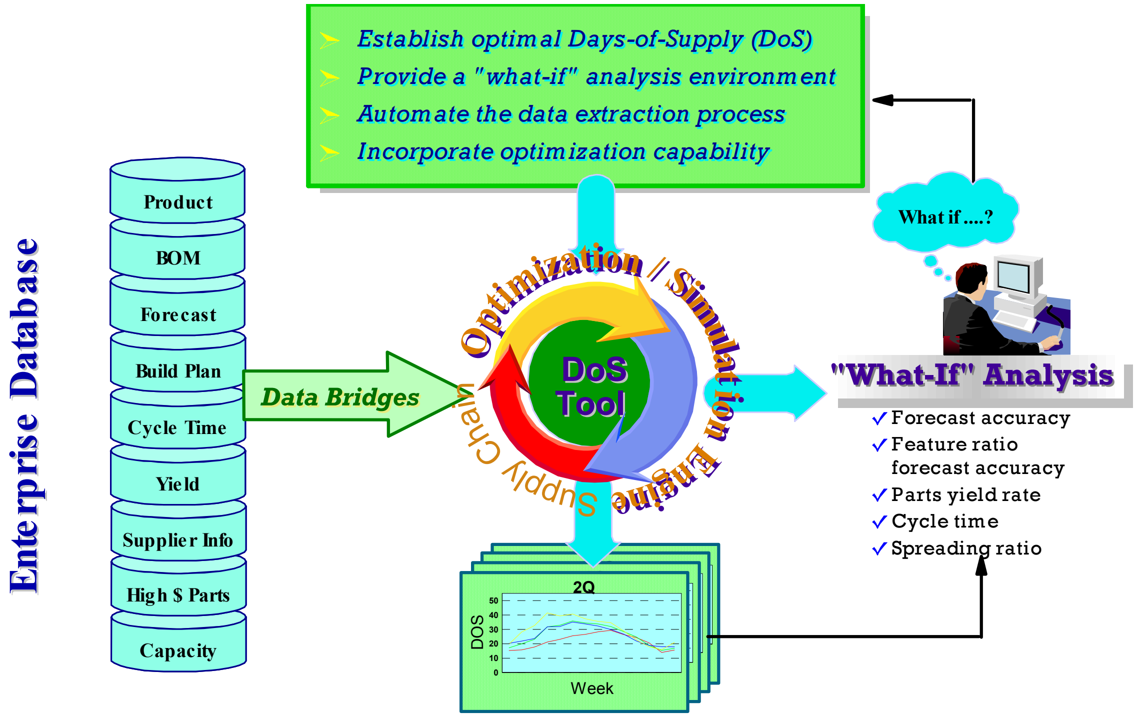

Figure 1: ESG Days-of-Supply Simulation Tool

formance improvements. Currently our tool allows inventory planners to perform sensitivity analysis on demand forecast accuracy, feature ratio forecast accuracy, component yield rate and build plan ratio distribution. The tool helps inventory planners assess the impact on inventory cost and DOS profiles if one of the above factors changes. The tool has a real time data linkage to keep the tool database synchronized with the production database.

In the following sections, we will present the overall architecture of our tool and the simulation model design, and describe how we addressed the modeling challenges in the server manufacturing environment. At the end of the paper, we will show some results collected from the tool and point out our future research directions.

2 ARCHITECTURE

The DOS tool is a distributed enterprise application system built with the multi-tier Java 2 Enterprise Edition (J2EE) framework as illustrated in Figure 2.

2.1 Tool Database

The DB2 database serves as the system's back-end tier. It stores all the input data needed for DOS simulation, such as end product demand distributions, configuration feature ratios, component test time distributions, BOM's, and simulation output reports such as customer serviceability, on-hand and pipe-line inventory. A custom data bridge utility has been built that can extract relevant raw data from production ERP systems, transform it into desired formats, and load it to the tool database. Whenever new data is available in ERP systems, the bridging utility is run and a new batch of data is loaded into the database to create a "Business-As-Usual" (BAU) DOS case, which will be used as a baseline for subsequent what-if scenarios.

2.2 DOS Application

Deployed as a J2EE Web application under the WebSphere Application Server, the DOS application sits in the middle tier between the back-end database and the front-end client, and contains all the business logic. End users connect to the application using a Web browser to view the current and previous DOS cases stored in the database, run DOS simulation on the current case, view simulation results of the current and previous cases, and conduct what-if analysis on the current case.

2.3 Simulation Engine

The simulation engine is a Java library deployed together with the Web application. It is invoked by the application upon request from an end user to compute the DOS profile for the current case, or when a what-if scenario is created and evaluated. Its design and functionality will be discussed in detail in Section 3.

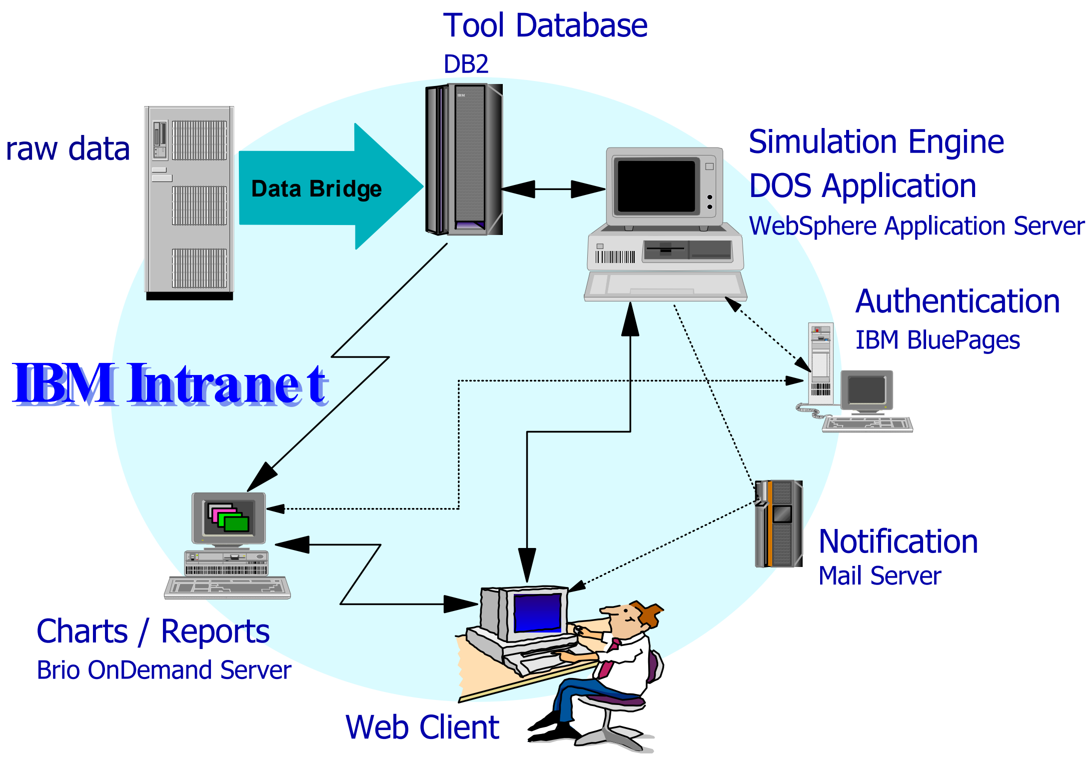

Figure 2: Solution Architecture

2.4 Reporting

To present the simulation results to business users in a meaningful way, custom reports were composed and deployed to a Brio® (http://www.brio.com/) OnDemand Server. These reports run predefined queries against the tool database, displaying query results in tables, charts, and pivot tables. When a user requests a Brio report, the appli-cation server redirects his request to the OnDemand Server that serves the report to his Web browser.

2.5 Intranet Services

Our system utilizes a few IBM intranet services to achieve better usability. To implement a single sign-on scheme, we have configured both the application server and the OnDemand server to use IBM Blue Pages, a corporate-wide directory service, for user authentication. The Web application also uses an intranet mail server to send Notes messages to notify users of completion of simulation runs.

3 SIMULATION OF SERVER MANUFACTURING DYNAMICS

Simulation, as the imitation of the operation of a real-world process or system over time (Banks 1999), includes both the process of designing a model with sufficient fidelity to represent its real world counterpart and carrying out time advance over the model to study its temporal behavior. In this paper, we term these two activities as *modeling* and *simulation execution*. The modeling task involves the computational representation of entities (e.g. components, subassemblies and final products) and the parameters of the manufacturing process when processing entities (e.g. build plans, costs, prices, bills of material, lead times, and yields). The simulation execution is done through a discrete event simulation library developed at IBM Research.

Existing commercial discrete event simulation packages such as SIMPROCESS, ARENA and PROMODEL offer general process modeling capabilities such as process delays, queues, branching, splitting, merging and resources (Swain 2001). These modeling capabilities can be deployed to build manufacturing and other types of process models. However, modeling complex supply chains with such low-level process description components is extremely time-consuming and prone to error. The IBM Supply Chain Analyzer (SCA) (Bagchi et al. 1998) offers more advanced supply chain modeling and optimization capabilities than general process modeling packages. However, SCA was built upon a proprietary simulation framework and implemented as a standalone desktop application. In its current form, SCA cannot be easily integrated into an operational enterprise computing environment (Kim et al. 2000).

Conceptually, we have aligned our simulation design with SCA's object-oriented approach, but we believe that adopting open industry standards will be the key to ensuring easy integration, interoperability and extensibility in supply chain modeling. As a result we have used XML to represent the supply chain simulation model data and the Java Event Delegation Model (Sun 1999) to implement the simulation runtime behavior of supply chain components (Cao et al. 2002). The application to the server manufacturing environment is an extension of our Java based supply chain simulation library. Figure 3 illustrates five of the most important classes in this library (*TimeMachine*, *EventHandler*, *EventDispacher*, *SimuAgent* and *SimuAgentGrp*). The abstract base class *SimuAgent* has an array list of *EventHandler* instances and *EventDispatcher* instances, and the *processEvent*() method of *SimuAgent* just iterates through these two array lists and invokes their *processEvent*()/*dispachEvent*() to process those events with a time stamp earlier than or equal to the current simulated time. The simulated time is advanced by *TimeMachine* which updates the current time clock to the smallest timestamp of all events in the event queues of all agents' event handlers/dispatchers. There are two types of events defined in the current model library, *internal* and *external*. Internal events trigger an object's internal state changes caused by simulation time advancement. External events such as orders and shipments are the vehicles for propagating state change notifications from a source object to one or more listener objects.

This base simulation library provides the discrete simulation time management for simulation execution, and the modeling details for the specific application need to be implemented in the subclasses of the SimuAgent and the objects used by those subclasses. For the DOS simulation model, three different types of supply chain entities, including FactoryAgent(FA), CustomerAgent(CA) and SupplierAgent(SA) are derived from SimuAgent to model the fabrication process, fulfillment process and procurement process respectively. The network relationship among agents is realized through registering one agent's event handler to the other's event dispatcher, e.g. the FA's order event handler has been registered with CA's order event dispatcher, and SA listens to FA's order event. Different SAs are instantiated to model the different lead time performance of the raw components from different type of suppliers. The demand distributions of the end products in different time period are maintained in the CA. In the processInternalTransEvent() method, CA creates order events according to the demand distributions and the current simulated time. The received orders are queued in the FA's unfilled order list, and will be processed in the processInternalTransEvent() method which implements the two stage fabrication/fulfillment process. In the fabrication process, all components are procured, built and tested ac-

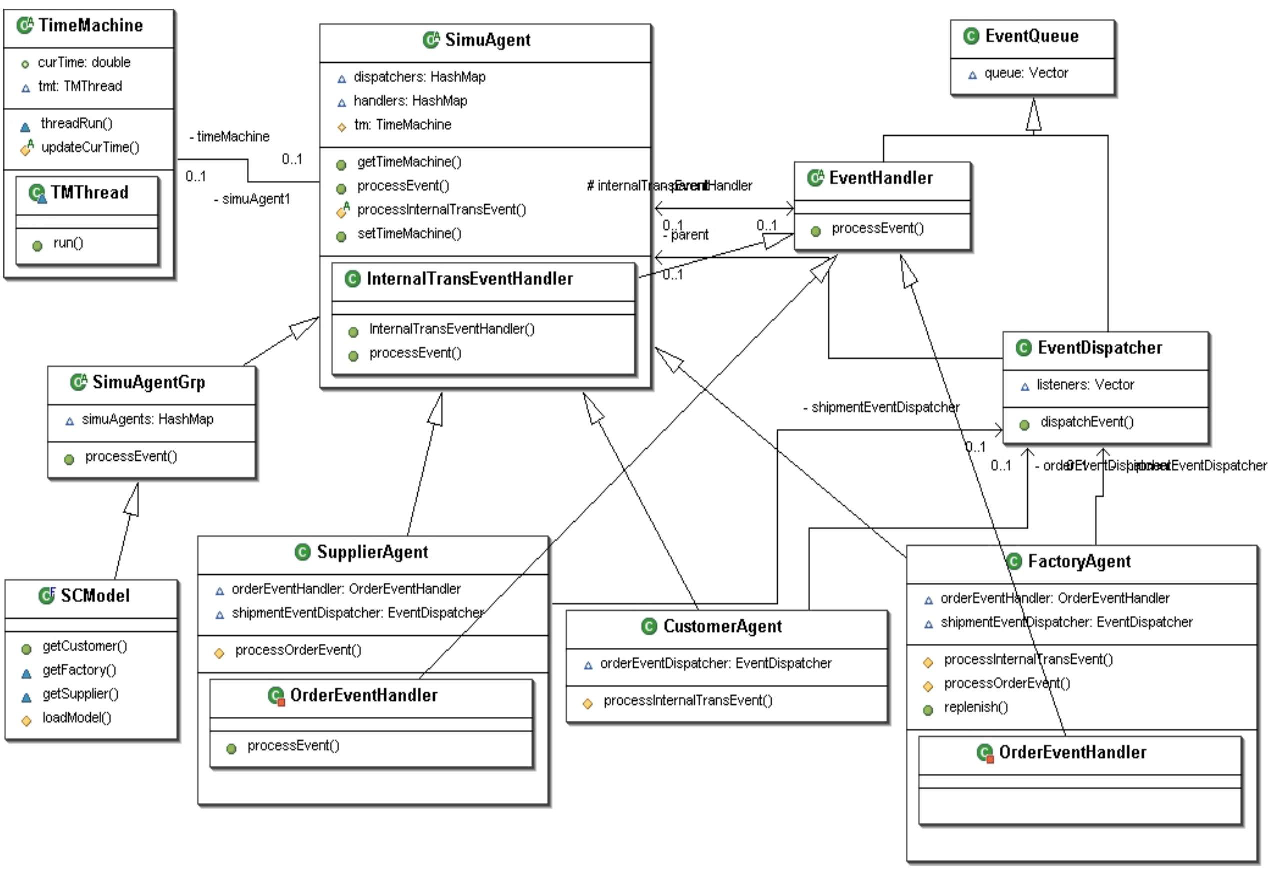

Figure 3: DOS Simulator UML Class Diagram

cording to a specified build plan. For those non leaf components, the actual quantities built are affected by the availability of the its subcomponents as defined in its BOMtree. Backlogged build quantity are assigned a higher priority, and will be fulfilled before regularly scheduled orders. After the fabrication process finishes, the fulfillment process starts the final assembly of unfilled orders according to the date they were received.

The report events from FA and CA are triggered at the end of each simulated period. We have defined two types of report event handlers, one that logs information into the database, and another one that outputs trace information to flat files for debugging purposes.

4 BUSINESS ANALYSIS AND APPLICATIONS

Generally, simulation tools have been used for business analysis at a strategic level. Using simulation to analyze and provide decision support at an operational level has not been widely practiced in industry. The need to create realistic models for complex manufacturing systems and to build live data links that provide the input data to a simulation engine is an extremely challenging task. In addition, the simulation tool needs to be very fast in order to provide meaningful results for tasks at an operational level.

The simulation tool described in this paper accomplished these challenging requirements for providing decision support at an operational level. The key functional tasks of the simulation tool include (1) providing the capability to project the future inventory performance for selected high-dollar parts in IBM Enterprise Sever Manufacturing; and (2) providing the capability to carry out scenario-based analysis which allows the decision makers to identify potential opportunities for performance improvement. Both tasks are performed at the operational level with modeling details including part-level information and the daily time buckets used for simulation.

4.1 Projecting Inventory Target Profiles

Given the product and manufacturing planning data, the tool uses the forecasts of future demands and estimated forecast accuracies to generate customer demand for end products, and compute the expected inventory levels over time at the part-level. The projected inventory levels are then converted to DOS profiles to provide a normalized measurement of inventory quantities. The simulation tool also provides a range for the projected inventory levels based on the confidence interval specified by the user. The projected inventory target profiles along with the ranges are used by the inventory planners as a comparison point to check against the actual inventory performance (see Figure 4).

The ability to project the future inventory performance is critical for IBM Enterprise Server Group since the inventory typically builds up early during the quarter as a result of highly skewed customer demand. Determining the right amount of inventory to accumulate is the key to managing inventory while providing a good serviceability to customers.

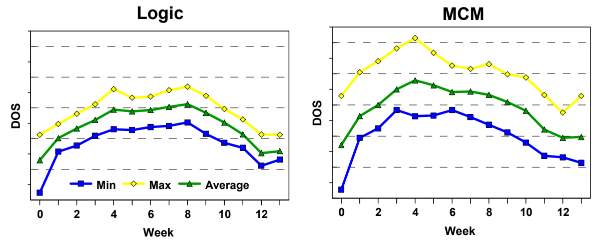

Figure 4: Simulated Inventory (DOS) Profiles for Parts of Selected Commodity Groups

4.2 Impact Assessment

Another important application of the tool is to provide the capability to perform what-if analysis. The user can define a number of scenarios with different supply chain parameters. By running the simulation and comparing the results obtained for different scenarios, the user can quickly identify the potential opportunities for improvement. In the example shown in Figure 5, we analyzed the effect of forecast accuracy improvement on inventory.

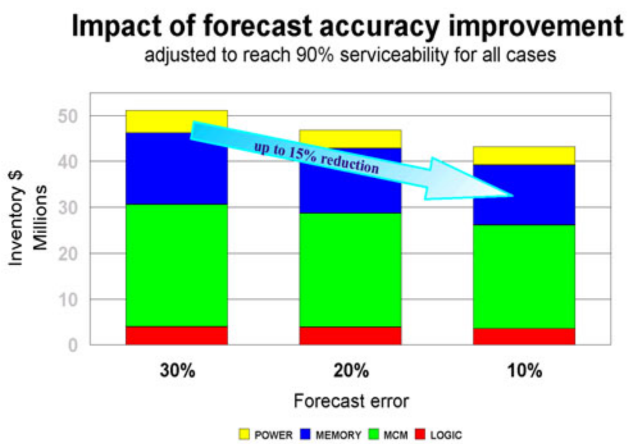

Figure 5: What-if Analysis on Forecast Accuracy

Similarly, the user can also evaluate the impact of other supply chain parameters, such as the spread ratio used for build plans, part yields, replenishment lead times, or manufacturing cycle times.

5 CONCLUSIONS AND FUTURE WORK

We have discussed our development of a distributed enterprise simulation system for inventory management in IBM Enterprise Server Group. This tool provides decision support at an operational level. An inventory planner is able to visualize the projected inventory level over time and also carrying out the impact access with different supply chain parameters.

We believe our component-based simulation design provides reusable software components for future extension and even other applications in business process modeling and simulation.

Our plan for future work includes developing a build plan optimization module and integrating it with the current web tool. Through this optimization module, a near optimal build plan is suggested to best utilize the available manufacturing capacity, and meet the service level agreement. we will also extend our simulator to model a more sophisticated manufacturing environment and multiple geographic manufacturing locations.

REFERENCES

Bagchi, S., S. Buckley, M. Ettl, G. Y. Lin. 1998. Experience Using the IBM Supply Chain Simulator. In *Proceedings of the 1998 Winter Simulation Conference*, ed. J. M. Charnes, D. M. Morrice, D. T. Brunner, and J. J. Swain, 65–72. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Banks, J. 1999. Introduction to Simulation. In *Proceedings of the 1999 Winter Simulation Conference*, ed. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G.W. Evans, 7–13. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Cao, H., G. Y. Lin, H. Xi and S. F. Smith, 2002. An Agent Based Enterprise Computing Framework for High Performance Supply Chain Simulation, In *Post-Conference proceedings of Int'l Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'02)*, Las Vegas, Nevada, USA, June 24-27.

Kim, D., H. Cao and S. Buckley. 2000. Modeling and Simulation of Supply Chain Management on Distributed DEVS Framework, in *Proceedings of AI, Simulation, and Planning 2000 Conference (AIS 2000)*, Arizona, March 2000.

Sun Microsystems, Inc. 1999. Delegation Event Model. Available online via <http://java.sun.com/products/jdk/1.1/docs/guide/awt/designspec/events.html>.

Swain, J. J. 2001. Simulation Software Survey. *ORMS Today*. February 2001.

AUTHOR BIOGRAPHIES

HENG CAO is a Staff Software Engineer at IBM T.J. Watson Research Center in Yorktown Heights, NY. She received her M.S. in Robotics from Carnegie Mellon University. In addition to modeling, analysis and simulation of supply chain systems, her interests include AI and data mining.

FENG CHENG is a Research Staff Member at IBM T.J. Watson Research Center in Yorktown Heights, NY. He received a Ph.D. degree in Operations Management from the University of Toronto. His current research interests include modeling and analysis of stochastic systems with applications in manufacturing systems, supply chain management, and e-commerce.

HAIFENG XI is an Advisory Software Engineer at the IBM T. J. Watson Research Center. He received his M.S. degree in Electrical and Computer Engineering from University of Maryland. His current interests include Web application architecture, business integration, grid computing, and supply chain simulation.

MARKUS ETTL is a Research Staff Member at the IBM Thomas J. Watson Research Center in Yorktown Heights, NY. He received M.S. and Ph.D. degrees from the Friedrich-Alexander University, Erlangen, Germany, in Computer Science. His research interests include adaptive supply chain management, business process modeling, simulation, and design of decision support systems for manufacturing logistics.

STEVE BUCKLEY manages the Analytic Models & Architecture department at the IBM T.J. Watson Research Center. He received the B.S. degree from Florida State University, the M.S. degree from Penn State University, and the Ph.D. degree from MIT, all in Computer Science. He has worked at IBM since 1978, and has research interests in simulation, supply chain management and business process management.

CARLOS S. RODRIGUEZ PMP, CPIM is the IBM Server Group Inventory Team Leader for Zseries and High End Pseries products. He received his B.S. degree from the State University of New York (Mathematics / Operations Research), and M.S. degree from Carnegie Mellon University (Industrial Administration). He is a Certified Project Management Professional (PMP) and APICs Certified in Production and Inventory Management (CPIM). His primary focus is Inventory and Supply Chain Optimization.