

USING INTERNET TECHNOLOGY FOR DESIGN OF FACILITIES AND MATERIAL HANDLING SYSTEMS

Douglas A. Bodner, T. Govindaraj, Eda Kemahlioglu,
Edgar E. Blanco, Marc Goetschalckx, Leon F. McGinnis and Gunter P. Sharp
Keck Virtual Factory Lab • School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205 USA
Email: douglas.bodner@isye.gatech.edu

ABSTRACT

Recent advances in internet technology have created opportunities to improve the state-of-the-art with respect to the computational tools available for design and analysis of facilities and material handling systems. For example, the internet facilitates remote experimentation and analysis of alternative designs. In addition, web-based client-server applications can support collaborative design and promote better management of model data. In this paper, we describe on-going research at Georgia Tech that leverages internet technology to create computational tools for facility and material handling system design. First, we describe a web-based simulation environment to support modeling and design of manufacturing and material handling systems. The environment is designed to facilitate structured analysis and experimentation of material handling systems. The second is an interactive, web-based application for warehouse design. It integrates a design database and heuristics to allow the user to make several design decisions involving configuration of storage areas in a warehouse, focusing on the forward pick area. This paper focuses on three important modeling issues that are exemplified in the applications presented. These include model representation to support tool integration, model representation to support reconfigurability, and design of databases to ensure data consistency. We conclude with a discussion of future modeling research.

1.0. INTRODUCTION

The rapid pace of technological change has important implications for manufacturing and material handling systems in the upcoming decade. New technologies are available for material processing and transport, for material tracking, for data management and for system modeling. For example, products in a factory can be tracked in real time as they progress through the system. Databases can capture detailed order and transaction histories. Powerful modeling tools such as simulation and optimization have become more commonly available, better suited to analysis of material handling systems, and easier to use. In addition, web-based applications bring analysis and querying power to remotely located computers.

At the same time, technological change has created new conditions in the marketplace. Consumers expect shorter lead times, greater product reliability and increased product customization. Product life cycles are made shorter by the obsolescence caused from technological innovation. Electronic commerce is changing the way in which companies conduct business. These trends have resulted in a competitive business environment, and have created challenges for the manufacturing systems that produce today's products.

To address the opportunities and challenges posed by these and other trends, the National Research Council (NRC) recently established a Committee on Visionary Manufacturing Challenges, which defined six "Grand Challenges" for research and development in manufacturing to be achieved by the year 2020 [National Research Council 1998]:

- concurrency in all operations,
- integration of human and technical resources,
- instantaneous transformation of information gathered from different sources into useful knowledge,
- reduction of production waste and environmental impact to near-zero,
- development of manufacturing operations that are reconfigurable to meet new needs,
- development of innovative processes and products with a focus on decreasing size.

In large part, these challenges share a common theme – creating and exploiting new information technologies to improve the effectiveness of the manufacturing enterprise.

Clearly, material handling systems have a major impact in most if not all of these focus areas, since material flow and its associated information flows tie together the manufacturing enterprise in many ways. In this paper, we concentrate on advances in modeling technologies that can be used to design and analyze material handling systems and industrial facilities. In particular, the focus is on the use of internet-based technologies and how they can be used for modeling and analysis. This type of research addresses several of the Grand Challenges posed by the Committee on Visionary Manufacturing Challenges, as shown in Table 1.

It is readily apparent that the worldwide web offers capabilities that can improve modeling and analysis of material handling systems. The main research questions relate to modeling – how to represent manufacturing and material handling systems to make models web-enabled, promote configurable models, facilitate integration of modeling tools, and ensure consistency of data.

The remainder of this paper is organized as follows. Section 2 discusses modeling and computational issues in material handling systems design and in facilities design with a focus on material flow. Section 3 describes a web-based simulation environment for analysis of material handling systems. Section 4 presents a web-based environment for configuring the forward pick area of a warehouse. We then conclude with a discussion of research issues and thoughts on future research.

Table 1. How internet-based modeling technologies can impact manufacturing challenges

Challenge	Relevant Enabling Technologies	Impact of this Type of Research
Concurrent manufacturing	<ul style="list-style-type: none"> • Enterprise modeling and simulation • Information technology • Collaborative software systems 	<ul style="list-style-type: none"> • Create methodologies and tools to support collaborative modeling, design and analysis.
Integrated human and technical resources	<ul style="list-style-type: none"> • Machine-human interfaces • Adaptable and reconfigurable systems • Enterprise modeling and simulation • Information technology • Education and training 	<ul style="list-style-type: none"> • Create capability for people to interact with complex models of manufacturing and material handling systems over the web. • Develop methodologies and tools for intelligent web-based training systems for material handling.
Converting information to knowledge	<ul style="list-style-type: none"> • Information technology • Enterprise modeling and simulation • Machine-human interfaces • Education and training 	<ul style="list-style-type: none"> • Provide computational tools and user interfaces that facilitate analysis of large datasets through a client-server system.
Reconfigurable enterprises	<ul style="list-style-type: none"> • Adaptable and reconfigurable systems • Enterprise modeling and simulation • Information technology • Improved design methodologies • Machine-human interfaces 	<ul style="list-style-type: none"> • Provide web-based capabilities for on-going analysis of new system designs or operational strategies to assess their effectiveness. • Create highly configurable models that facilitate experimentation.

2.0. MATERIAL HANDLING SYSTEM DESIGN

Design of facilities and material handling systems is a complex task. To date, there does not exist a comprehensive design methodology for these domains, although many useful models and analytic formulations have been developed to address specific parts of the overall design problem. Due to the quantitative and often data-intensive nature of these models, it is appealing to deploy them as computational tools that can aid designers and analysts. This section discusses research issues involved in developing computational tools and leveraging new internet-based technologies.

2.1. Computational tools and modeling issues

A variety of methodologies and computational tools currently are in use for design and analysis of material handling systems. Methodologies include analytic approaches, optimization, queueing network analysis and discrete-event simulation. Computational tools include spreadsheets, commercially available software packages (e.g., for optimization and simulation), research prototype software, and proprietary software developed for in-house use by a particular company. Typically, computational tools and their accompanying models are deployed as applications residing on a single computer.

One trend evident in research and modeling practice is the integration of models and methodologies to support a wider scope of analysis. For example, Peters and Yang [1997] develop an integrated computational approach to designing material handling systems and layouts in semiconductor manufacturing. Another trend is the design of models to be configurable. For instance, Sreekanth *et al.* [1993] propose a graphical user interface that can be used to specify behavioral representations in simulation modeling. Many commercially available simulation packages now feature graphical front-ends that allow a modeler to configure a model with minimal programming. When these types of front-ends are used, it is important to note that the underlying abstractions provided by the modeling software are critical in determining the quality of the model. Finally, management of model data has become a major issue, due to the increasingly data-intensive nature of modeling. There is a trend toward integrating spreadsheets and databases with modeling packages to manage model data, rather than relying on text files or data embedded in a model. This type of approach promotes modularity and also configurability, since analysis using different datasets can easily be performed, for example, by interchanging spreadsheet files.

2.2. New technologies, tools and opportunities

The advent of the worldwide web has created a new way for distributing and manipulating information. Clearly, the technologies used in business web applications have potential application in engineering design and analysis. These technologies range from databases, to web and application servers, to programming languages and communication protocols. Table 2 summarizes the different technologies relevant to this discussion and presents examples of each.

At the core of a complex web-based computing application is the application server. The application server uses databases to manage information about a system being designed or modeled, as well as information about sessions or scenarios that are in use or that have been archived by a user. Commonly available web browsers provide an interface for users. Using appropriate protocols for authentication and encryption, the application can provide secure transactions, depending on the need. Web pages are state-dependent and are generated dynamically. Generally, software is developed in an object-oriented language such as Java™, along with one or more scripting languages. The application server provides the means for system calls, which can be used, for example, to generate a simulation model from available data, execute an experiment, and report back the results.

Properly used and integrated, these technologies provide a platform for developing and deploying web-based computational tools. In the next two sections, we discuss prototype applications that have been developed for material handling system and facilities analysis.

Table 2. Tools and technologies

Function	Key Characteristics	Examples
Databases	<ul style="list-style-type: none"> • Relational and/or object-oriented databases • JDBC or ODBC interfaces and drivers for specific application servers 	DB2™, FrontBase™, Informix™, Oracle™, MySQL™, PostgreSQL™, SQL Server™, Sybase™, Versant™
Web servers	<ul style="list-style-type: none"> • Web servers with options for secure transactions 	Apache™, Jigsaw™, Internet Information System™, Java Web Server™
Application servers	<ul style="list-style-type: none"> • Software for development and deployment of dynamic websites • Often used with integrated development environments (IDEs) • Forms a middle tier in a multi-tier environment of software components • Other tiers and components include databases, web servers and web clients 	Enhydra™, GemStone™, Sapphire/Web™, SilverStream™, WebLogic™, WebObjects™, WebSphere™
Communications protocols and interfaces	<ul style="list-style-type: none"> • Software interfaces between different components in a distributed computing environment 	CORBA (Common Object Request Broker Architecture), DCOM™, OMG Interface Definition Language
Scripting languages	<ul style="list-style-type: none"> • Scripting for web pages • Client/server side • Mostly object-oriented 	Perl, PHP, WebScript™, ASP (Active Server Pages™), ColdFusion™, JSP (Java Server Pages™)

3.0. WEB-BASED SIMULATION OF MATERIAL HANDLING SYSTEMS

There has been growing interest in the use of the worldwide web as a platform for simulation of industrial and logistics systems (e.g., Dessouky *et al.* [1998], Du [2000], Narayanan *et al.* [1998], Smith [1997]). The notion of a simulation model that is available over the web is appealing for a variety of reasons. First and perhaps foremost, it makes simulation technology more accessible to a variety of users. These might be managers wishing to perform analysis, or manufacturing engineers wishing to decide between different operational strategies on the shop floor. In addition, web-based simulation models would allow one to store models and data on a server, where they could be made more secure and could be better managed (e.g., preventing data redundancy or inconsistency).

The idea behind web-based simulation is to host a simulation engine, as well as simulation models and data, on a server. Then users would access the simulation models, configure them, and run experiments on them. In this section, we discuss issues involved and present a prototype application. The goal is to create configurable models that are

amenable to experimentation; hence underlying system and data representations are important. This work is demonstrated via a case study in semiconductor fabrication and is implemented using the Common Object Request Broker Architecture (CORBA), Java and ARENA™.

3.1. Technology and Research Issues

There are two sets of issues associated with web-based simulation. The first set relates to the types of technologies to use in creating web-based simulation capability. The technologies selected should provide the following functionalities:

- allow users to access the simulation model using a thin-client interface (through web browsers and network programming languages such as Java);
- provide capability for users to configure and execute a simulation model (through user interfaces);
- provide users with output statistics from a simulation experiment (i.e., converting information to knowledge); and
- manage multiple concurrent users (through communication protocols, application servers and databases).

In the last item, our focus here is on multiple independent users. Extending this, though, we could envision multiple users and simulations that interact with one another to model, for example, an entire supply chain.

The second set of issues, which is more in line with material handling research, addresses how one models a system to support web-based simulation capability. Here, we are interested in model configurability, or the degree to which a user can easily specify characteristics of the system being studied for experimental purposes (e.g., number of machines, number of transporters, routing parameters, even control logic). One way to achieve this involves the specification of generic code fragments for elements of the simulation model (e.g., machines). A simulation code generator can then be used to assemble a model from components (e.g., Gong and McGinnis [1990]).

3.2. A Web-Based Simulation Application

To address the requirements above, we have developed a prototype application that integrates several internet and other technologies. The simulation engine uses ARENA, a commercially available simulation package from Systems Modeling. The user configures an ARENA simulation model via Java applets that allow entering of model parameters. A Java applet is a program downloaded from the server that runs on the client computer. The user needs only a Java-enabled web browser to access the simulation. Communication between the client and the server is handled through a CORBA interface. CORBA provides a standard protocol for software objects and components to communicate and exchange data over the web [Orfali and Harkey 1998]. As such, it allows a client application to send configuration information to a server that hosts a

simulation model, and it also allows the server application to send model output to the client. The CORBA code actually resides on the server. To provide this functionality, CORBA was selected from several alternatives, including Java sockets and Java remote method invocation (RMI), because it provides support for different platforms and for languages other than Java [Narayanan *et al.* 1998]. The application uses the ORBacus implementation of CORBA, which is available from Object-Oriented Concepts, Inc. Figure 1 shows the architecture for the application.

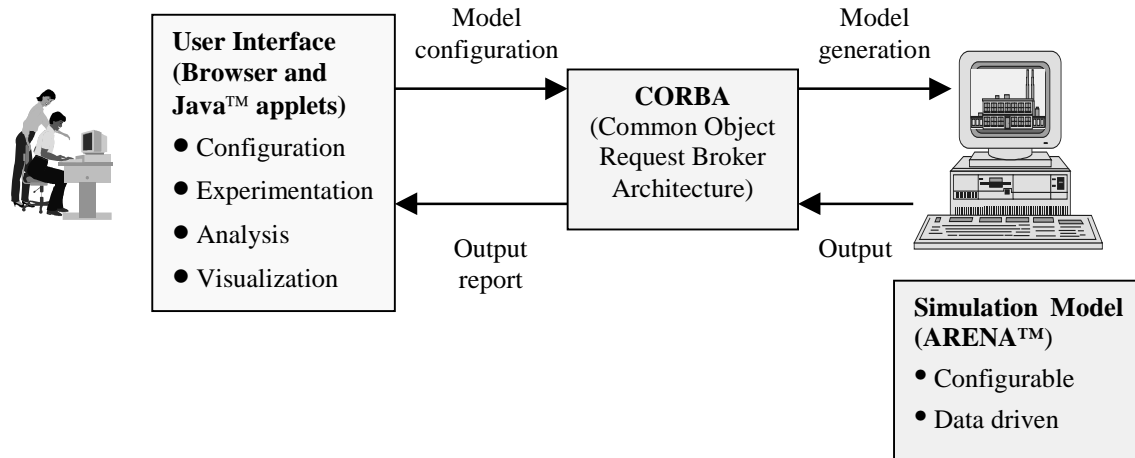


Figure 1. Web-based simulation architecture

3.3. Example

As a proof-of-concept, a simulation model of a bay in a semiconductor fabrication system (i.e., a fab) has been implemented. This model is similar to Kempf's data specification for a five-machine bay [Kempf 1994]. One major difference is that the simulation model includes an automated material handling system. Kempf's specification does not explicitly consider material handling. For this application, we consider a single-loop rail-guided vehicle system. Such automated material handling systems are predicted to become more commonplace in semiconductor manufacturing, as manufacturers transition to process technology to support production of 300mm wafers, instead of the current wafer size of 200mm. Industry is pushing toward 300mm wafer sizes to improve productivity (i.e., larger wafer sizes yield more semiconductor chips using the same number of operations). At the same time, 300mm fabs require extensively automated material handling, since human operators cannot safely carry cassettes containing wafers of this size.

At present, the model is fairly simple, consisting of a single loop system with process tools at various locations and an automated storage/retrieval system that houses incoming and outgoing cassettes of wafers. The model is parameterized so that the user may specify the following system characteristics:

- Number of part types;
- Arrival stream characteristics of parts (part mix and interarrival time distributions);
- Specification of processing tools (number of stations, number of tools at each station, and buffer capacity at each station);
- Process plans of each part type (sequence of process tools to visit and operation times and distributions);
- Configuration of the guidepath loop (length of the loop and position of each station);
- Number of vehicles; and
- Vehicle characteristics (load and unload times, speed and idling policy).

To run the model, the user first accesses the web site. The web site then presents him or her with a series of Java applets, which are used to configure the simulation model. After the user configures the model, the next step is to register the model. The application keeps track of concurrent user sessions by having each user register under a unique identification using an internal tracking scheme. Each user's model is stored separately, and user requests for model execution are performed sequentially. After the model is registered, the user initiates creation of the ARENA model and experiment files by the server. The server compiles and links these files using the ARENA scenario manager, and then it runs the simulation. After the simulation has finished executing, the user can request the ARENA output report.

The current application is a good example of how web-based technologies can be used to make simulation more widely available. In addition, it presents a user with intuitive interfaces that can be used to configure the model. Hence, a simulation novice could run a simulation experiment, using structured experimentation and analysis to determine the effects on cycle times or work-in-process inventory of adding additional vehicles, changing vehicle speeds, adding processing tools, changing process times, etc. It would be desirable to enhance the basic model so that the user can configure the following additional items:

- more complex guidepath networks,
- processing tool behaviors (e.g., batching, setups, failures),
- vehicle routing and dispatching policies, and
- job routing and dispatching policies.

Work toward these goals currently is underway in the Keck Lab at Georgia Tech. The eventual goal is to create a high-fidelity, configurable, web-based simulation environment for semiconductor manufacturing [Park *et al.* 2000]. This will require careful design of the underlying abstractions used by the simulation model, including explicit modeling of the control logic used by the material handling system.

4.0. WEB-BASED WAREHOUSE DESIGN

Consider the problem of designing a warehouse. Typical design issues include high-level design (e.g., determining which departments to have, aggregate material flow networks between them, and automation levels), material handling system design, layout, selection of storage technologies, storage system configuration and specification of labor. Here, we consider a subset of the overall design problem, namely the forward-reserve problem as described by Frazelle *et al.* [1994]. In this problem, the designer must design a forward pick area and assign products, or stock keeping units (SKUs) to it. A forward pick area is used to enable order filling of certain items in a warehouse, by using storage technologies that facilitate rapid picking of SKUs. For example, a forward pick area might use gravity flow racks to store SKUs as individual items or in cases, so that operators can quickly pick them. From there, they are moved to an order sortation and accumulation area. A reserve storage area, on the other hand, is designed to promote storage efficiency of many SKUs. Typical storage technologies include stacking pallets on the floor (blockstacking) or using racks for pallet storage.

4.1. Design Decisions and Cost Analysis

In the forward-reserve problem, we assume that the designer is provided with the square footage available in the warehouse for storage. One design decision, then, is to determine the amount of this area to allocate to the forward pick area versus the reserve storage area. In addition, the designer selects storage technologies for SKUs in the reserve area and in the forward pick area, and he or she assigns space to products in the forward area. A warehouse manager generally wants to place high-demand products in the forward area, to have the largest impact on reducing material handling time in order filling (and hence labor costs for order filling).

However, placing a large number of SKUs or large amount of inventory in the forward area has two disadvantages. First, by increasing the amount of space required in the forward area, it reduces the amount of space available for reserve storage. Generally, reserve storage is more space-efficient. Hence, a large forward area reduces space efficiency in the warehouse. Second, creation of a forward pick area entails creation of a replenishment task, since the forward pick area is restocked from reserve storage when its inventory levels become low. Having many SKUs in a forward pick area increases labor costs by increasing the need for these replenishments. Designing a forward pick area, therefore, has a number of trade-offs associated with it.

Frazelle *et al.* [1994] consider the following costs in their model for designing a forward pick area:

- Capital cost of storage equipment,
- Labor cost of replenishing the forward area from the reserve area, and
- Labor cost of order picking.

They also propose heuristics to calculate estimates for these costs. This model is used as the basis for a web-based design environment that allows a user to make and evaluate design decisions for a forward pick area.

4.2. Web-Based Design Environment

A prototype web-based design environment has been developed to demonstrate web-based warehouse design capability. The initial prototype was developed using Perl (a scripting language), with databases implemented in PostgreSQL, a freely available object-relational database. It uses the Sun Web Server™ and runs on a Solaris™ workstation.

The databases contain information about a given scenario that is under consideration for design. This information includes the following:

- General information (warehouse size, maximum lane depths allowed, labor costs);
- Product information for SKUs (unit load characteristics, stackability, days of inventory stored);
- Demand profile for SKUs (demand per time period, picks per time period);
- Storage equipment characteristics (lane depths, dimensions, aisle widths); and
- Storage equipment efficiency (cost per square foot, replenishment rate, throughput).

Using this information, the designer makes several design decisions. First, he or she can determine the storage technology to be used for pallets in the reserve area, choosing from blockstacking, single-deep racks or double-deep racks. Second, the designer can allocate a percentage of the available storage space to the forward area. Finally, the designer can decide how much space each SKU is allocated in the forward area. Then the designer can cost his or her design using cost heuristics that are part of the application. Several scenarios are implemented currently, including a computer parts distribution center, an apparel warehouse and a grocery warehouse. Figure 2 illustrates the architecture of the design environment.

Current work consists of migrating it to a more robust and maintainable implementation using WebObjects™, an application server from Apple. WebObjects runs on multiple platforms, including Windows NT™, Unix (Solaris and HP-UX™) and MacOS X Server™.

4.3. Discussion

The design environment illustrates use of a server-based design database to manage data about a system under analysis. In addition, this database is used in managing user sessions. Using a secure login, a user can log into the design environment, make decisions and evaluate them, then log out without committing to a final design. The

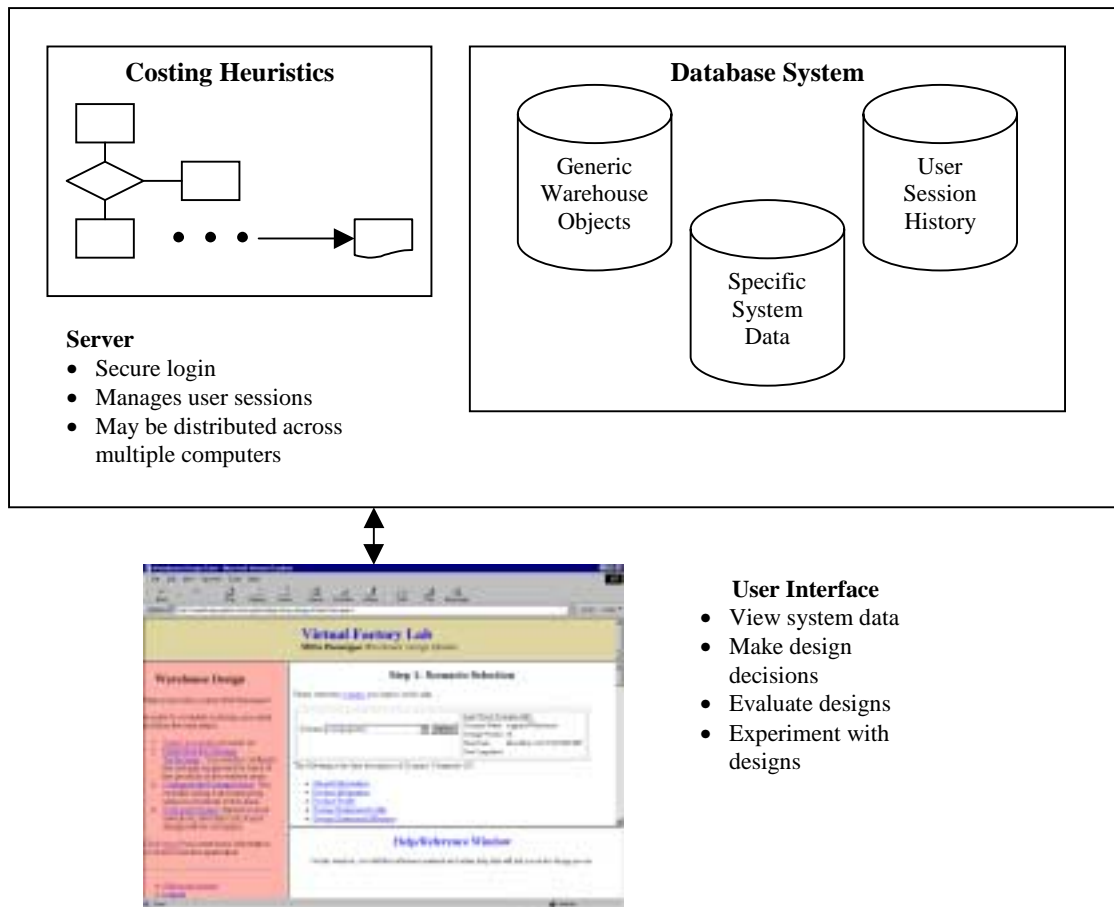


Figure 2. Architecture for design environment

design-in-progress is stored for later analysis by that user under his or her login identification. When a design is complete, the user can commit it to the archive.

The design of the database is an important issue. While its primary use in this application is to support analysis of the forward-reserve problem, the database used in the web-based design environment actually is based on an object model created to support the overall warehouse design problem [Goetschalckx *et al.* 2000]. The goal of the object model is to serve as a unifying data structure for the warehouse design problem. In implementation, it takes the form of a design database that captures information necessary for a design, such as layout geometry and zones, material flows between departments, storage system characteristics, material handling equipment characteristics, product and container information, order and inventory histories, labor requirements and standards, etc. In addition, it is designed so that constraints and interrelationships are captured. For example, a particular material handling equipment imposes constraints on aisle widths. A particular unit-load container can be handled only by certain types of material handling and storage equipment. These types of relationships must be captured in a design database to ensure that the design does not contain inconsistencies.

The ultimate goal is the creation of a web-based warehouse design workstation that integrates a number of analysis tools. These might include such things as heuristics contained in the existing design environment, optimization routines for layout and design, statistical analysis tools for partitioning SKUs into different categories, visualization of data and three dimensional walkthroughs, and even simulation of dynamic warehouse behavior to assess design effectiveness. The object model and design database would serve as the integration platform, providing a common data structure for all these tools. Further, it would be desirable to design this application so that it facilitates collaboration among multiple designers. As envisioned, this web-based design workstation could have a number of uses in research and education, and would move industrial practice closer to the goals set by the NRC in terms of concurrent manufacturing/distribution and reconfigurable enterprises.

5.0. DISCUSSION AND FUTURE RESEARCH

The two applications presented demonstrate the usefulness of many of today's available internet technologies, applied to the modeling of material handling systems and industrial facilities. In creating them, there are significant research challenges in terms of modeling, apart from requirements to make them web-enabled. Particular challenges include:

- model specification to promote configurable models that are amenable to experimentation by users,
- design of database systems and object models to manage complex datasets and ensure data consistency, and
- design of databases and object models to facilitate integration of modeling tools and methodologies through the existence of a common and consistent data structure.

While we have made progress, there is much work ahead. Our research group at the Keck Virtual Factory Lab is actively engaged in addressing these research issues and in creating capabilities for web-based modeling and analysis of industrial systems.

One can ask, how are web-based modeling technologies likely to impact design and analysis of material handling systems in general? There are several answers to this question. First, by making modeling technology more accessible, a wider population of users can be created. This is especially true if the user interface tools are powerful enough to translate between the way in which a manager or a manufacturing engineer conceptualizes a system, and the way in which that system is represented in an abstract model. Second, improved access also impacts existing users. At times, these users might need to perform some analysis, but also might be in situations where they do not have direct access to a computer with the models that they need. Having access to server-based models would improve their ability to perform analysis "in the field." Finally, by housing data on a server, data management is likely to be improved, which is important

in today's increasingly data-intensive world. This would occur in a number of ways, ranging, for example, from providing server-based security for sensitive equipment or production data, to providing database management tools to handle complexity and ensure against redundancy. These types of impacts relate directly back to many of the challenges posed to the manufacturing research community by NRC.

There is at least one caveat. A major concern with web-based modeling relates to validation. A user who interacts with a model over the web could well question its validity, as the result of at least two causes. There may be limited technical information about the model itself available to the user. Further, since the model is housed on a server remote from the user, there may be psychological factors that cause the user to "trust" the model less than he or she would if the model were on a local computer. One interesting approach to addressing this is proposed by Davis [1998], who envisions that equipment makers may in the future provide detailed models of their equipment to potential customers, which can be downloaded from a web site for testing in a model of the customer's proposed factory. Research seeking to develop generic models of equipment behavior is likely to benefit validation methodologies, as well. Currently, this work focuses on processing equipment (e.g., Du [2000]), but future work might address material handling. If either of these approaches yields a way to "certify" model validity, it would be a tremendous aid to modeling in general, not just to web-based modeling.

6.0. ACKNOWLEDGMENTS

The authors wish to acknowledge the W. M. Keck Foundation for its funding of the facilities and activities of the Keck Virtual Factory Lab at Georgia Tech. In addition, this work has been partially funded by UPS Worldwide Logistics, by the Manufacturing Research Center at Georgia Tech, and by the National Science Foundation under Grant No. DUE 9950301.

7.0. REFERENCES

- [1] Davis, W. J. "Looking into the Future of Simulation," *IIE Solutions*, Vol. 30, No. 5, pp. 24-30, 1998.
- [2] Dessouky, M. M., D. E. Bailey, S. Verma, S. Adiga, G. A. Bekey and E. J. Kazlauskas. "A Virtual Factory Teaching System in Support of Manufacturing Education," *Journal of Engineering Education*, pp. 459-467, 1998.
- [3] Du, Xiaohua. *Equipment Controller Design for Resource Monitoring and Supervisory Control*, Ph.D. dissertation, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 2000.
- [4] Fowler, M. and K. Scott. *UML Distilled, Second Edition*, Reading, MA: Addison-Wesley, 2000.

- [5] Frazelle, E. H., S. T. Hackman, U. Passy and L. K. Platzman. "The Forward-Reserve Problem," in *Optimization in Industry 2*, T. A. Ciriani and R. C. Leachman (eds.), New York: John Wiley & Sons, 1994.
- [6] Goetschalckx, M., M. M. Amirhosseini, D. A. Bodner, T. Govindaraj, L. F. McGinnis and G. P. Sharp. "Object Models and Design Databases for Warehouse Applications," in *Proceedings of Solutions 2000*, Atlanta, GA: Institute of Industrial Engineers, to appear, 2000.
- [7] Gong, D.-C. and L. F. McGinnis. "An AGVS Simulation Code Generation for Manufacturing Applications," in *Proceedings of the 1990 Winter Simulation Conference*, Piscataway, NJ: Institute of Electrical and Electronics Engineers, pp. 676-682, 1990.
- [8] Kempf, K. "Intel Five-Machine Six Step Mini-Fab Description," Intel/ASU Report, Arizona State University, 1994.
- [9] Narayanan, S., P. Malu, A. P. B. Krishna, J. DiPasquale and T. M. Carrico. "A Web-Based Interactive Simulation Architecture for Airbase Logistics Systems Analysis," *International Journal of Industrial Engineering*, Vol. 5, No. 4, pp. 324-335, 1998.
- [10] National Research Council, *Visionary Manufacturing Challenges for 2020*, Washington, DC: National Academy Press, 1998.
- [11] Orfali, R. and D. Harkey. *Client/Server Programming with Java and CORBA, Second Edition*, New York: J. Wiley & Sons, 1998.
- [12] Park, J., S. A. Reveliotis, D. A. Bodner, C. Zhou, J.-F. Wu and L. F. McGinnis. "High Fidelity Rapid Prototyping of the Real-Time Fab Operations through Discrete Event System Modeling," in *Proceedings of the International Conference on Modeling and Analysis of Semiconductor Manufacturing*, Tempe, AZ: Arizona State University, to appear, 2000.
- [13] Peters, B. and T. Yang. "Integrated Facility Layout and Material Handling System Design in Semiconductor Fabrication Facilities," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 10, No. 3, pp. 360-369, 1997.
- [14] Smith, J. S. "Simulation over the Web: An FMS Design Exercise Using Discrete Event Simulation," <http://mfg.tamu.edu/fmssim/>, 1997.
- [15] Sreekanth, U., S. Narayanan, D. A. Bodner, T. Govindaraj and C. M. Mitchell. "A Specification Environment for Configuring a Discrete-Part Manufacturing System Simulation Infrastructure," *Proceedings of the 1993 IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 1, pp. 349-354, 1993.