Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

TRADEOFFS IN BUILDING A GENERIC SUPPLY CHAIN SIMULATION CAPABILITY

Sanjay Jain

The George Washington University
2201 G Street NW, Suite 415
Washington, DC 20052, USA

ABSTRACT

Building a simulation model for any large complex system requires high expertise and effort. These requirements can be reduced through building generic simulation capability that includes artifacts for facilitating the development of the simulation model. The artifacts can have a range of capabilities depending on the design goals for the simulation. This paper focuses on issues to be considered in building a generic simulation capability for supply chains. A number of approaches used in recent years for building generic supply chain simulation capability are discussed. Such approaches include data-driven simulators, interactive simulators, and sub-models for supply chain components. Tradeoffs are identified that should be considered in selecting an approach for building a generic supply chain simulation capability.

1 INTRODUCTION

The use of discrete event simulation for analyzing supply chain designs and operations has been increasing. This is evident by the increasing number of publications on the topic in recent time. Industrial supply chain are complex systems. A large effort and expertise is required for building models of supply chains using general purpose simulation software. Over the last decade, a number of efforts have focused on facilitating the process of building simulation models of supply chains. Some efforts have appeared as commercial solutions while many others are continually reported by academia and research organizations. The general intent of these efforts has been to reduce the requirement for dealing with complexities of a general purpose simulation software used for building and updating models of supply chains. The capabilities developed using these efforts allow people with supply chain domain knowledge to take a larger role in building, updating, and using the models of their supply chains and indeed be able to do so fully on their own in some cases.

The approaches for facilitating the process of building supply chain simulation models include specific language for modeling a supply chain, sub-models of supply chain nodes, templates, and different types of simulators. These approaches vary in their expectation of expertise required for building models of supply chains. They correspondingly vary in other dimensions such as the flexibility available in such things as supply chain configurations and operational policies that can be modeled.

This paper identifies the various approaches that have been reported in recent years to create generic supply chain simulation capabilities. The paper does not intend to exhaustively review the literature for generic supply chain simulation capability developments. The intent of the paper is to identify a number of different approaches and provide a comparison among them. The comparison helps in judging the nature of these approaches with respect to the flexibility they offer and the effort they require for building simulation models of subject supply chains. The comparison also helps determine the tradeoffs involved that should be considered when either selecting or building a generic supply chain simulation capability.

The next section identifies the alternative approaches as represented by efforts reported in the literature. This is followed by the comparison of approaches for developing generic supply chain simulation capabilities provided in Section 3. Section 4 discusses the involved tradeoffs and the last section concludes the paper.

2 ALTERNATIVE APPROACHES

The approaches for building generic capabilities for supply chain simulation have evolved over the years. This evolution has been driven by multiple factors including maturity of software technologies, simulation software, supply chain management knowledge and practices, and simulation model design and implementation practices in the industry.

2.1 Data-Driven Simulators

Early efforts to make the process of building supply chain simulation models easier involved use of spreadsheet interfaces or templates coupled with a generic supply chain simulation model. A data-driven simulator may be defined as a simulation model that can be completely paramete-

978-1-4244-2708-6/08/$25.00 ©2008 IEEE

rized by providing data through a set of data forms, tables, spreadsheets, or templates and is designed specifically for modeling an identified set of systems. The model is thus "developed" by providing data on the subject system. Benson (1997) demonstrated an early data-driven simulator for modeling supply chains built using the template feature of ProModel. The template provided data input tables for a user to describe the supply chain.

The early supply chain simulators took after the earlier developed manufacturing simulators. The simulation software company AutoSimulations (now known as Brooks Software) had a product by the name "Simulator" that was described as a manufacturing template that could be used to model a certain class of manufacturing systems (Rohrer 1999). The simulator consisted of a number of data tables that an analyst could populate to configure a model of the manufacturing system.

The data-driven simulator approach is easiest for the user for configuring a generic model. Model building using a data-driven simulator requires entering the data on the supply chain including nodes, production rates, logistics duration, and customer order rates. The approach however provides little flexibility to include a configuration or practice that is not comprehended by the options provided by the template. For example, the inventory replenishment policies may be limited to reorder up to a defined level only, or to a few such predefined policies for reordering. The data-driven simulators thus provide a quick option for building supply chain models. The generated model would be an accurate representation if it is completely within the scope of the configurations provided for by the simulator. Otherwise, it will provide the proverbial "quick and dirty" model.

A perhaps more critical limitation of the data-driven simulators is the inability to configure the graphic representation of the supply chain. A graphic representation if available will be in a standard configuration provided by the simulator developers. In theory, the data templates can allow defining coordinates for placing supply chain nodes on the screen. However, this is rather tedious and that may be the reason such a feature was not found reported for the data-driven simulators for supply chain simulation.

Data-driven simulators are typically implemented in the software through a compact code with modules for components of supply chain such as production node, distribution node, or logistics link. The same code module is executed over and over by entities carrying parameters describing different components of the same type. For example, a code module for modeling production nodes may be executed multiple times by different entities and parameterized using the attributes carried by the entity to model various production nodes in the supply chain. Such code usually gets quite complex as it provides for multiple options within one compact module. However, the user is shielded from the complexity of the code since interaction with the system through the data template interface.

Chatfield, Harrison, and Hayya (2006) describe a data-driven simulator. The user describes the supply chain data in a file using the eXtensible Markup Language (XML) standard (ISO 2002). The XML file can be used by various simulation packages that may provide an appropriate parser and features to model the supply chain. A prototype simulator, Simulator for Integrated Supply Chain Operations (SISCO), is developed and used by the authors to generate supply chain models based on the XML file description. Cope et al. (2007) also use an XML file editor to generate a supply chain description. They transform the file using the eXtensible Stylesheet Language Transformation (XSLT) standard (W3C, 1999). The transformed file is used by Visual Basic routines and the simulation software Arena (Kelton, 2006) to generate a model of the supply chain. They developed supply chain modules corresponding to the Supply Chain Operations Reference (SCOR) framework using Arena template features. These modules are automatically assembled to match the supply chain description provided in the XML file.

2.2 Interactive Simulators

An interactive simulator may be defined as a simulation software that allows interactive development of a model for an identified set of systems using specifically designed modules. The interactive simulators for modeling supply chains provide modules for representing supply chain components such as supplier, manufacturer, retailer, and distribution center. These modules can be interactively assembled by the supply chain analyst in configurations representing the logistics networks to develop supply chain models. Data screens or tables associated with the modules can then be populated to parameterize the model.

The interactive interface allows development of the supply chain network configuration. This allows involving the supply chain domain experts in developing the model and arriving at a graphic version of the model that is representative of the geographic network. Most of the simulation software applications allow importing maps as background and thus achieving a good visual representation of the real life supply chain.

It may be noted that modules defined in this context are usually not objects as defined in the discipline of object oriented programming (OOP) but they serve a somewhat similar role. Most of the discrete event simulation software applications provide a menu of constructs that can be viewed as pseudo-objects. These constructs are variously named as blocks, processes, modules, etc. Instances of pseudo-objects are assembled in a logic network to develop a model. Some simulation software applications, such as Plant Simulation (earlier name, eM-Plant) from Tecnomatix, do provide objects as defined in the OOP paradigm.

The data-driven simulator for modeling supply chains from ProModel mentioned earlier led to the development of an interactive simulator, Supply Chain Guru, brought to market by Llama-Soft (Hicks 1999).

IBM's Supply Chain Simulator (later renamed to Supply Chain Analyzer) was among the early interactive simulators per its description by Bagchi et al. (1998). It was initially developed to support IBM's consulting practice and later offered as a commercial product. The task of building models using an interactive simulator can be further facilitated by providing sample models that can be modified by the user. Chen et al. (1999) describe a web-accessible version of the IBM's supply chain analyzer that includes industry specific template models to assist the user.

Pathak, Dilts, and Biswas (2003) use an agent-based implementation using Madkit, a Java-based agent platform to develop the interactive supply chain simulator CAESAR (Complex Adaptive Supply Chain Simulator). The simulator analyzes the growth of supply chains. An evaluation engine and a visualization engine have been developed to record and analyze the growth.

A simulator, data-driven or interactive, is generally built on top of a general purpose simulation software application but it may be built ground up for the identified purpose. Simulators appear as standalone software and may or may not allow an interface to the programming capabilities of the base software for building custom features. Some of the simulators provide "user exits" or other similarly named features for allowing users to utilize base software capabilities and build custom code.

2.3 Simulator Extensions

A simulator extension is defined as a set of modules provided in general purpose simulation software for ease of modeling an identified set of systems. While the simulator extensions are quite similar to interactive simulators described in the previous sub-section, the key difference is that the user has direct access to all the features of the base simulation software. Hence, a user can mix the modules provided within the simulator extensions with those provided in the base simulation software for developing the model. Such an option provides quite a bit of flexibility to a model developer albeit with the requirement of increased expertise.

Some supply chain simulator extensions appeared near the end of the last millennium. Some of these simulator extensions were developed by organizations for modeling their internal supply chains. The environment with the focus on modeling internal supply chains with a limited set of configurations and practices common within the organization was well suited for development of simulators. Compaq Corporation, a manufacturer of personal computers, developed CSCAT, a simulator to model its internal supply chains using modules built in Arena (Ingalls and Kasales 1999). The development appears to be the precursor for the templates feature of Arena used by later efforts to develop simulator extensions.

Lian and Landeghem (2002) utilized Arena templates for building models of supply chains for evaluating lean manufacturing benefits. They define templates for customer, processing, super-market and flow-line for building models of supply chains. The approach is presented for supporting the simulation of lean manufacturing examples and not as a simulator extension for generic supply chain simulation. The templates are indeed rather customized to the described scenarios. Wang and Takakuwa (2006) utilize modules developed using Arena template feature for developing a simulator extension for supply chain simulation. Modules are provided corresponding to supply chain components including Factory, Warehouse, Distributor, Retailer, and Products. A user can drag the modules in to the model window and connect them in a configuration representing the logistics network to create a model of the supply chain.

The Supply Chain Builder from Simulation Dynamics is a simulator extension built on top of the general purpose simulation software, Extend+Industry (Phelps, Parsons, and Siprelle 2001). The builder provides a library of blocks that can be connected to represent the supply chain. General purpose constructs from Extend+Industry can be used as needed to include unique features in the supply chain model. Merkuryev et al. (2008) describe a simulator extension to Enterprise Dynamics simulation software for supply chain modeling in the ECLIPS project. Enterprise Dynamics "Atoms," or modules, are developed to represent supply chain artifacts.

2.4 Sub-Models

The sub-model approach provides a generic version of hierarchical models that can be replicated and assembled as needed to represent a system. The primary differences between providing sub-models and modules in an interactive simulator are in the level of detail and the access to the code. Providing sub-models at multiple levels of detail will allow a model developer to understand and select the closest representation to the situation being modeled. This flexibility again comes with the requirement of increased expertise in simulation software.

The access to the code may vary among different implementations. Implementation of templates in Arena, used by a few efforts for developing interactive simulators, does not allow access to internal code of modules and thus would not allow access to details. The template developer can provide access to the module code if desired.

Vieira (2004) proposed using generic sub-models at multiple levels of hierarchy for use in developing supply chain simulation models. The first level sub-models include suppliers and manufacturers while second level submodels include processes such as component ordering and product manufacturing.

2.5 Object Class Libraries

Another approach for providing generic simulation capabilities for an identified domain is to develop a library of objects representing artifacts and/or processes in the domain. Simulation analysts can utilize the objects to develop models quicker than using the base language. This approach is quite similar to simulator extensions of general purpose software, except that these are used with a simulation or general programming languages.

Verbraeck and van Houten (2005) built supply chain component objects using Java on top of Distributed Simulation Objects Library for on-line development of supply chain simulation models. The objects include SupplyChainActor, Trader, Retailer, Manufacturer, Customer, and Supplier.

2.6 Custom Models

In many applications, custom simulation models are provided with a data interface to allow domain users to modify a limited number of parameters. While useful for the custom applications, such developments do not provide a generic capability, i.e., the models can not be changed using the provided interfaces to represent another system in the same domain. Such customized simulation models are included in the comparison in the next section for reference even though they do not provide generic capabilities.

2.7 General Purpose Simulation Languages

One can select the approach of developing a general purpose simulation language to gain the maximum flexibility available. Needless to say, the effort for developing a general purpose simulation language will be much higher than the other options discussed above. This option is also included for reference.

3 COMPARISON OF APPROACHES

The various approaches identified in the previous section are really part of a spectrum extending from general purpose simulation software to highly specific simulators narrowly focused on an identified application domain. The approaches are compared with respect to the estimated effort for development of the generic simulation capabilities and the estimated effort by the targeted user to build models using such capabilities. Effort is defined here in relative terms with respect to the alternative approaches. It is recognized that a developer with high expertise and long experience may require less person hours than one with low expertise and short experience.

The key objective in developing such generic simulation capabilities is to reduce the time and expertise required to develop models within the identified application domain. However, reduction in time to build the model comes with reduced flexibility in the scope of the models. The more specific and defined the scope of the identified application domain, the easier it is to develop an associated simulation capability that allows shorter time for development of models by the targeted user. That is, time required to build a model using generic simulation capabilities is directly inverse of the flexibility provided in the scope of the model.

Figure 1 compares the alternative approaches presented in the previous section with respect to the development effort required for building the generic simulation capabilities and the flexibility they allow in scope of the identified application domain. The custom model and general purpose simulation language points are included for reference. The data-driven simulators provide limited flexibility in scope of the model since all the options have to be defined using data values. They are hence easier to develop than other options. At the other end of the spectrum are the object class libraries that provide a large flexibility in the scope of the model through the attributes provided for the objects representing artifacts and processes. The assumption here is that the object class libraries are designed to model a wide scope of supply chain configuration and processes. One can conceive of object classes that are narrowly defined but that would be a poor use of the capability.

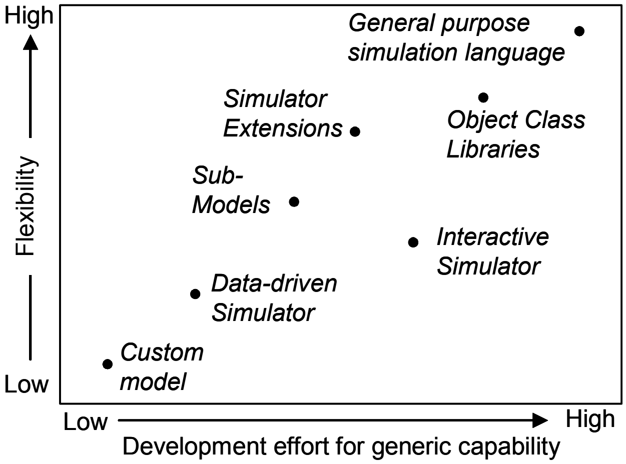

Figure 1: Comparison of alternative approaches with respect to effort to develop the generic simulation capability

Figure 2 compares the approaches with respect to the effort required by the targeted user to build models and the flexibility allowed by the generic simulation capability.

The pattern is quite similar to that of figure 1 with some exceptions. The position of interactive simulator is different in the two figures. While the interactive simulators require larger effort to develop compared to sub-models and simulator extensions, it requires less effort than these two options on the part of the targeted user to develop simulation models. The two comparisons thus suggest that the additional effort to develop interactive simulators over sub-models and simulator extensions may be worthwhile to achieve lower effort on the part of the targeted users. It will behoove the analysts to consider the option particularly if there are a large number of targeted users or expected instances of applications within the identified domain.

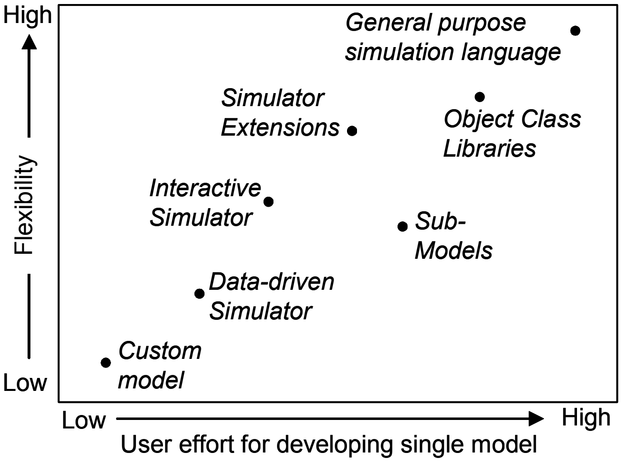

Figure 2: Comparison of alternative approaches with respect to user effort to develop a single model

The placement of the points corresponding to different approaches are in relative terms and based on general expectations. Specific features provided associated with these approaches may change the location of the points somewhat. For example, provision of industry-specific templates such as in Chen et al. (1999) will increase the development effort for the generic capability (i.e., move the interactive simulator point to the right in the graph in figure 1) and will reduce the time required by the user to develop a single instance of the model (i.e., move the interactive simulator point to the left in the graph in figure 2). Similarly, templates may be provided specific to applications such as supply chain scheduling vs. supply chain capacity planning parallel to those suggested for manufacturing simulations by McLean and Shao (2003). These templates would guide the user in the level of detail required for the specific simulation application for the subject supply chain. The impact of such specific features should be considered when addressing the tradeoffs described in the next section.

4 TRADEOFFS

The comparison of the alternative approaches highlighted the primary tradeoffs that are usually considered when building a generic supply chain simulation capability. These include the considerations of effort required to develop the capability, effort required to build models using the capability, and the flexibility in the scope of the simulation models to be built. This section briefly discusses these primary tradeoffs followed by a number of others that should be considered when embarking on a project to develop a generic supply chain simulation capability. Many of the tradeoffs are also applicable for selecting a generic capability among those available commercially or through other sources including internal, academic, and research organizations.

4.1 Effort of Targeted User for Developing Simulation Model Instances

The effort and the associated costs for the targeted user for developing simulation model instances should be the primary criterion to develop generic simulation capabilities. In many cases it may be a constraint defined by the existing personnel in an organization or in the market segment targeted by the developers of the generic simulation capabilities. The target users for a generic supply chain simulation capability may be managers or analysts at large production organizations anchoring the supply chain. If the primary target users are the managers, the generic simulation capability should be primarily data-driven and minimize any module or code level interaction. If the target is supply chain analysts with advanced engineering degrees, it may be worthwhile to consider interactive simulators and options to the right in figure 1. The position titles are sometimes misleading. Supply chain analysts and even industrial engineers at small suppliers may have a different skill set and be comfortable with data-driven simulators.

4.2 Flexibility

An associated criterion with the effort of the targeted user is the flexibility to be provided in the scope of the simulation models. A capability designed to require low effort from a targeted user for developing model instances will necessarily have to limit the flexibility available in the scope of the model. Data-driven interfaces limit the choices to a defined set of configurations (such as number of supply chain stages) and processes options (such as inventory policies at each supply chain node). Even interactive simulators offer a predefined set of attribute options for the provided modules, but provide a lot of flexibility in supply chain configurations.

4.3 Effort for Developing the Generic Capability

The effort for developing the generic supply chain simulation capability is driven by the previous two primary criteria. The desired generic simulation capability, however, may be limited by the expertise and budget available within the organization. The primary criteria may call for a flexible generic simulation capability, but the organization may not have the required expertise internally to build it and may not have the budget to buy commercial products and/or the consulting required. The available expertise and cost should hence be carefully considered.

4.4 Base Software

The base software to be used for developing the generic supply chain simulation capability should be carefully selected. Developers may build the capability using a general purpose programming language such as C# for complete flexibility, but with very high requirements for effort and expertise. At the other end of the spectrum, an organization may simply buy a commercially available data-driven simulator. Developers may also consider taking an interactive simulator available commercially and utilize it as a base to develop data-driven simulators. The selection of the base software is closely tied to the tradeoffs considered for effort of developing the generic capability and the effort of the targeted user.

4.5 User Involvement

The role of the user and the level of interaction in model building should be carefully considered. The criteria discussed above may suggest using a data-driven option with a limited role for the user, however, that may also result in the user having a low confidence in the model generated by the generic simulation capability. S/he may view the model as a black box and not buy in to its results. Interactive simulators and other similar options allow the user a deeper involvement in the model building process. Such options should be considered in cases where personnel are available to guide the user. These options should also be considered in situations where external experts, such as consultants, guide the users in model building.

A parallel consideration is user involvement in the analysis of the simulation output and the facilities in the generic simulation capability for the purpose. Similar to the input, more flexible options for analysis come with a higher requirement for expertise in building and using the options. Based on the application, routines for automatic extraction of performance measures may be provided. Some applications may call for summary measures such as supply chain lead time for use by higher level planning and scheduling purposes. The impact of features for analysis by users and for reporting for other applications need to be considered.

4.6 Animation

Animation capability is generally found useful by decision makers for understanding and verifying the supply chain flows. The development of animated representations of supply chain usually requires interactive drawing and/or drag and drop of selected icons. For environments that require animated representations of supply chain models, approaches including interactive simulators and beyond in flexibility will be needed. Data-driven simulators either do not provide an animation or provide a very basic standard arrangement of the supply chain network. The animation capability varies widely across the interactive simulators provided by commercial vendors. Simulator extensions provide access to the animation capabilities of the base simulation software and thus allow the user to build as detailed animations as desired but with associated expertise and effort requirements.

4.7 Standards

The process of building the models can be speeded up through use of standards for interfaces of the generic supply chain simulation capability. Standards should be considered in all aspects including the data input, system description and terminology, and user interactions.

A number of standards exist or are under development that address different parts of data required for supply chain simulation. These include OAGIS (OAGi 2008, for supply chain interactions messages), CMSD (Leong et al. 2006, for manufacturing system design data), and ISA95 (ISA 2008, for manufacturing system control data). Dominant software products in this space include SAP and Oracle and have thus resulted in de facto standards. In addition to the content, the current standards for data format such as XML should be considered for improved integration of the capability with other systems.

Some standards exist for defining a supply chain and associated terminology. The best known and prevalent standard for the purpose is the Supply Chain Operations Reference model, SCOR (Supply Chain Council 2008). Some supply chain simulators provide modules aligned with SCOR (see for example, eSCOR described in Barnett and Miller 2000).

The use of standards may require additional effort for developing knowledge about the applicable standards but would generally improve the applicability of the generic supply chain simulation capability. The effort invested in utilizing standards will lead to returns in terms of reduced effort in developing customized data files and reduced confusion and associated errors in concepts and terminology.

4.8 Maintenance Effort

Effort for maintaining a generic supply chain simulation capability should also be kept in view when embarking on such an endeavor. As users become more comfortable with the capability and more aware of the limits on the provided flexibility, there will be demands for additional features and options. Also, as is the case for any software, there will be requests for fixing bugs. These should be anticipated ahead and provided for. The development effort can go to waste if the users are not able to use the product effectively and shelve the software.

The maintenance expertise requirement may not necessarily follow the graphs shown in figure 1 and 2 based on the internal implementation. The data-driven simulators may be implemented using a highly compact code that provides for multiple configuration and policy options. Maintenance of such highly compact code may require a larger effort than an object class library that provides a number of separate objects each designed for a narrow focus.

4.9 Execution time

The execution time can be a concern in some cases even with the fast processors available currently. The developers need to consider if the compact module used multiple times will execute faster or if multiple instances of objects assembled together would. The use of distributed simulations may be considered, though that would require additional expertise for the developer team. Some simulation software may not allow turning the animation off and that can dramatically slow the execution down. These considerations for execution time need to be traded off against other factors such as development and maintenance efforts.

5 CONCLUSION

This paper presented the tradeoffs that should be considered when embarking on an effort to develop a generic supply chain simulation capability. Five major approaches were identified based on the efforts reported in the literature for developing generic supply chain simulation capabilities. Two additional approaches were considered at each end of the spectrum for reference. The approaches were compared with respect to the effort required for developing the capabilities, the effort required to build model instances using the capabilities, and the flexibility available in the scope of the generated supply chain simulation models. The comparison of the alternative approaches was utilized to identify the involved tradeoffs in such efforts. While the paper focused on the domain of supply chain, the tradeoffs discussed should be considered for all efforts for developing generic simulation capabilities regardless of the domain.

ACKNOWLEDGMENT

This effort was executed under the cooperative agreement number 70NANB7H6165 on “Simulation Based Supply Chain Interoperability Testing” awarded by the National Institute of Standards and Technology (NIST), the technical arm of U.S. Department of Commerce. The work described was funded by the United States Government and is not subject to copyright.

DISCLAIMERS

A number of software products are identified in context in this paper. This does not imply a recommendation or endorsement of the software products by the author or NIST, nor does it imply that such software products are necessarily the best available for the purpose.

The identification of the approach used by specific efforts is based on the information presented in the publications with some conjecture on the details in some instances. The reader is encouraged to contact authors of individual papers for details of specific implementations.

REFERENCES

Bagchi, S., S.J. Buckley, M. Ettl, and G. Lin. 1998. Experience using the IBM supply chain simulator, In *Proceedings of the 1998 Winter Simulation Conference*, ed. D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, 1387-1394. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Barnett, M.W., and C.J. Miller. 2000. Analysis of the virtual enterprise using distributed supply chain modeling and simulation: an application of e-SCOR. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 352-355. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Benson, D. 1997. Simulation Modeling and Optimization Using Promodel, In *Proceedings of the 1997 Winter Simulation Conference*, ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, 587-593, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc..

Chatfield, D.C., T.P. Harrison, and J.C. Hayya. 2006. An XML-centered approach to creating object oriented supply chain simulation models. *International Journal of Simulation and Process Modeling* 2 (3/4):115-123.

Chen, H.B., O. Bimber, C. Chhatre, E. Poole, and S.J. Buckley. 1999. eSCA: a thin-client/ server/web-enabled system for distributed supply chain simulation. In *Proceedings of the 1999 Winter Simulation Conference*, ed. P.A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 1371-1377. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Cope, D., M.S. Fayez, M. Mollaghasemi, and A. Kaylani. 2007. Supply chain simulation modeling made easy: an innovative approach. In *Proceedings of the 2007 Winter Simulation Conference*, ed. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 1887-1896. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Hicks, D.A. 1999. A four step methodology for using simulation and optimization technologies in strategic supply chain planning. In *Proceedings of the 1999 Winter Simulation Conference*, ed. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 1215-1220. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Ingalls, R. G., and C. Kasales. 1999. CSCAT: the Compaq supply chain analysis tool. In *Proceedings of the 1999 Winter Simulation Conference*, ed. P.A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 1201-2106. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

ISA, 2008. ISA-95 Manufacturing Enterprise Systems Standards and User Resources. Available on-line via: <www.isa.org>[Last accessed on July 7, 2008].

ISO, 2002. ISO/IEC 8825-4:2002 Information technology -- ASN.1 encoding rules: XML Encoding Rules (XER). International Standards Organization.

Kelton, W. 2006. *Simulation with Arena.* 4th edition. New York: McGraw Hill Inc.

Leong, S., Y. Lee, Tina , and F. Riddick. 2006. A core manufacturing simulation data information model for manufacturing applications. In *Proceedings of the Systems Interoperability Standards Organization 2006 Fall Simulation Interoperability Workshop*. Paper 06F-SIW-028.

Lian, Y.-L., and H.V. Landeghem. 2002. An application of simulation and value stream mapping in lean manufacturing . In *Proceedings 14th European Simulation Symposium,* ed. A. Verbraeck, and W. Krug.

McLean, C., and G. Shao. 2003. Generic case studies for manufacturing simulation applications. In *Proceedings of the 2003 Winter Simulation Conference*, ed. S. Chick, P. J. Sdncher. D. Ferrin. and D. J. Morrrice, 1217-1224. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc

Merkuryev, Y., G. Merkuryeva, J. Hatem, Desmet, and Bram. 2008. Supply chain simulation in the ECLIPS project. *Second Asia International Conference on Modeling & Simulation, (AICMS 08)*, 684 - 690.

OAGi. 2008. Open Application Group's Integration Specification (OAGIS) 8.0 SP2 with AIAG Overlay 1.0a. Available on-line via: <http://www.openapplications.org/downloads/oagisaiag/oagisaiag.htm> [Last accessed on July 7, 2008].

Pathak, S.D., D.M. Dilts, G. Biswas, 2003. A multiparadigm simulator for simulating complex adaptive supply chain networks. In *Proceedings of the 2003 Winter Simulotion Conference*, ed. S. Chick, P. J. Sanchez, D. Ferrin. and D. J. Morrice, 808 - 816. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Phelps, R.A., D.J. Parsons, and A.J. Siprelle. 2001. SDI Supply Chain Builder: simulation from atoms to the enterprise. In *Proceedings of the 2001 Winter Simulation Conference*, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 246-249. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Rohrer, M., 1999. Automod product suite tutorial (Automod, Simulator, Autostat) by Autosimulations. In *Proceedings of the 1999 Winter Simulation Conference*, ed. P.A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 220-226. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Supply Chain Council. 2008. Supply Chain Operations Reference-model, Version 9.0. Available on-line via <http://www.supply-chain.org/cs/root/scor_tools_resources/scor_model/scor_model> [Last accessed on July 7, 2008].

Verbraeck, A., and S.P.A. van Houten. 2005. From simulation to gaming: an object-oriented supply chain training library. In *Proceedings of the 2005 Winter Simulation Conference*, ed. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 2346-2354. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Vieira, G.E. 2004. Ideas for modeling and simulation of supply chains with Arena. In *Proceedings of the 2004 Winter Simulation Conference*, ed. R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 1418-1427. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

W3C, 1999. XSL Transformations (XSLT) 1.0. World Wide Web Consortium. Available on-line via < http://www.w3.org/TR/xslt> [Last accessed on July 7, 2008]

Wang, X., and S. Takakuwa. 2006. Module-based modeling of production-distribution systems considering shipment consolidation. In *Proceedings of the 2006 Winter Simulation Conference*, ed. L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 1477-1484. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHY

SANJAY JAIN is an Assistant Professor in the Department of Decision Sciences, School of Business at the George Washington University (GWU). Prior to joining GWU, he was a research faculty member at Grado Department of Industrial & Systems Engineering at Virginia Tech for over three and a half years. Before moving to academia, he accumulated many years of industrial R&D and consulting experience working at Accenture in Reston, VA, USA, Singapore Institute of Manufacturing Technology, Singapore and General Motors North American Operations Technical Center in Warren, MI, USA. His research interests are in the development and application of decision science techniques to complex systems with current focus on supply chain networks, emergency response project management and the intersection of the two. Sanjay serves as an associate editor of the *International Journal of Simulation and Process Modeling* and also as a member of the editorial board of *International Journal of Industrial Engineering*. He is a senior member of the Institute of Industrial Engineers and a member of APICS - The Association for Operations Management. He received a Bachelors of Engineering from Indian Institute of Technology (IIT)-Roorkee, India, a Post Graduate Diploma from National Institute of Industrial Engineering, Mumbai, India, and a Ph.D. in Engineering Science from Rensselaer Polytechnic Institute, Troy, New York. His email address is <jain@gwu.edu> and his webpage is available at <http://home.gwu.edu/~jain>.