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

INTEGRATION OF DISCRETE EVENT SIMULATION MODELS WITH FRAMEWORK-BASED BUSINESS APPLICATIONS

Peter Lendermann
Nirupam Julka
Lai Peng Chan
Boon Ping Gan

Production and Logistics Planning Group
Singapore Institute of Manufacturing Technology
71 Nanyang Drive
Singapore 638075, SINGAPORE

ABSTRACT

Simulation models and business application software as they are used for decision support in enterprise management are both representations of an enterprise's actual operations. This paper describes a unified simulation and application framework where it is possible to represent the entire performance process along a supply chain in a unified business model, improve its performance with discrete event simulation technology, and then generate and implement the corresponding business application software from the same unified model, based on a so-called framework-based application technology which allows implementation of changes derived from simulation analysis with minimal effort and time. This enables a company to optimise not only operational processes such as shopfloor or warehouse operations but also business processes such as planning, order management and scheduling through simulation.

1 INTRODUCTION AND MOTIVATION

In today's business environment "change" has emerged as the only constant. The speed, cost and efficiency at which these changes are executed have become an important competitive advantage for a corporation. However, improvements in operations and business are often discarded at the design table today as their implementation involves too much work and the advantage is lost by the time the changes are made. Further, the effects of business changes on operations and vice-versa are investigated by either employing external consultants or by assigning internal resources but seldom involve proper validation of the findings through techniques like simulation.

In this paper a novel framework for business process optimisation is described. It overcomes some of the major limitations of today's simulation approaches and state-of the art business application technology at the same time.

1.1 Challenges for Applying Discrete Event Simulation in a Demand-Driven Environment

Discrete event simulation has been established as a powerful technology to tackle a wide range of strategic, tactical and operational challenges in manufacturing, logistics, and supply chain management. In today's fast changing business environment a high degree of flexibility and scalability is required for simulation technology. It must be possible to address the entire cycle from simulation modeling, model validation, configuration of simulation runs, data input, execution of simulation, output data analysis, optimization, implementation of optimized business execution models all the way to model maintenance sufficiently fast.

Development of robust, high-performance distributed simulation has enabled to address some of these issues on a supply chain scale: geographical distribution of operations, data shielding, local maintenance of models and scalability. Application of distributed simulation technology for supply chain management has been pioneered for example by the Production and Logistics Planning Group at Singapore Institute of Manufacturing Technology (SIMTech) in collaboration with Nanyang Technological University (Gan et al., 2000) and Georgia Institute of Technology (Julka et al., 2002), and also by the Manufacturing Engineering Laboratory of National Institute of Standard and Technology (McLean and Riddick, 2000). Both groups are leveraging on the High Level Architecture (HLA) as a general-purpose architecture for simulation reuse and interoperability (Kuhl et al., 1999).

In conventional approaches, simulation models typically represent the operational execution on the shopfloor or in a warehouse, and simulations are driven by simulated release of materials into the system. Input releases, however, are difficult to generate in today's pull-environments that are driven by customer demand scenarios with high variability and frequent phase-in of new products. In such an environment, input releases are the result of a complex translation from customer demand into material quantities to be released into and moved within the system at pre-specified times (see Figure 1). Consequently, in a demand-driven business world, simulation models can be made much more realistic if the process of translating demand into input release (i.e. planning and order management processes) and the complex interdependencies between the business processes and the operational execution are fully incorporated into the model.

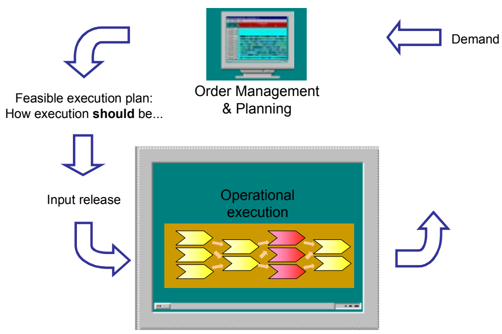

Figure 1: Simulation Models in a Demand-Driven Business Environment

Today, simulation models address this issue to a limited extent since they generally incorporate a relatively crude abstraction of the associated planning processes. A more straightforward way of translating customer demand into feasible input release rates is to integrate the underlying planning and customer order management procedures into the simulation (Lendermann et al., 2002).

1.2 Framework-Based Business Application Technology

Planning and order management processes as described above are typically handled by Enterprise Resource Planning (ERP) or Advanced Planning and Scheduling (APS) systems. One of the major shortcomings of today's ERP and APS packages is that they are inflexible and have limited ability to handle very complex business processes. Such business applications are initially created using company-specific business model templates. Business processes are encapsulated within the application and changes in the business processes are always constrained by the templates. Customisation can be accomplished only by manually changing settings and parameters in the existing software application.

Each system has its proprietary, closed data model with redundancies to the other system(s). Systems are connected through point-to-point interfaces with fixed data format.

The shortcomings of this so-called "integrative" approach are addressed by a novel so-called "framework-based" business application technology. The underlying principle is to create a meta-model of the business processes and then use this model to directly generate the application. The meta-model is created using simple elements (process, resource and order). Subsequent changes of any kind can then be made without any constraints imposed by any template because the new application is not just a variation of the old application with changed settings and parameters but it is generated directly from the updated meta-model and actually replaces the old application. Framework-based applications allow easy configuration of components to create dynamic applications with flexible data models. This allows dramatic reduction of the overall time for the creation of a business application from a business model and the effort for reconfiguration of the application every time the business model changes.

Framework-based applications for commercial use have been pioneered by companies such as SKYVA International (now part of ABB; see SKYVA, 2000) and Ramco Systems (http://www.ramco.com/). They use a combination of object-oriented and software agent technology. The various business documents and the operations that can be performed over these documents are modelled as objects. The meta-model is prepared by using these documents and operations objects and linking them together using a visual interface. The business application is generated by the creation of software agents, who execute the operations over the documents. The connections between different agents are created based on the processes defined in the meta-model. This agent framework is essentially the backend of the application.

Framework-based business application software systems have drastically reduced the implementation time and the effort that is needed to reconfigure enterprise application to incorporate alternative policies and changes in the highly complex business processes of corporations.

1.3 Limitations of Using Business Applications for Business Re-Engineering

Within a framework-based implementation of a business application, simulations are presently used to increase the efficiency of the application and test for system integrity and robustness. The business processes themselves are deterministic in nature. Randomness can be introduced into a scenario by adding programs that emulate the real situation with some probability distribution (i.e. simulation models) and data files having scenario-specific data to emulate the databases. Statistics collected are for an individual agent and for each scenario. Statistics collected over multiple replications are used for analysis. The simulation is performed in real time.

This kind of simulation - which is not discrete event simulation (DES) - is not sufficiently able to provide direct and obvious information on the impact of changes in the business model upon the operational and business performance.

Both framework-based and conventional business applications such as ERP and APS systems are tools to run a business rather than to re-engineer and optimize the business. They are not able to address all aspects of the complex interdependencies with operational execution and variability on the shopfloor. A conventional solution to overcome this problem would be to translate the processes of the business application into a simulation model, using a commercial simulation tool. However, this would mean a cumbersome translation of one computer representation of the business model into another one. It would be possible to analyse different business scenarios and optimise the underlying models based on the results but implementation/realisation of these changes in the corresponding business applications would remain cumbersome and in some cases impossible.

An alternative would be to use the business application itself within the simulation model. But business applications by nature are designed to run in real time only but not in simulation time. A novel approach to overcome this problem would be to make the business application itself compliant with discrete event simulation (DES-compliant). A replicated copy of the application could then be used as part of the simulation model. This means that ultimately the same piece of software could be used to run the business in the real world and also to represent the business operations in the simulation world for re-engineering and optimization of the overall business performance. Synchronization of real time and simulation time would be achieved by using the same technology that is used to synchronize federates in a distributed simulation, namely the Run-Time Infrastructure (RTI) of the High Level Architecture.

2 DESCRIPTION OF FRAMEWORK

2.1 Problems Addressed

The idea described here is to enable framework-based applications to perform discrete event simulations and to combine these applications with discrete event simulation model components. In this way it is possible to address the shortcomings when applying framework-based business applications or discrete event simulation systems individually:

- Even with advanced framework-based applications the performance of business operations can only be addressed based on the real history of the system.
- Operational models that are optimised using conventional discrete event simulation technology are difficult to implement.
- The scope of traditional simulation models does not sufficiently take into account the fact that the underlying business is ultimately driven by customer demand rather than by release of materials into the system.

2.2 Solution

2.2.1 Overall Philosophy

Figure 2 describes the overall philosophy of the framework. The business of any enterprise comprises of an operations model (representing shopfloor/warehouse operations etc) as well as a business process model (representing planning, order management etc). The application to execute the business is created using a framework-based implementation. This application is usually an abstraction of the business process model. On the other hand the operational simulation is the abstraction of the operations model. The idea is to use the operational simulation in conjunction with the framework-based business application to perform an integrated business simulation. This allows to make improvements over both the business and the operations model at the same time.

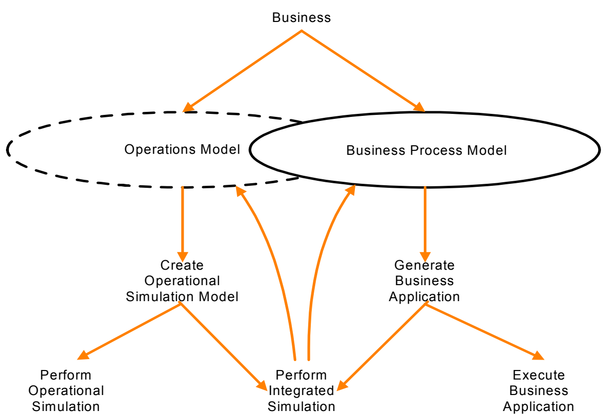

Figure 2: Overall Philosophy of the Framework

Figure 3 describes how the applications are improved using simulation and how the transition between an old application and a new application is carried out. The application is DES-compliant. A replication of the application (Application 2) and the connected databases (DB2) is created. Application 2 and DB2 are used for simulation. Based on the simulation, changes are suggested for the business model. These changes are incorporated in the model and a new Application 2 is generated or, in other words, Application 2 is updated (see Figure 3a). At the point where Application 2 is considered to be the better application to work with, a similar process is repeated (see Figure 3b). Application 1 is discarded and the original database (DB1) is used to upload the information of the various active objects into Application

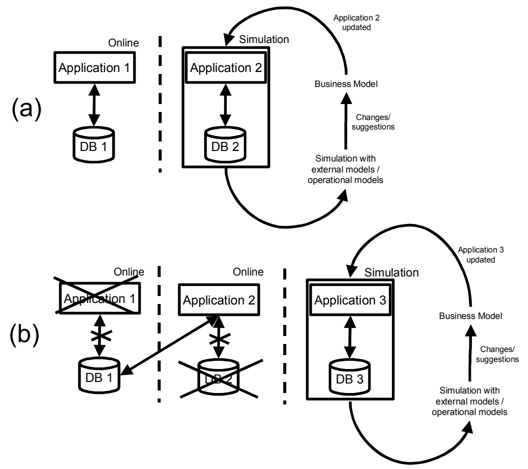

Figure 3: Improvements over Business Model and Application by Combining DES and Framework-Based Application

2. A new Application 3 is generated along with database DB3 (a snapshot of DB2). Application 3 and DB3 are now used for simulation purposes.

To achieve this, the following challenges have to be resolved:

- All the relevant manufacturing, logistics and decision-making events as far as they are not triggered by the application itself have to be emulated.
- The application has to be made DES-compliant (i.e. it can be synchronized with the simulation model).
- The capability to configure various scenarios on the fly and perform multiple simulations to analyse the performance of the business operations and changes made in it has to be created.

2.2.2 Relevant Business and Execution Processes

For overall performance optimisation all processes that are related to operational execution have to be included into simulation such as:

- Process customer order (very critical in an environment where the available-to-promise procedure stretches across several IT systems),
- (Re-)generate execution plan/schedule,
- Execute production.

Examples for "Process customer order" and "Execute production" are given in Figures 4 and 5, respectively. Process steps that are irrelevant for operational performance (such as credit check or order book update, see Figure 4) can be skipped. Other supporting business processes such as monetary transactions, execution of equipment maintenance, customer inquiries that do not lead to orders, do not have to be included into the simulation.

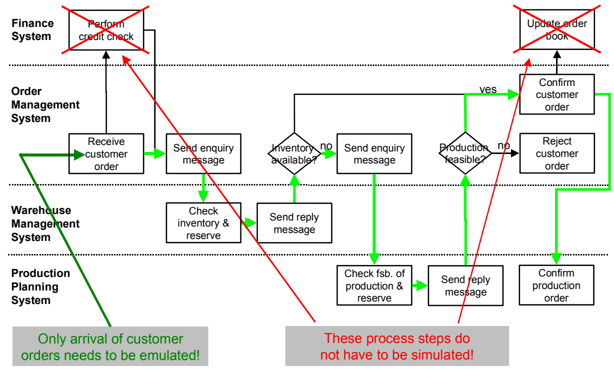

Figure 4: Sample Process "Customer Order Processing"

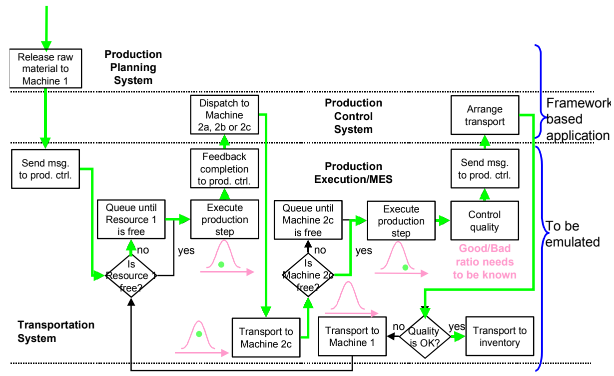

Figure 5: Sample Process "Production Execution"

Assuming that it is fully automated, the entire customer order process as shown in Figure 4 is to be simulated and executed on the framework-based application. On the other hand, anything that is not part of the application has to be emulated such as arrival of incoming customer orders, release of material into production systems (push environment), start of production order execution (pull environment), execution of a production step and timely feedback to production control system. This is illustrated in the example shown in Figure 5.

Examples of business process changes that one might want to simulate, assess and implement with such a technology are:

- From "Order Management queries inventory level" to "Inventory changes are reported to Order Management",
- Change elements and/or sequence of queries in available-to-promise process,
- Change dispatching rules in production scheduling/control,
- Temporarily change resource structure,
- Outsource parts of the performance process.

2.2.3 DES Compliance of Business Application

There are three general conditions to be considered in making the software application DES-compliant.

The application should be made of modules and components which are linked to each other to represent a process. The software components may be software agents and also encompass other ways of achieving such functionalities.

The triggering of subsequent steps in any process should be able to be paused indefinitely through some control mechanism outside the application. This is done so as not to interfere with the processing of any other orders, documents, instructions, etc. in the application. In other words, the step between components can be controlled externally.

Also, the process flow should be exposed outside the application. For example, the process flow can be available as a separate document.

A discrete event simulation comprises of a number of events occurring over instances of time in an orderly fashion. Each event takes place at a particular simulation time. To make the application DES-compliant the application and its internal components needs to be synchronised with the external models. The external models advance in simulation time whereas the application advances in real time.

Figure 6 explains the situation with an example. Agent 1 receives event A at real time RT_A and simulation time ST_A. Agent 2 receives event B at real time RT_B and simulation time ST_B. Agent 1 processes event A to generate event C. The processing of event A takes p seconds (real-time).

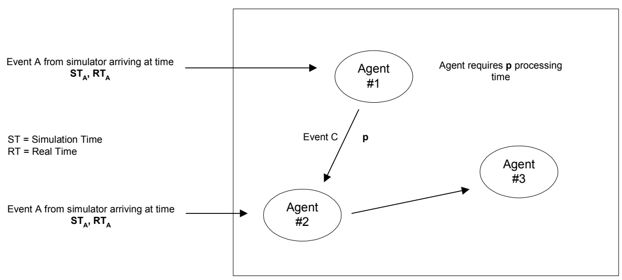

Figure 6: Illustration of the Focal Problem of Synchronisation

Event C has to be processed by Agent 2 before event B at simulation time ST_C. In this case, since the application advances in real-time, Agent 2 may end up processing event B before event C. Equations 1 and 2 illustrate this situation. Event C has to be processed before event B but the time it takes to be generated (RT_A + p) is past the time event B is generated and sent to Agent 2.

$$RT_A + p > RT_B \quad (1)$$

$$ST_C < ST_B \quad (2)$$

Further complications can arise due to latency caused by network traffic, where a message from an agent's past arrives in the future. This causes a paradox and thus the discrete event simulation may be no longer emulating the truth. Thus a mechanism is also needed to ensure that it is safe to advance the time of an entity to the next designated value. A mechanism for synchronisation of simulation time with real time will solve these problems.

2.2.4 Mechanism of Synchronization

A two level solution is provided to address the issues mentioned above:

1. Development of the mechanism of synchronisation for applications advancing time using
   - State changes,
   - Message-based communication.
2. Development of a standard communication interface between the application and the HLA network using
   - Adapters provided by the application,
   - Direct interaction with the agents.

2.2.4.1 Pre-Requisite Steps

Events in an application may represent changing of states of certain objects or arrival of certain messages. In the two cases the synchronisation mechanism - although similar in nature – is different. This approach requires that all the process maps are available and additions/modifications to the application are permitted. Irrespective of the type of application, there are two steps which need to be performed as a pre-requisite to our solution. These steps are:

- **Discretisation of the Processes:** The process maps are to be broken down into discrete sub-processes. The smallest sub-process will then be represented by one operation over an object.
- **Development of the Application Federate:** The application federate performs two functions. The first is to interact with the other federates and represent the application. The second function will be to control the various operations inside the application and synchronize it with the rest of the federation. This will be performed in two different ways depending upon the nature of the application i.e. state-based or message-based.

2.2.4.2 Applications that Advance Time through State Changes

State changes trigger agents to operate on various objects in a manner defined by the business processes. These operations on objects change their states further triggering off other agents. The important point is the last operation/state change in any sub-process. If left to continue on its own, it would trigger off a new sub-process. Take for example the case in Figure 7. There are two sub-processes A and B. Each one of them have some tasks to be performed. The sequence of tasks specifies that an external event E has to occur before sub-process B is started. Thus there is a need of synchronisation. The agent which triggers off B waits for the change of state of object O. The last operation in sub-process A is the changing of state of O.

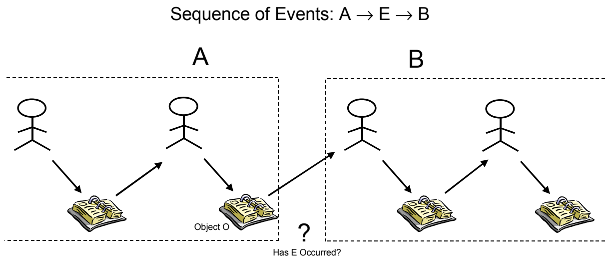

Figure 7: Applications that Advance Time through State Changes

Two solutions are feasible to address this issue. They depend upon the choice of user and vary in the amount of changes that need to be made in the manner the applications are generated. (see Figure 8a and 8b).

Each agent in the application is capable of performing two extra functions (federate-sent and federate-receive). These functions are triggered depending upon the position of the agent in any sub-process. If the agent is the last in sub-process A then it triggers federate-sent once it completes its designated task. The federate-sent function informs the application federate that the particular sub-process has been completed. In case an agent is at the start of sub-process B and the state change occurs that triggers it, then the agent calls the function federate-receive. This function informs the application federate that a new sub-process has to be started. The agent starts the new sub-process when it gets a call-back to this function. In case before a call-back is received, the state of the same object changes, the agent then performs the operation related to the latest change. This solution is illustrated in Figure 8a.

The last agent in sub-process A change the state of object O. There is a federate agent which is designed just like any other agent in the application apart from the fact that it can communicate with the application federate. The federate agent changes the state of an object O' (clone of object O) based on the synchronisation with externals. This change in state of object O' triggers the first agent in the following sub-process (see Figure 8b).

2.2.4.3 Applications that Advance Time through Message-Based Communication

Arrival of a message triggers an agent to perform operations on various objects. The agent can send a message anytime to another agent to start off other tasks. This application already has a number of features of DES and requires very less amount of modifications. The destination agent name/id is included in the attributes of a message. All messaging between agents belonging to two different

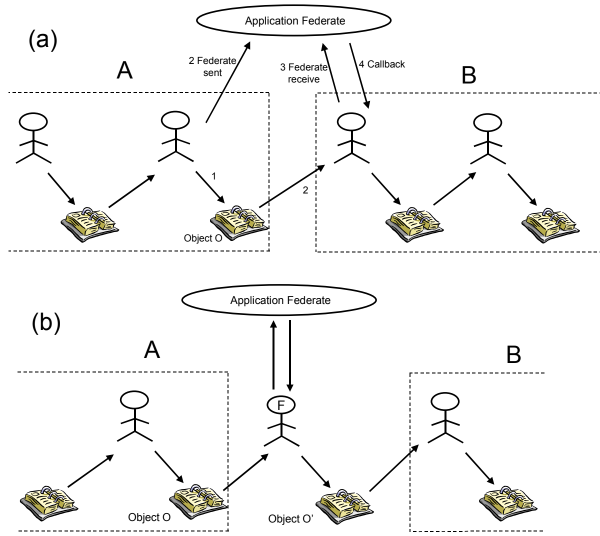

Figure 8: Solution for Applications That Advance Time through State Changes

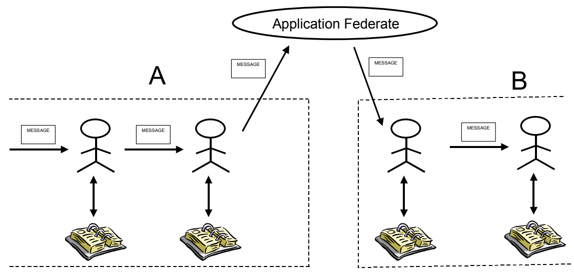

Figure 9: Solution for Applications that Advance Time through Message-Based Communication

sub-processes is routed through the application federate (see Figure 9). This allows the application federate to control the flow of information inside the application and synchronise with the other federates.

2.2.5 Interface between the Application and the Application Federate

The interfaces between the High-Level Architecture (HLA) and the application can be using an already available adapter or a direct connection with the agents. The cases are shown in Figure 10. A further layer of a translator can be used such as the communication language used on the application side which can be one of the available XML standards.

3 BENEFITS AND CONCLUSIONS

The framework described in this paper enables integrated optimisation of business processes and operations of an enterprise. It allows an industrial user to actively manage transitions from one business model to another and to an-

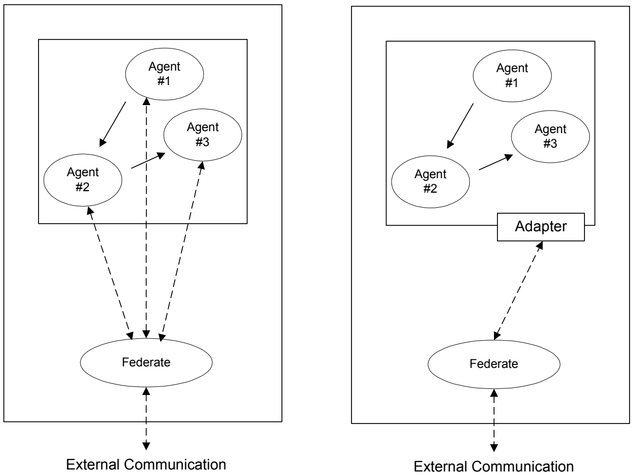

Figure 10: Interfaces between Application and the Application Federate

alyse the effect of each small change on the operations. The combination with distributed simulation technology makes the framework scalable. Realistic supply network design, analysis and enhancement becomes possible since the dynamic behaviour and variability are taken into account and planning algorithms can be incorporated. Supply networks can be managed in a collaborative manner with globally distributed suppliers and customers. Results can be obtained fast since simulation models can be developed and configured rapidly and parallel execution on distributed resources is possible.

By using framework-based business application technology it can be ensured that changes suggested and tested in the simulation can be implemented in the real life system with minimal effort. The approach may also work with conventional business applications but the impact would be lost.

As a result it will be possible to tackle a wide range of strategic, tactical and operational challenges. This allows to consider such a technology a worthwhile investment rather than a one-time expense.

The technology appears to be applicable in environments that are subject to high variability and stochastic uncertainties across the entire supply chain and where a lot of complex operational interdependencies between suppliers and customers (e.g. waferfab and semiconductor assembly & test) bear significant potential for optimisation and therefore foster the search for collaborative performance improvement.

4 FUTURE WORK

A laboratory prototype is currently being developed to establish the feasibility and applicability of the framework described in this paper. This prototype will consist of

- a business application encompassing, in minimum, an order management system, a scheduling system, and a job tracking system,
- a simulation model of the production execution system,
- an order arrival simulator, to simulate the arrival of customer orders based on historical data,
- an application federate that acts as the mediator between the business application and the simulated production execution and is responsible for synchronizing the business processes with the simulation models.

Concurrently, plans of collaboration with a commercial framework-based business system provider are underway to further develop this prototype into a commercial framework that allows to reconfigure, analyze, adapt, implement and execute business operations, all with the same piece of technology.

REFERENCES

Gan, B.P., L. Liu, S. Jain, S.J. Turner, W. Cai, and W.J. Hsu. 2000. Distributed Supply Chain Simulation Across the Enterprise Boundaries. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J.A. Joines, R.R. Barton, K. Kang, and P.A. Fishwick. 1245-1251.

Julka, N., B.P. Gan, D. Chen, P. Lendermann, L.F. McGinnis, and J.P. McGinnis. 2002. Framework for Distributed Supply Chain Simulation: Application as a Decision-Making Tool for the Semiconductors Industry. In *Proceedings of the International Conference on Modeling and Analysis of Semiconductor Manufacturing (MASM 2002)*. 376-381.

Kuhl, F., R. Weatherly, and J. Dahmann. 1999. *Creating computer simulation systems: An introduction to the high level architecture*. Prentice Hall PTR.

Lendermann P., N. Julka, B.P. Gan, and L.F. McGinnis. 2002. Framework for Distributed Supply Chain Simulation: Technical Feasibility and Implementation Approaches for the Semiconductor Industry. In *Proceedings of the International Conference on Modeling and Analysis of Semiconductor Manufacturing (MASM 2002)*. 370-375.

McLean, Charles, and F. Riddick. 2000. The IMS Mission Architecture for Distributed Manufacturing Simulation. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fiswick. 1539-1548.

SKYVA International. 2000. *Creating competitive advantage: An Executive Overview*. SKYVA International, Inc., Medford, Massachusetts.

AUTHOR BIOGRAPHIES

PETER LENDERMANN is a Senior Scientist in the Production and Logistics Planning Group at Singapore Institute of Manufacturing Technology (SIMTech). Previ-

ously he was a Managing Consultant with agiConsult in Germany where his focus was on the areas of supply chain management and production planning. He also worked as a Research Associate at the European Laboratory for Particle Physics CERN in Geneva (Switzerland) and Nagoya University (Japan). He obtained a Diploma in Physics from the University of Munich (Germany), a Doctorate in Applied Physics from Humboldt-University in Berlin (Germany) and a Master in International Economics and Management from Bocconi-University in Milan (Italy). His research interests include parallel and distributed simulation and advanced methods for supply chain planning and production scheduling. His email address is <peterl@SIMTech.a-star.edu.sg>.

NIRUPAM JULKA is a Research Engineer with the Production and Logistics Planning Group at Singapore Institute of Manufacturing Technology. He received a Bachelor of Technology (Honors) from Indian Institute of Technology (IIT) Kharagpur in 1999 and a Master of Engineering from the National University of Singapore in 2002. His research interests are parallel and distributed simulation, supply chain optimization and management and decision support systems. His email address is <nirupamj@SIMTech.a-star.edu.sg>.

LAI PENG CHAN is a Senior Research Engineer with the Production and Logistics Planning Group at Singapore Institute of Manufacturing Technology. She obtained her BSc and MTech at the National University of Singapore. Her research interests include modelling and simulation, optimization and intelligent systems. Her email address is <lpchan@SIMTech.a-star.edu.sg>.

BOON PING GAN is a Research Engineer with the Production and Logistics Planning Group at Singapore Institute of Manufacturing Technology. He is currently leading a research project that aims to apply distributed simulation technology for supply chain simulation. He received a Bachelor of Applied Science in Computer Engineering and a Master of Applied Science from Nanyang Technological University of Singapore in 1995 and 1998, respectively. His research interests are parallel and distributed simulation, parallel programs scheduling, and application of genetic algorithms. His email address is <bpgan@SIMTech.a-star.edu.sg>.