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

A GENERIC FRAMEWORK FOR REAL-TIME DISCRETE EVENT SIMULATION (DES) MODELLING

Siamak Tavakoli
Alireza Mousavi
Alexander Komashie

Systems Engineering Research Group (SERG)
Brunel University
School of Engineering and Design
Uxbridge, UB8 3PH, UK

ABSTRACT

This paper suggests a generic simulation platform that can be used for real-time discrete event simulation modeling. The architecture of the proposed system is based on a tested flexible input data architecture developed in Labview, a real-time inter-process communication module between the Labview application and a discrete event simulation software (in this case Arena). Two example applications in the healthcare and manufacturing sectors are provided to demonstrate the ease of adaptability to such physical system.

1 INTRODUCTION

Manufacturing systems comprise of large and complex interrelated processes and human inspired initiatives. Continuous transformations take place inside these enterprises in specific time intervals. By transformation we mean changes in system parameters and the rules that govern these systems. The result has been the introduction of more computers and computer controlled equipment on the factory floor. This has led to the concept of computer integrated manufacturing (CIM) (Jones and Saleh 1990) and Discrete Event Simulation modeling amongst other technologies as tools for analyzing and controlling these complex systems.

For several years, simulation has been applied to the long-term planning, design and analysis of manufacturing systems. These models have been termed “throw away models” because they are seldom used after the initial plans or design are finalized (Son and Wysk 2001; Smith and Brett 1996; Harmonosky 1995). Over the past decade, however, researchers and practitioners have taken advantage of the power of simulation technology to develop simulation models that can be fully integrated into complex manufacturing systems and run in real-time. The ability to automatically generate simulation models for certain application has also been achieved (Son and Wysk 2001).

Harmonosky (1995), conducted a review of simulation based real-time scheduling and found the need to further explore the concept with look-ahead and what-if capabilities. More recent attempts to use real-time simulation modelling in the control and analysis of manufacturing systems may be found in Mullarkey et al. (2000), Rabbath et al. (2000), Lee et al. (2002), Dangelmaier et al. (2006) just to mention a few.

In recent years, DES is finding application in several non-manufacturing environments including healthcare. The labor intensive nature of healthcare systems however has made this application more challenging. Subsequently, a real-time application of DES in healthcare is non-existent. In this paper, we present a framework for accomplishing real-time control of a manufacturing system and which is adaptable to non-manufacturing systems as demonstrated by a healthcare application in the body of the paper.

The proposed framework is based on a tested flexible input data architecture developed in Labview, a real-time inter-process communication module between the Labview application and a discrete event simulation model developed with the Arena simulation software. Two example applications in the healthcare and manufacturing sectors are provided to demonstrate the ease of adaptability of the framework.

2 A REVIEW OF LITERATURE

2.1 Discrete Event Simulation in Manufacturing Process Control and Scheduling

This section reviews various works that have been dedicated to the application of discrete event simulation (DES) in shop floor control and scheduling.

Vaidyanathan et al. (1998) developed a discrete event simulation model as a daily scheduling tool. They employed a hybrid approach that integrates a scheduler and a simulation model. The simulation model plays the role of modifying the output from the scheduler, and the two together become a tool for day-to-day production scheduling.

An earlier work by Onut et al. (1994) also shows how simulation was integrated into a complete shop floor control system for a semi-integrated Manufacturing System (S-IMS). They developed a framework that interfaced the simulation system with a Material Requirement Planning (MRP) system, a host computer, a database Management System (DMS), a shop floor control system and a supervisory input system. This greatly enhanced the effectiveness and control of the manufacturing operations.

Gupta et al. (2002) propose shop floor scheduling with simulation based proactive decision support in a highly manufacturing complex system where multiple product parts, sequence dependent setup, molding machine specifications, mould restriction's etc with a variety of scheduling and operational choices are integrated. Gupta et al. (2002) developed a simulation model that generates a feasible schedule and has ability to reschedule the system when sudden changes occur. The ahead of time system parameters provided the scheduler with the opportunity to find best schedules efficiently.

Potoradi et al. (2002) also developed a simulation-based scheduling to maximize demand fulfillment in a semiconductor assembly facility. They used simulation as an engine to generate schedules and to control various machines at execution time and also to plan for the start of materials. The schedule adapts to "unforeseen" changes on the shop floor by the use of online data availability. However, their data entry from the shop floor and planning system is not fully automated, hence the model-update is quite slow and requires an expert and is not done frequently. From the above review, it will be noticed that simulation plays a vital role in the understanding, control and improvement of complex systems. Additionally, it is observed that there is basically not much difference in the approach to applying simulation in scheduling. However, the details of the integration and framework are custom made to suit the particular environment.

The area of shop-floor control and scheduling demonstrates an aspect of the capability or power of simulation but there is more to it. Model generation could be automated and shop floor control performed in real-time. A number of these cases are discussed in the next section.

2.2 Discrete Event Simulation in Real-Time Control

Real-time systems differ from traditional data processing systems in that they are constrained by certain non-functional requirements (e.g. dependability and timing constraints or requirements). An efficient simulation of real-time system requires a model that satisfies both simulation objectives and timing constraints, Lee et al. (2001). Son and Wysk (2001) developed a structure and architecture for automatic simulation model generation for very detailed simulation models intended to be used for real-time simulation based shop floor control. They identified two essential stages to be automated for automatic simulation model generation: System specification and the associated model construction. In this work, Son and Wysk (2001) proposed a methodology for generating an Arena simulation model from a resource model (in MS access 97) and a message-based part state graph (MPSG) based shop floor control model. This was made possible because the Arena simulation software supports visual basic application (VBA), which enables application integration and automation. Lee et al. (2001) undertook the development of a modeling methodology to efficiently model real-time systems to satisfy given simulation objectives and to achieve arbitrary timing requirements.

Manivannan and Banks (1991) also provide a brief review of earlier attempts to building intelligent controllers for managing operations in a manufacturing cell with and without simulation.

In all the above cases, the gain in system performance as a result of the use of simulation has been noticeable. In an earlier work by Wu and Wysk (1988), they observed that significant improvements could be made by using a simulation model to determine the future course for a manufacturing system.

Up to this point, the literature review has focused on development and applications in the manufacturing sector even though it is not the sole focus of this work. It is clear that simulation has been used most extensively in manufacturing than probably any other field. The present framework seeks to extend the application of real-time simulation into healthcare but before presenting the concept, a brief critique of traditional simulation is presented in the next section.

2.3 Weakness of Traditional Simulation

By traditional simulation is meant the approach to systems modeling and simulation that follows the methodology of Banks (1998) and which Son and Wysk (2001), described as "throw-away" tools. This approach normally requires a simulation expert to build the model and is heavily dependent on historical data. The main drawbacks of this approach are evident in numerous published work in simulation and are summarized below with particular reference to simulation projects in healthcare undertaken by the authors of this paper (see Komashie and Mousavi 2005 and Komashie et al. 2008). The main shortcomings of traditional simulation are:

2.3.1 Time consuming

Traditional simulation often requires the manual collection and analysis of input data. Mining the data and preparing them for use in a model is always time consuming. Sometimes data that is seemingly available may not be in the format usable for a simulation study. In healthcare for instance most processing time data and proportion of patients at various branches in the system are unavailable and have to be collected. Apart from being a time consuming exercise, the data obtained and processes defined are also time dependent and subject to change in short cycles.

2.3.2 Time dependent

Due to the fact that traditional simulation is heavily dependent on historical data, as the input data gets old the results of the model also become less reliable. This is a critical issue in dynamic and complex systems like healthcare and manufacturing. In some cases, by the time the modeling project is completed, the input data that was collected for the model may obsolete and may cause doubt in usability of the results. When the input data to a model is not as reliable, then it becomes difficult to use the model to predict future events accurately.

2.3.3 Inaccurate for Prediction

Simulation models help to understand the operations of a system and serve as a cost effective, risk free platform for testing different configurations of the system. In addition, a well validated model of a system may be a useful tool for predicting future events. However, this is not a reliable exercise with traditional simulation which is dependent on historical data.

This has been one of the incentives for researchers and practitioners to research into real-time data acquisition and control systems.

2.3.4 Costly

With the expertise required and the time it takes to build and run good simulation models, the cost of keeping a traditional simulation model up to date and fit for predictive analysis would be prohibitive. To reduce this cost and providing a reliable platform for system managers to predict future events more accurately is part of the motivation for the proposed generic framework.

The above review provide some evidence of the extent of the application of real-time discrete event simulation modeling in the manufacturing industry. With regards to the service industry particularly healthcare, evidence from comprehensive reviews conducted by Brailsford (2007), and Eldabi et al. (2007), indicate that the application of real-time discrete event simulation is non-existent.

The concept of the proposed framework as presented in the next section highlights with examples, the ease of adaptability that makes the system applicable to a healthcare environment.

3 CONCEPT OF THE PROPOSED GENERIC FRAMEWORK

3.1 Main Features

Our proposed framework is intended to satisfy three main features; Flexibility, Real-time, and Fast-forward. Flexibility guaranties the generality of the framework in terms of working with any set of data entries and cost model for any application. Real-time framework intends to make effective use of real-time data for both cost index calculations and simulation modeling. However, data will also be stored for possible historical analysis which may help with fast-forward simulation.
Below paragraphs define the role of these features within the framework;

3.1.1 Flexible Input

By adopting the proposed flexible data input layer architecture (FDILA) definition of any number of input points, and acquire their data in any type becomes possible (Tavakoli et al. 2008). An assortment of proprietary data acquisition cards to wireless communication ports data can be defined and scanned in this system. The framework allows a wide variety of data type conversions. Within the architecture a number of data inputs can be fused to build a single input variable representing one virtual input. A decision modeler decides whether a set of data input has any noticeable effect on system performance or not. And if so the priority associated with that parameter is also determined. Therefore, the sampling frequency of input data can be determined (system scan rate) in a dynamic fashion..

3.1.2 Flexible Model Definition

The system is capable of allowing for a wide range of mathematical models for defining the relationships between raw data inputs, intermediate variables, and final cost factors.

3.1.3 Real-time Data Acquisition

A dynamic frequency of data input rate (scan rate) guarantees access to real-time data. The frequency of sampling may change based on the levels of importance of the input data determined by the optimizer.

3.1.4 Real-time Simulation

The current sampled data, together with stored data from the past builds up the basis for measuring Key Performance Indicators of the enterprise system. The current scanned data, either in raw format or translated as information, triggers events on the simulation platform. As a consequence of real-time performance analysis.

3.1.5 Fast-forward Simulation

At any desired time during running the real-time simulation, it may be possible to pause the stream of real-time data into the simulation, and instead, feed it with modified input data according to a 'what-if' scenario. The result of the simulation and cost model will be virtually result of the conditions set in input modifications.

3.2 Data Flow

The data flow diagram in figure 1 and the following paragraph provide an illustration and explanation of how framework provides real time data for simulation and analysis.

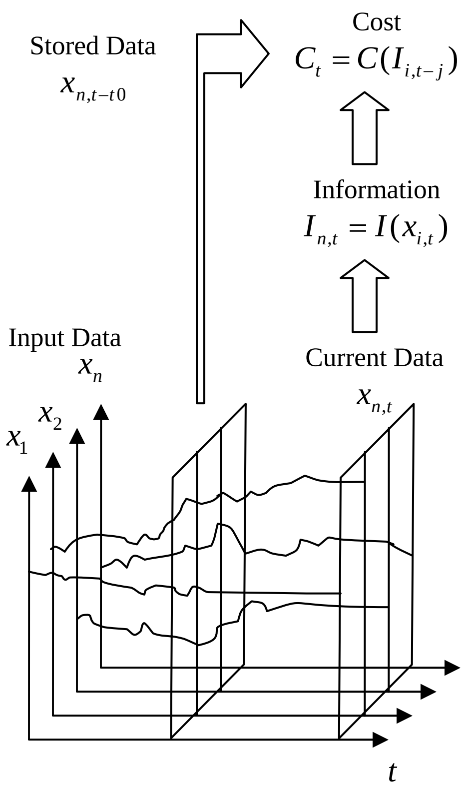

Figure 1: Real-time Data Flow into the Cost Model

Based on the proposed framework, and as illustrated above, raw data of any type may be read at any frequency and restructured for later reference. Acquired data may further be translated to any new type of information which may build up components of the cost index. A wide variety of methodology may be implemented for cost index modeling, which may need some of the prepared information. Since all input data values may not be needed by cost modeling algorithm at all times, a filtering algorithm may be structured to act as the key input selector prior to calculating the cost index components, Tavakoli and Mousavi (2006). Both raw data and their associated translated information are available for triggering events on a discrete-event based simulation model. Therefore, events of the simulation model follow the real-time activities of the physical environment. The acquired real-time data is dumped into a database as well as being used on the real-time simulation. This provides supply of historical data for both data mining techniques if needed for a part of cost modeling methodology, and curve fitting algorithm which in turn supports Fast-forward simulation.

4 ARCHITECTURE OF THE PROPOSED FRAMEWORK

An architecture is proposed in order to handle tasks and responsibilities explained in section 3 (above paragraphs). Two major components comprise the main functionality of the framework. Data Integration and Processing component is responsible for real-time data acquisition, data fusion, and pre-processing, including filtering and curve-fitting. Simulation Modeling Engine component takes care of discrete-event simulation and modeling the cost index.

4.1 Data Integration and Processing

Flexible Data Input Layer Architecture (FDILA) introduced by Tavakoli et al. (2008) is used as basis for data acquisition and data restructuring as well as for building up the cost index components. In this layered architecture, two main layers exist. First, a Data Integration Module (DIM) interfaces the network of input sensors with the next layer and prepares the input data as variables to be taken by the next layer for data processing. DIM consists of two internal layers, Data Interface Unit (DIU) handles scanning and reading sensor data. In Data Fusion Unit (DFU) scanned input data get a structured format and possibly combine together to generate new variables. In the second layer, the Data Processing Module (DPM) data storage, filtering and analysis take place. The first layer inside DPM, the Pre-Processing unit (PPU) controls the rate of data storage and access to historical data for curve fitting. The other layer of DPM, the Post-Processing Unit (PoPU) prepares the essential components for the Simulation Model for Cost Index calculation. Figure 2 shows the main components and layers designed for Data Integration and Processing.

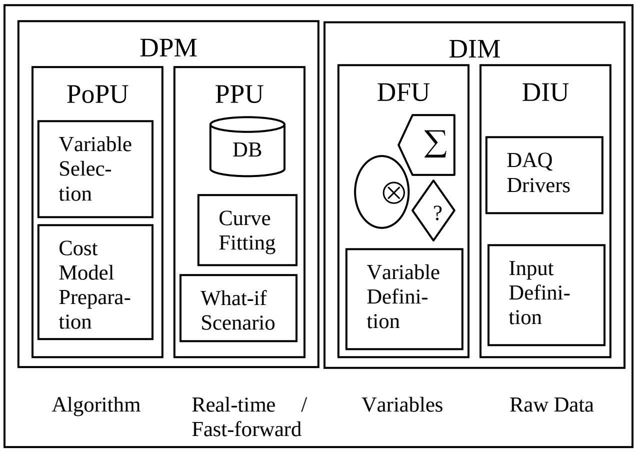

Figure 2: Data Integration and Processing layers and components

4.2 Simulation Modeling Engine

Discrete event simulation as an established methodology for simulation of shop floor events is used on the proposed framework. This component is prepared for being triggered by real-time events flowing from Data Integration component.

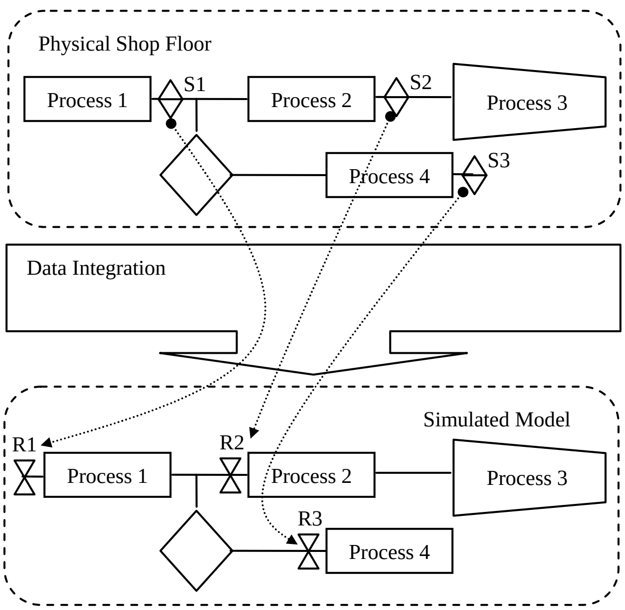

Figure 3: Real-time Model Matching Mechanism

Therefore, the process flow on the simulated model runs continuously based on receiving discrete events from shop floor activities occurring at certain points. Creation and progress of each entity, or batch of entities, may depend on the contents of the message which is sent to the Simulation Engine. Wherever on the physical environment a sensor provides data for the simulated model, at the corresponding point on the model, a mechanism is applied to react to the new coming data. This triggered reaction may be creation of a new entity, progress of an entity, or change of status of an entity or resource on the process flow. This Real-time Model Matching Mechanism (R3M) is illustrated on figure 3.

In figure 3, S1, S2, and S3 are sensors which provide the simulated model with new data. R1, R2, and R3 are receivers following the data sent by the corresponding sensor data.

4.3 R3M Implementation

Implementation of R3M necessitates resolving certain issues which are described at the following paragraphs. Once these issues are taken into the account of the design, a modeling engine is provided which is capable of handling real-time simulation. Such modeling engine may be used in quite a vast range of real-time applications where real-time simulation plays a role.

4.3.1 Connection Establishment

An IP connection between the machine on which Simulation Engine runs, and the machine which holds the Data Integration component is sought. Only one connection is established between the two components even if both components run on the same machine. Simulation Engine component receives messages from Data Integration component.

4.3.2 Message Handling

Once a message is received by the Simulation Engine, it meets three consecutive tasks:

1. Decoding the message, the IP packet is opened and the actual message data separated from the metadata.
2. Filtering the message, messages are built to target particular receivers within the simulated model. A unique address is associated with each message to guaranty the unique relationship.
3. Extracting key information from message, depending on the type of the reaction that the triggered receiver must reflect on the process certain information is sent through the message which must be extracted and interpreted.

4.3.3 Entity Handling

When the content of the message is to create or change the status of an entity, two tasks must be accomplished in order to control the access to the entity:

1. Triggering the correct entity inside the model, once all information is ready for the reflection on the receiver process, if the reaction belongs to creation or change of status of an entity, or a bunch of entities, the targeted entities must exist at the corresponding receiver point. Each entity on the simulated model, at the moment of creation, is given a unique identifier similar to for each entity. carrying the serial number of the entity with the message and checking it against the serial number of the entity on the simulated model guaranties the concurrency of the simulated model with the physical environment.
2. Handling entity queues and timings, at certain receiver points where entities need to be checked against a message before progress, arrival entities build up a queue. Upon receiving a message on the receiver point, the queue is searched for the particular entity. The entity which is found is de-queued and released to progress to the process.

5 A HEALTHCARE AND MANUFACTURING SYSTEMS EXAMPLE

5.1.1 Description of the Manufacturing Example

The demo is currently installed in the AMEE lab in Brunel university. A conveyor system is modeled as shown in figure 4. RFID readers and tag system together with light reflection switches are used to generate arrival data. There are six stations representing six processes as an example of a sandwich factory. First three processes model sandwich preparation and are set in series. The last two processes representing the packing stage are set in parallel. The fourth process represent a quality check and division point for the two packing processes. Therefore, two different types of products may be processed. In order to guaranty traceability, each product is placed on a container throughout the whole conveyer system with an RFID tag attached to each container. Consequently, each product corresponds a unique identifier to be scanned at certain points for example at the entry into the conveyer system, and at the division point.

A similar system is modeled in Arena simulation software, as shown in figure 5. R3M is implemented in connection with all six processes as well as the entity arrival point. R3M uses Arena's real-time messaging capability for communicating with Data Integration Module built in Labview software.

Figure 4: A manufacturing conveyer and RFID example

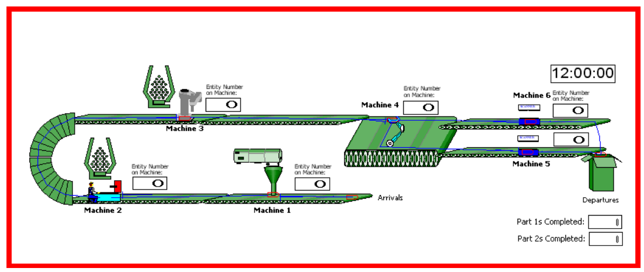

Figure 5: Manufacturing simulation system example

RFID tags and switch status are scanned by data acquisition modules implemented in the Labview software. The input data is pre-processed and reformatted in another part of the Labview software to build input variables to the data processing system, in which structuring the message and storage are done. Structured messages are then sent to the simulation model built in Arena software. In the simulation model the received messages are processed by corresponding processes to whom a message is addressed.

Both generated events in simulation model and historical data stored in Labview software connected database help with calculation of the cost factors. Historical data also helps with curve fitting of the behavior of events and processes for when a 'what-if' scenario is to be tried instead of real-time data.

5.1.2 Description of the Healthcare Example

The manufacturing example was adapted to an existing model of an emergency department to test how the system will work in a labor intensive environment. For the sake of demonstration, both the actual and simulated systems have been modeled as shown in figure 6.

Just as in the manufacturing example, when a patient arrives in the actual system, an arrival message is sent to the simulated system which then creates an entity that represents the patient in the simulated system. As the patient arrives at various stations in the system, messages are sent to the simulated system which moves the entity on accordingly. This is based on the real-time model matching mechanism described n section 4.2.

The current demo uses the messaging element in the Arena simulation software in which the model was developed. Eventually, patients arriving in the system will be given RFID wrist bands which will help track the entire patient journey.

In healthcare, however, some of the data necessary to determine performance cannot be tracked as in manufacturing. Patient satisfaction information for instance is best obtained through patient surveys. A separate system is therefore being developed for online patient survey data acquisition to be integrated into the current framework.

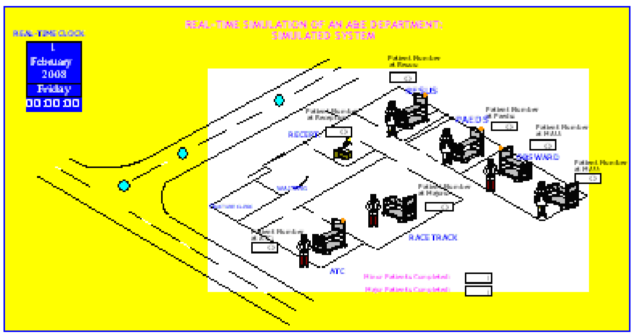

Figure 6: A healthcare simulation system example

6 SUMMARY AND FURTHER WORK

A generic framework for real-time discrete event simulation has been suggested. Starting with a review of existing applications of real-time simulation in manufacturing, the concept of this framework has been presented all the major components of the framework have been explained. Four drawbacks of traditional simulation have been identified which the proposed framework effectively addresses.

To illustrate the operating principles and adaptability of the proposed framework, two examples and in manufacturing and healthcare have been shown. The data requirements of the framework are twofold; first to drive the simulation and secondly to provide information for deriving a cost index in the manufacturing mode and a quality index in the healthcare mode.

Work on the cost and quality indices are currently ongoing and have not been presented here. Also special data collection devices are being developed that will be suitable for the healthcare environment.

REFERENCES

Banks J. 1998. Principles of Simulation. In *Handbook of Simulation; Principles, Methodologies, Advances, Applications and Practice.* Ed. Banks, J. New York, John Wiley & Sons Inc.

Brailsford, S. C. 2007. Tutorial: Advances and challenges in healthcare simulation modeling. 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, 1436-1448. J.W. Marriott Hotel, Washington, D.C.

Dangelmaier, W., K. R. Mahajan, T. Seeger, B. Klöpper, and M. Aufenanger. 2006. Simulation Assisted Optimization and Real-Time Control Aspects of Flexible Production Systems Subject to Disturbances. 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, 1785-1795. Portola Plaza Hotel, Monterey, CA.

Eldabi T., R. J. Paul, and T. Young. 2007. Simulation Modeling in Healthcare: Reviewing legacies and investigating futures. *Journal of the Operational Research Society* 58:262-270.

Gupta, A. K., A. I. Sivakumar, and S. Sarawgi. 2002. Shop Floor Scheduling With Simulation Based Proactive Decision Support. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yucesan, C. Chun-Hung, J. L. Snowdon, and J. M. Charnes, 1897-1902. Manchester Grand Hyatt, San Diego, California.

Harmonosky, M. C. 1995. Simulation-Based Scheduling: Review of Recent Developments. In *Proceedings of the 1995 Winter Simulation Conference*, ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman, 220-225. Hyatt Regency Crystal City, Arlington, VA.

Jones, A., and A. Saleh. 1990. A Multi-level/Multi-layer Architecture for Intelligent Shop floor Control. *International Journal of Computer Integrated Manufacturing* 3(1):60-70.

Komashie A., A. Mousavi, and J. Gore. 2008. Using Discrete Event Simulation (DES) to Manage Theatre Operations in Healthcare: An Audit-based Case Study. In *Proceedings of the 10th Int. Conf. Computer Modeling and Simulation*, Cambridge, UK. IEEE Xplore.

Komashie A., and A. Mousavi. 2005. Modeling Emergency Departments Using Discrete Event Simulation Techniques. In *Proceedings of the 2005 Winter Simulation Conference*, ed. M. E. Kuhl, N. M. Steinger, F. B. Armstrong, and J. A. Joines, 2681-2685. Orlando, Florida.

Lee, K., and P. A. Fishwick. 2001. Building a Model for Real-Time Simulation. *Future Generation Computer Systems* 17(5):585-600.

Lee, S., S. Ramakrishnan, and R. A. Wysk. 2002. A Federatopm Object Coordinator for Simulation Based Control and Analysis. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yucesan, C. Chun-Hung, J. L. Snowdon, and J. M. Charnes, 1986-1994. Manchester Grand Hyatt, San Diego, California.

Manivannan, S., and J. Banks. 1991. Real-Time Control of a Manufacturing Cell Using Knowledge-Based Simulation. In *Proceedings of the 1991 Winter Simulation Conference*, ed. B. L. Nelson, W. D. Kelton, and G. M. Clark, 251-260. The Arizona Biltmore, Phoenix, AZ

Mullarkey, P., S. Gavirneni, and D. J. Morice. 2000. Dynamic Output Analysis for Simulation of Manufacturing Environments. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 1290-1296. Wyndham Palace Resort & Spa, Orlando, FL.

Onut, S., F. G. Gunay, T. Demirel, and H. I. Erdem. 1994. Using Simulation for Shop Floor Control. *Yugoslav Journal of Operations Research* 4(2):195-203.

Potoradi, J., O. S. Boon, J. W. Fowler, M. E. Pfund, and S. J. Mason. 2002. Using Simulation-Based Scheduling to Maximize Demand Fulfillment in a Semiconductor Assembly Facility. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yucesan, C. Chun-Hung, J. L. Snowdon, and J. M. Charnes, 1857-1861. Manchester Grand Hyatt, San Diego, California.

Rabbath, C.A., M. Abdoune, J. Belanger. 2000. Effective Real-Time Simulation of Event-Based Systems. *In Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 232-238. Wyndham Palace Resort & Spa, Orlando, FL

Smith, J. S., and P. A. Brett. 1996. Short Term Scheduling Using Discrete Event Simulation. Department of Industrial Engineering, Texas A&M University, Submitted to Southwestern Bell-Technology Resources, Inc. Available at <http://tamcam.tamu.edu/Research/short.htm> [accessed April 14, 2008].

Son, Y. J., and R. A. Wysk. 2001. Automated simulation model generation for simulation-based, real-time shop floor control. *Computers in Industry* 45:291-308.

Tavakoli S., and A. Mousavi. 2006. Input Variable Monitoring for Quick-Response Decision Making in Volatile Systems. In *Proceedings of Virtual Concept 2006.* Mexico.

Tavakoli, S., A. Mousavi, and A. Komashie. 2008. Flexible data input layer architecture (FDILA) for quick-response decision making tools in volatile manufacturing systems. In *Proceedings of the IEEE, ICC'08 mini-conference*. China: IEEE Xplore.

Viadyanathan, B. S., D. M. Miller, and Y. H. Park. 1998. Application of Discrete Event Simulation in Production Scheduling. In *Proceedings of the 1998 Winter Simulation Conference*, ed. D. J. Medeiros, E. F. Watson, J. S. Carson , and M. S. Manivannan, 965-971. Grand Hyatt, Washington, D.C.

Wu, S. D., and R. A. Wysk. 1988. Multi-pass Expert Control System – A Control/Scheduling Structure for Flexible Manufacturing Cells. *Journal of Manufacturing Systems* 7(2):107-119.

AUTHOR BIOGRAPHIES

SIAMAK TAVAKOLI is a PhD Research Student in the School of Engineering and Design at Brunel University. He received a B.Sc. in Electronic Engineering from Amirkabir University of Technology in Tehran. His current research interests include real-time data acquisition and simulation for decision making tools. Siamak is a member of IEEE and ComSoc. His email address is <Siamak.Tavakoli@Brunel.ac.uk>.

ALIREZA MOUSAVI is a Lecturer in the School of Engineering and Design at Brunel University. His research interests include simulation modeling and analysis, especially techniques to improve simulation analysis in practical applications. <Ali.Mousavi@brunel.ac.uk> and his Web address is <www.brunel.ac.uk/~emstaam>.

ALEXANDER KOMASHIE is a PhD Research Student in the School of Engineering and Design at Brunel University. He received a B.Sc. in Mechanical Engineering from the University of Science and Technology in Ghana and an M.Sc in Advanced Manufacturing Systems from Brunel University. His current research interests include application of queuing theory and real-time discrete event simulation for improving the quality of healthcare, patient satisfaction and decision making. He is a student member of the INFORMS Simulation Society and The OR Society Simulation Special Interest Group. His email address is <Alexander.Komashie@brunel.ac.uk>.