Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

DESIGN, DEVELOPMENT AND APPLICATION OF AN OBJECT ORIENTED SIMULATION TOOLKIT FOR REAL-TIME SEMICONDUCTOR MANUFACTURING SCHEDULING

Chin Soon Chong

Gintic Institute of Manufacturing Technology
Nanyang Technological University
71 Nanyang Drive
638075 SINGAPORE

Appa Iyer Sivakumar

Mechanical and Production Engineering
Nanyang Technological University
Nanyang Avenue,
639798 SINGAPORE

Robert Gay

Electrical and Electronics Engineering
Nanyang Technological University,
Nanyang Avenue
639798 SINGAPORE

ABSTRACT

Real-time scheduling of semiconductor manufacturing operations, semiconductor test operations in particular, is complicated due to the following factors; multi-head resources, multi-level hardware dependency, temperature and hardware criteria, dynamic determination of processing time and indexing time, batch processing and re-entrant flow. A first-of-its-kind, object oriented (OO), discrete event simulation (DES) toolkit, RTMSim++ for real-time simulation-based scheduling applications has been conceptualized, designed, developed to resolve real-time scheduling problems in manufacturing. This paper reviews the work done in the development of RTMSim++ toolkit, and a case study in a real-time scheduling application. The salient features of the toolkit includes flexibility to customize and extend its functionality, real time shop floor status data initialization, and capability for modeling complex resource and process relationships. In the case study, RTMSim++ has been customized to incorporate very company specific heuristic rules, with the objectives of improving delivery performance, equipment utilization and cycle time.

1 INTRODUCTION

Information technology and Internet has transformed the world into a new global economy community, made smaller and more competitive. Anyone in business who is not considering the impact of globalization or the increasing integration of the world's economies will need to deal with the inevitable soon enough (Ferranti 2000). The competition and pressure is severed in the recent global economy downturn. In manufacturing for instance, in order to cope or even succeed in global environment, nimble and highly flexible organizations with low fixed costs is the key. Attaining this requires partnerships with suppliers and customers, and the development of real-time systems that can plan, schedule, and execute an operations strategy effectively (Corney 2002).

The ability to quickly manage and compensate the manufacturing system, due to external and internal disturbances, is becoming an important factor for competition (Jackson and Johansson 1997). To cope with an environment dominated by change and uncertainty it is necessary to quickly react based on the information from decision support systems. The primary focus of our research work is in the real-time manufacturing operation scheduling. It is hoped that through the introduction of real-time scheduler, a manufacturing company can accomplish operations strategy, optimize its goal, and deal with daily disruptions such as broken machines, arrival of a new high priority order, etc in a more effective way.

Simulation-based approach is adopted in our research work to tackle real-time scheduling problems. A real time simulation-based scheduling system takes an accurate factory status update and simulates forward in accelerated time, much like traditional analytical models but with model's starting condition is the current factory status. An advantage of using discrete event simulation for operational support is the ability to perform offline system analysis on the same model. It also provides long-term benefits by using the same model for testing and exploring new ways of improving the manufacturing system. The benefits of using computer simulation are comprehensively documented in simulation-related publications and books such as (Banks et al. 1996) and (Law and Kelton 1991). Simulation-based scheduling systems usually require a two-stage approach. Offline experiments are carried out in the first stage to determine policies and scheduling rules that can achieve the desired performance of the manufacturing system (Yang and Chang 1998). This set of policies and dispatching rules are then used in the second stage to generate the schedules.

Most of the prior work done by others in developing simulation toolkits and packages such as SimPack (Fishwick 1992) and C++SIM (Little and McCue 1994), which are general-purpose simulation, are used more for academic and teaching purposes. BLOCS/M (Douglas et al. 1993) and OOSIM (Douglas and Spiridon 1997) focus on issues in automated operation and control in discrete-part flexible manufacturing systems. A prototype real-time DES of a printed circuit board (PCB) production system for operational support has been built previously (Jackson and Johansson 1997). The main purpose of the work is to investigate the feasibility of DES in supporting the real-time information in production system. The existing academic research efforts in object-oriented manufacturing simulations have also been extensively assessed (Narayanan et al. 1998),

In this paper, an object-oriented toolkit for real-time manufacturing operation scheduling system is reviewed. The review starts from the objectives and justification, challenges, conceptualization, design, development, and its subsequent extension and successful implementation in a multi-national company, in one of the its semiconductor test facilities. Although the focus is on semiconductor backend, the concepts are equally applicable to other complex supply chains.

2 OBJECTIVES AND JUSTIFICATION

Two of the 10 critical capabilities for simulation and modeling identified in the Integrated Manufacturing Technology Roadmap (IMTR) initiative (IMTR 1999) are real time enterprise optimization and interactive models linked to real time information systems. Our objectives are aligned to the scheduling and optimization using real time information and our focus is on manufacturing applications.

Before embarking into developing our very own object-oriented real-time simulation toolkit, a survey has been conducted to evaluate the most suitable simulation software for the scheduling work in mind. Numerous discrete event simulation software tools have been reviewed, from simple spreadsheet based types on the one extreme to sophisticated languages and tools on the other. Most of these tools have execution speed limitations in handling complexity in manufacturing scheduling and real-time data, which are the two most critical success factors in our application. We concluded that these tools are not suitable for real-time simulation of semiconductor manufacturing.

Few of the commercial tools that are considered appropriate for the real-time application suffer from relatively high capital investment and maintenance costs. The tools also require extensive customization to partially fulfill the semiconductor manufacturing requirements. The limitation arises from many aspects including the modeling aspects of initiation, resource relationship definition, alternative resource combinations, dynamic computation of process time based on resource selection, sequence dependent setups, alternative routes, selection of low level resources, and ability to customize scheduling policies, rules and optimization algorithms.

It is finally decided by the team to use the commonly available C++ language as the platform for the development work. An added advantage of using C++ is that object-oriented paradigm has a significant compatibility with discrete-event world-view formalism (Narayanan et al. 1998). By using C++, simulation practitioners who are comfortable with the popular C++ language can immediately begin creating simulations without having to learn entirely new language syntax.

The decision has resulted in the need to design and develop a real-time simulation toolkit focusing initially on specific application in the semiconductor backend operations. The development is also based on the need for scheduling, dispatching rules and optimization algorithms as customizable part of the simulation toolkit. This is an important criterion for providing manufacturer a competitive advantage as the customized policies, rules and optimization algorithms will become sole property to the manufacturer in each application.

3 THE CHALLENGE

The challenge the team confronted in the construction of RTMSim++ arises from three distinct areas: the simulation modeling of complex semiconductor manufacturing operations; the flexibility to allow extensive customization of scheduling policies and rules; and the run time performance. This is described in detailed in the following subsections.

3.1 Semiconductor Manufacturing Modeling

Semiconductors are manufactured in highly specialized facilities. The production processes in these facilities usually has four phases: wafer fabrication, wafer probe, assembly or packaging and final testing. The team's initial focal point is on the modeling test operations of semiconductor backend.

There are a number of specific aspects to modeling the semiconductor backend test operations. The test operations are considered as one of the most complex to model and schedule. Some examples of these complex characteristics are outlined. Each of the highly re-entrant test steps requires additional resources from a multiple choice of secondary equipment (i.e., test handler) and multi-layered hardware (conversion kits, loadboard, dutboard, etc). There are a large variety of test step conditions and alternative routes, based on test equipment, secondary equipment, hardware, test programs, test time, and process parameters such as test temperature. Tester setup may involve any combination of these testing conditions, and it is based on the existing setup and the new setup (sequence dependent setup). Some of the products require burn-in batch process between test steps, involving partial or entire jobs.

The test processing time is made up of test time and index time. Test time is a function of test equipment, the device type and the number of parallel test sites in use. Index time is a function of the handler capability in terms of single or parallel indexing, which depends upon the lower level resources such as, hardware and their capability. Multiple-site hardware attachment to a handling equipment creates more than one site to pick and simultaneously place the parts to the test head.

Testing equipment can have more than one test head, each capable of parallel testing different products. Sum of test time and index time is the process time for a single head test equipment. The processing time on each test head of a multiple head tester depends on the products being concurrently tested on each of the test heads and the indexing times of the lower level resources including handler. In addition co-process constraints makes certain products not concurrently testable on testing equipment, and the constraints can be sequence dependent. The co-process constraints for products also vary from testing equipment to testing equipment depending on the equipment testing capability.

Tested units are often binned into different bin types where the mother lot splits into many sub lots. The subsequent process or test steps may be different for each bin type and may involve lot merging.

These requirements make semiconductor test manufacturing as one of the most challenging environment to model and simulate. In addition to that, the standard constraints such as precedence constraints, routing constraints, machine-eligibility constraints also need to be modeled.

3.2 Scheduling Policies and Dispatching Rules

One of the major drives of our work to design and develop RTMSim++ is the flexibility to extend, and to allow extensive customizability of company specific scheduling policies and dispatching rules in the toolkit. The team believes that scheduling problems in practice have multiple objectives that can vary significantly from company to company. The overall goal normally involves a weighted combination of several objectives.

This conceptual capability for real-time simulation comes from the team's past experience in developing and implementing semiconductor manufacturing related simulation projects. It can be concluded that no two manufacturers have the same operation strategy or scheduling policies. This is even true for different manufacturing facilities of the same manufacturer. It is thus important to provide easily customizable features in the toolkit for specific scheduling policies and dispatching rules.

Object-oriented programming (OOP) allows a one-to-one mapping between objects in the manufacturing system being modeled and their abstractions in the simulation model (Narayanan et al., 1998). OOP is considered as a more natural mapping paradigm for simulation application, however an amount of effort is still needed to understand the object-oriented simulation (OOS) framework for complex application before anyone can customize scheduling policies or dispatching rules. The required effort will hinder engineers to test out new policies or dispatching rules. Hence, a careful design to isolate the customizable part from remaining OOS framework details is also considered important by the team.

3.3 Real-Time Simulation Performance Criteria

In real-time application, the execution speed of the model is an important criterion for the successful deployment of the application. The manufacturing equipment and resource data including secondary resource and low level hardware, co-process, sequence dependent setup, preventive maintenance schedule of the resources, lots for scheduling, work-in-progress, and real-time shop floor status must be retrieved, checked for data consistency, corrected for data inconsistency. The manufacturing model will be executed with manufacturer specific policies and rules, and reports generated at the end of run. The schedule will be post-processed and the schedule is transferred to the screen display unit of the equipment on the shop floor.

In a typical semiconductor manufacturing dealing with the stated manufacturing complexity, hundreds of testing equipment, and thousands of lots, sophisticated scheduling policies and rules, simulation time duration of weeks, a simulation run with performance of tens of minutes is considered acceptable on a Pentium II PC running Windows NT.

4 RESEARCH AND METHODOLOGY

The research objectives and efforts of the team are motivated and directed by the following issues:

- **Natural mapping** – develop an OOS framework with natural mapping to physical manufacturing objects for representing the interactions between parts and processes for manufacturing systems.
- **Customizability** – develop an OOS modeling environment that permits programming-free model creation and problem solving approaches in a single base model.

- **Efficiency** – develop simulation models so they run efficiently.
- **Reusability** – develop a reusable library of classes to support modeling of manufacturing systems.
- **Ease of maintenance** – make the class library more reusable and easily comprehensible, and simplify the description of complex systems.

Object oriented analysis and design technique using Unified Modeling Language (Moore 1997) is applied throughout the development of RTMSim++. Representing complex behavior in a simulation model always has been a challenge (Narayanan et al. 1998). In a search for better ways to represent complex manufacturing behavior, a multi-layered OOS software architecture is used in the design. This is considered as the most promising architecture as it allows multi-level customizing flexibility from data driven model at the highest level to language programming at the lowest. A more detailed explanation is in the following subsections.

4.1 Multi-Layered OOS Software Architecture

To cater for different user groups of RTMSim++, a six-layered OOS software architecture is designed as depicted in Figure 1. From the highest level of data driven simulator to the lowest level of foundation and C++ libraries. The flexibility to customize the toolkit ranges from limited customizability for novice users to total extendibility for programmers. The ease of customizability goes in the reverse direction from difficult to easy.

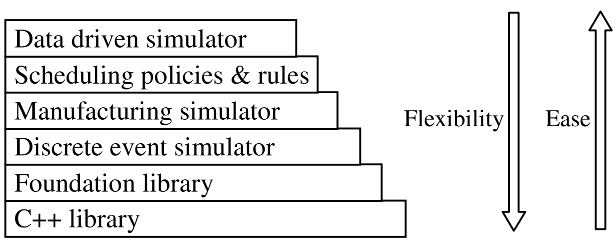

Figure 1: Multi-layered Object-Oriented Architecture

At the first layer, the model is driven entirely by data. A set of standard database tables has been designed to capture all the manufacturing data required to model and simulate the manufacturing systems. The most important tables are listed as shown:

- **Equipment** – defines all equipment data, which includes testing equipment, handler, hardware, equipment group and family in the manufacturing system.
- **Equipment Status** – defines real-time shop floor status for all equipment, with lots, handler and hardware details.
- **Equipment Event** – defines shop floor event and its duration, for example preventive maintenance, engineering lot test, etc.
- **Process** – defines the process routing for all part numbers, with other information such as equipment preference, process time, yield, and so on.
- **Binning** – defines the binning requirement for mother lot that splits into many sub lots after testing.
- **Co-process** – defines the co-process constraints for testing equipment.
- **Setup** – defines the sequence dependent setup details.
- **Inventory** – defines all the lots with its part number, process and step information in the system.
- **Relation** – defines the eligibility and preference of handler and hardware, and the corresponding indexing time.
- **Parameter** – defines all the control variable parameters for the simulator.

An auto model generator can be built to map the existing data of manufacturing system to the pre-defined standard tables of the simulator. The goal of this effort is to make the architecture more useful to the actual users, while satisfying the need for rapid prototyping. Beyond this layer, knowledge of C++ is required for toolkit customization. However, reusable classes are implemented and stored in a software library. These classes can facilitate rapid model development, with minimum knowledge of the underlying language.

At the second layer, a centralized rule control class is defined to manage the dispatching rules. Company specific scheduling policies and rules can be integrated into the toolkit through object inheritance of the class. Policies and rules can be constructed with little knowledge of the entire OOS framework. Customization of rules can be done through OO inheritance from "QueueManager" (refer to Figure 2).

At the third layer, various manufacturing system entities, and the coupling mechanisms to establish the interrelationships between manufacturing entities are implemented. The behavior of an entity over time during a simulation is realized at this layer through events. The events are represented procedurally as methods of classes. At this layer, new type of equipment can be incorporated by inheritance from the original class definition.

At the fourth layer, the basic structure of discrete event simulation is defined. The necessary simulation infrastructure, such as scheduler, event queue, containers for simulation entities, simulation clocks, serialized input and output are defined. In RTMSim++, simulation processes execute according to their simulation time. Only one process executes in any instance of real time, but many processes may execute at any instance of simulation time. Those processes, which are currently inactive, are put on to a simulation event queue, which is arranged in increasing order of simulation time. The simulation scheduler is used to manage the execution of the processes. At this layer, existing classes can be replaced as long as they conform to the original class definition. Using framework at this layer, a completely new domain can be developed. Examples are logistic, supply chain, and service-based simulation.

At the fifth layer, the C++ foundation library for simulation is defined. This is an extension of the standard C++ library to manage discrete event simulation entities. At this layer, classes such as priority queue, sorting, string match, input and output are defined. The sixth layer is the standard library provided by C++.

4.2 Manufacturing Classes and Hierarchy

The team focuses on the development of a library of reusable classes. The classes are designed according to the one-function-one-object principle, in which each object does only one thing. This makes for simple classes. The number of class hierarchy levels is kept to minimum to allow greater reusability of the object classes, and to make learning easier. The major OOS classes and hierarchy is shown in Figure 2 and 3. Only the most superficial description of the architecture is given. Interested reader should contact authors for additional details.

Sub-classing is used to develop objects with new behavior from pre-existing classes. For example, a resource object (e.g., Tester-2) that has a multi-head capability for testing is sub-classed from a machine class (e.g., ProcResource) that does not exhibit this behavior.

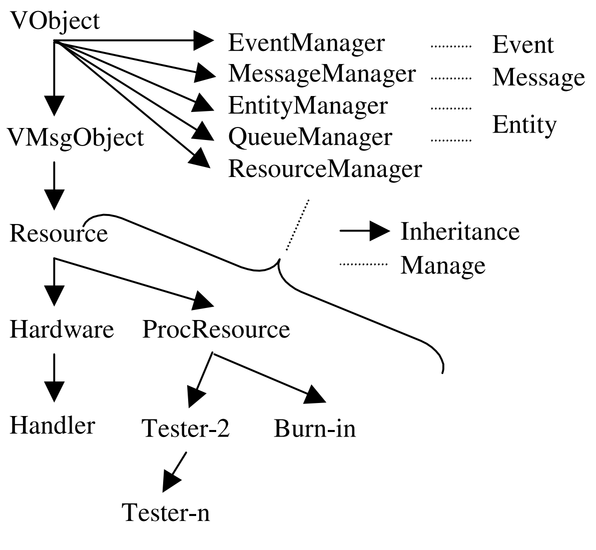

Figure 2: Major OOS Classes and Hierarchy

Information Classes

ResourceInfo
ResourceStatus
ResourceEvent
Process
Binning
Co-process
Setup
Inventory
Relation
Products
ResourceMatrix

Figure 3: Information Classes

RTMSim++ has four fundamental abstractions to represent objects in manufacturing: lots or jobs (i.e., entity in RTMSim++ terminology), resources, tasks, and routes. Lots comprising the work-in-process (WIP) inventory move from one resource to another, where operations are performed. An operation sequence specifies the route for a lot. Physical elements such as resource are distinguished from the decision-making entities such as "QueueManager" in the factory. In addition, information about a manufacturing system (e.g., a process plan) is captured in a set of database classes (e.g., "ResourceInfo", "ResourceStatus", etc.).

The two important issues in the architecture: the coupling mechanism between class instances and the specific behavior representation are addressed in some details in the following subsections.

4.2.1 Coupling Mechanisms

The coupling mechanisms in the modeling architecture dictate how to establish the interrelationships between manufacturing entities or resources (Narayanan et al. 1998). In RTMSim++, all interactions between resource classes occur through message and event "points" (i.e., specific methods) in each class. Messages in RTMSim++ are object instances of non-simulation-time related classes encapsulated with subject matters and other related details that are passed among entities. These interactions are "instant" (i.e., immediate) in simulation time. Events are similar to messages in the contents except for they are linked to simulation-time. Controllers (i.e., managers in RTMSim++ terminology) are structured using the client-server paradigm and each controller responds to its clients via messages or events. For example, a resource object may send a message to object "EventManager" to schedule a down event on the resource itself 30 simulation time unit later.

4.2.2 Behavioral Specifications

The behavior of an entity or resource over time during a simulation is dictated by the behavior specification of the modeling abstractions and is realized through events. In RTMSim++, events are represented procedurally as methods of classes. For example, consider a resource which upon receiving a job, begins with a load operation, followed by a setup, processing, complete processing, and then unload operation. Methods are defined for each of the operations. Multiple resources with the same behavior but different data (e.g., processing time) can be modeled by creating instances of the class.

5 CASE STUDY: REAL-TIME SCHEDULING FOR A SEMICONDUCTOR BACKEND COMPANY

A case study to implement a scheduling system using RTMSim++ for a semiconductor backend facility has been carried out. The purpose of this project was to schedule and optimize the semiconductor test operation with the following primary objectives: improve delivery accuracy, maximize resource utilization, and minimize cycle time. The scheduling constraints considered are already described in Section 3.1.

5.1 Background

The facility manufactures over 500 million IC's per year. The varieties include ASIC's, PC products, telecommunication, and many other types of IC's for local as well as overseas markets.

The study was made in the test operation department for a major product group. There are about 50 test equipment, 130 handlers, and 1300 hardware consists of conversion kit, loadboard and dutboard. About 2000 lots to be scheduled at any instant of time.

5.2 Project Description

The project duration is about 5 months, and most of the efforts focus on the requirement specifications, and the design of the company policies and dispatching rules. The rules are customized through the "QueueManager" class by inheritance. The customized rules involve more than 8 levels of priority categories based on the important of the rules to the company. The examples range from lot types (e.g., tag lot, urgent lot, etc.), lot location (e.g., assembly lot, test only lot, etc), conversion and setup priority, burn-in, test steps, sophisticated optimization weightage factors that are computed based on dynamic information in the package.

The configuration of the implemented system is shown in Figure 4. The system implemented at the facility automatically triggered by the company's internal systems and dynamically generates schedules and reports for the testing machines at a specified frequency, anytime, throughput the year. The company information system provides status of machines, status of current work-in-process inventory and exact location of each WIP lot, the process flow, the relationship and preference of the required primary and secondary hardware, setup and conversion, co-process, schedule for preventive maintenance and other scheduled activities that make machine unavailable within scheduling time window. The other shop floor real-time status is obtained from the status update PCs through barcode.

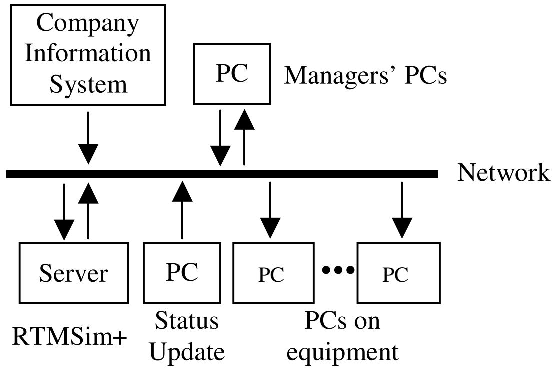

Figure 4: RTMSim++ System Configuration

RTMSim++ is installed at the server PC running Windows NT Pentium workstation. The system takes less than 10 minutes to generate the schedule. The schedule generated by the system will be displayed on the PC monitor of the test machines after each scheduling run. Shop floor operation staff makes use of the dynamic lot selection decision generated by the system.

A client-server application has been implemented by the company that enables user-friendly system setups and easy access to the reports generated by the system. This application runs on the managers' PCs.

5.3 Results

The company has realized tangible and intangible benefits from this implementation. Intangible benefits include the achievement of high delivery accuracy, higher tester utilization, a high cycle time repeatability, highly accurate response to customer on order completion time, and rapid and accurate capability of "what if" analysis. Another significant benefit is that the company is able to accurately model their manufacturing system using the system.

6 CONCLUSIONS

This paper presented the concepts, and application of an OO real-time simulation system. The developed system has the flexibility to be customized in most relevant aspects and extendable to most manufacturing environments. Created simulation models are true reflection of critical resources and resource relationships in advanced manufacturing systems and resolves identified limitations of commercially available simulation packages.

One of the limitations identified is the C++ skill requirement for customization. It is also identified that availability of accurate and consistent real time data in manufacturing environment is crucial for real-time simulation and therefore currently limits the application to environments such as semiconductors, where advanced manufacturing information systems are available.

The research and application demonstrated advancements particularly in the following aspects:

- Real-time simulation is achieved using real time integration of manufacturing information systems with OO simulation model.
- The central contribution to the outcome is the OO design and approach to the simulation toolkit.
- Automated dynamic creation of the simulation model is achieved, eliminating the need for highly trained specialists for maintenance.
- The OO approach enabled complex customized scheduling and optimization algorithms to be integrated with discrete event simulation.

The results of the simulation package are encouraging and the team intends to develop it further in light of continued experience.

ACKNOWLEDGMENTS

The team gratefully acknowledges the General Manager of Infineon Technologies AP, and his management team for providing the opportunity and support to test and implement the concepts.

REFERENCES

Banks, J., John S. Carson, II, Barry L. Nelson, 1996. *Discrete-Event Simulation*, Second Edition, 4-18. Prentice Hall.

Corney, William, Global Manufacturing. APICS, The performance advantage, Feb 2002, 33-34.

Douglas A. Bodner, Suzanne Dilley-Schneider, S. Narayanan, UdaySreekanth, T. Govindaraj, Leon F. McGinnis and Christine M. Mitchell, Object-Oriented Modeling and Simulation of Automated Control in Manufacturing, In *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, May 1993, Vol. 3, 83-88.

Douglas A. Bodner and Spiridon A. Reveliotis, Virtual Factories: An Object-Oriented Simulation-Based Framework for Real-Time FMS Control, In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation*, Sept. 1997, Los Angelos.

Ferranti, Marc, The new global economy: Bracing for the wave, *InfoWorld* 11/27/00. Available online via <http://www.itworld.com/Tech/2418/IW001127hnglobal/pfindex.html> [accessed April 1, 2002]

Fishwick, P. A., Simpack: Getting started with simulation programming in C and C++. In *Proceedings of the 1992 Winter Simulation Conference*, 154-162.

IMTR 1999. Integrated Manufacturing Technology Roadmap, *Modeling & Simulation*, Pub. IMTR, Oak Ridge Centers for Manufacturing Technology, Tennessee, 1999.

Jackson M., Johansson C., 1997. Real Time Discrete Event Simulation of a PCB Production System for Operational Support, In *Proceedings of the 1997 Winter Simulation Conference*, ed. D.J. Medeiros, E.F. Watson, J.S. Carson, and M.S. Manivannan, 832-837.

Law, A.M., Kelton, W.D., 1991. *Simulation Modeling and Analysis*, 114-115, 298-323. New York: McGraw-Hill, Inc.

Little, M. C., McCue, D. L., Construction and Use of a Simulation Package in C++, Computing Science Technical Report, University of Newcastle upon Tyne, Number 437, July 1993 (also appeared in the *C User's Journal* Vol. 12 Number 3, March 1994).

Moore, G., 1997. Keynote address- Manufacturing the bridge to the next millennium, 1997. In *IEEE International Symposium on Semiconductor Manufacturing*, San Francisco.

Narayanan, S., D. A. Bodner, U. Sreekanth, T. Govindaraj, L. F. McGinnis and C. M. Mitchell. Research in Object-Oriented Manufacturing Simulations: An Assessment of the State of the Art, In *IIE Transactions*, Vol. 30, 795-810, 1998.

Yang, Y., Chang, T.S., 1998. Multiobjective scheduling for IC sort and test with a simulation testbed, *IEEE Transactions Semiconductor Manufacturing*, 11:304-315.

AUTHOR BIOGRAPHIES

CHIN SOON CHONG obtained his degree in Electrical and Electronics Engineering from the City University of London, UK. He joined Gintic, a research institute in Singapore, ten years ago and is currently in the Manufacturing Planning and Scheduling Group of Manufacturing Information Technology Division. He obtained his Master of Engineering in Computer Integrated Manufacturing from Nanyang Technological University, Singapore. During these ten years, he has been involved in simulation, scheduling and optimization related projects in logistic and manufacturing IT domains. The industrial projects include cargo container operation and yard capacity planning simulation for a container port, dairy-printing process simulation for a printing company, manufacturing cycle-time modeling, scheduling and optimization for various MNCs. His current research interest includes intelligent and integrated simulation, planning, scheduling, optimization in the area of manufacturing, logistic, and supply chain. He can be reached via email at <cschong@gintic.gov.sg>.

APPA IYER SIVAKUMAR (**Senior member IIE**) is an Associate Professor in the School of Mechanical and Production Engineering at Nanyang Technological University (NTU), Singapore and a Fellow of Singapore Massachusetts Institute of Technology (MIT) Alliance (SMA). Prior to this he was at Gintic Institute of Manufacturing Technology, Singapore. His research interests are in the area of simulation based optimization of manufacturing performance, supply chain, and dynamic schedule optimization. Prior to joining Gintic in 1993, he held various management positions including technical manager and project manager for nine years at Lucas Systems and Engineering and Lucas Automotive, UK. He received a Bachelors of Engineering from University of Bradford, UK and a PhD in Manufacturing Systems Engineering from University of Bradford, UK. He has been the technical chair and co-edited the proceedings of the 3rd and 4th International Conference on Computer Integrated Manufacturing (ICCIM '95 and ICCIM'97), Singapore. His email and web addresses are `<msiva@ntu.edu.sg>` and `<www.ntu.edu.sg/home/MSiva/>`.

ROBERT GAY obtained his B. Eng, M. Eng and PhD degrees at the University of Sheffield, England, in 1965, 1967 and 1970 respectively. He was awarded the Grouped Scholarship in Engineering and Metallurgy by the University of Sheffield from 1967 to 1970. Since obtaining his PhD, he has been involved in Education and R&D working in institutions such as Singapore University (1972-1979), Rutherford and Appleton Laboratory (England, 1979-1982), NTU (1982-1995 and 1999-present) and Gintic Institute of Manufacturing Technology (1989-1999). He has also been actively involved in promoting innovation in Singapore through work in various committees: Science Quiz (MOE), Science Centre Board, National CAD/CAM (NCB), Tan Kah Kee Young Inventors Award (TKK Foundation & NSTB), National IT Plan (NCB), Technopreneurship incubation center (ITE), Commercenet Singapore, Singapore Computer Society. Currently Professor Gay is in the School of EEE at NTU and also Director and CEO of the ASP Centre. He has more than a hundred publications in journals, conference proceedings and books. His email and web addresses are `<eklgay@ntu.edu.sg>` and `<www.ntu.edu.sg/eee/icis/cv/robertgay.html>`.