Proceedings of the 1997 Winter Simulation Conference
ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson

FLEXIBLE MODELING OF MANUFACTURING SYSTEMS WITH VARIABLE LEVELS OF DETAIL

Tillal Eldabi

Centre for Applied Simulation Modelling
Department of Information Systems and Computing,
Brunel University, Uxbridge, Middlesex UB8 3PH, U.K.

Ray J. Paul

Centre for Applied Simulation Modelling
Department of Information Systems and Computing,
Brunel University, Uxbridge, Middlesex UB8 3PH, U.K.

ABSTRACT

This paper presents an investigation of three simulation packages: VS7, SIMAN/CINEMA IV, and SIMFACTORY II.5. These packages were investigated with regard to their capabilities of modeling problems related to a Manufacturing Systems Design (MSD) Framework, which involves different levels of detail. These levels are the Conceptual Modeling level and the Detailed Design level. The investigation is based on a case study which relates to manufacturing systems. The main objective of this investigation was to examine the existing manufacturing simulation packages and their abilities to offer variable detail modeling. The paper also suggests that it may not be possible to have an existing simulation environment that offers flexible facilities for variable detail modeling of manufacturing systems. The paper presents a method of data arrangement that may be able to help existing simulation packages cope with problems of detail variability.

1 INTRODUCTION

The manufacturing systems design process is divided into two levels of detail, the *conceptual modeling* phase and the *detailed design* phase. The first one relates to developing the basic principles by which the system will work, whilst the second one relates to providing a detailed account of what is required (Wu 1992a; Ülgen, Onur, and Sanjay 1997). Simulation is considered a very important computer aid to the design process, due to the increased complexity of manufacturing systems, and due to their dynamic and stochastic behavior (Carrie 1988; Kochhar 1989; Law and Haider 1989).

The main purpose of the research is to study and analyze the ability of existing simulation software to offer variable detail modeling during the design process. That is, to study the capability of simulation software in modeling the conceptual phase in the manufacturing design process, and then increase the level of detail from conceptual to detailed models without repeating the modeling and data collection effort.

This research is divided into two main aspects. The first one is to establish some features to assess simulation software with regard to manufacturing systems with different levels of detail. Based on this, existing simulation packages are then investigated to examine their ability to model different levels of detail. The second aspect is to establish some guidelines for analysts to arrange system information to enable the package to perform the modeling process for systems with variable levels of detail.

2 CAMSD AND MSD

The Computer Aided Manufacturing Systems Design (CAMSD) Research Group at Brunel University aims to develop a formal framework for Manufacturing Systems Design (MSD) which will be implemented within a flexible IT environment. In this way, MSD engineers will be supported by a complete CAMSD solution using any chosen MSD methodology (CAMSD 1994).

There are two common approaches to systems design, top-down and bottom-up. Top-down begins by creating the objectives, then creating a system model which fits these particular objectives, paying less attention to the current situation. A bottom-up approach is based on the existing system, producing a design which requires less capital investment.

The MSD approach, introduced by Wu (1992a), is a combination of the top-down and bottom-up approaches. It initiates the project by analyzing the current and the desired future positions. A set of objectives can be identified by analyzing the desired future position under the constraint of current position analysis findings. The structure of the design process, as shown in Figure 1, can be summarized as follows: The first two stages are *analysis of situation* and *setting of objectives*. These require analysis of the current state of the manufacturing organization. They initiate an analysis of current markets and their future prospects. The next steps in the framework are design phases which transform this operation from the current state to the desired state.

The next stage is *conceptual modeling*, that is, identifying the building blocks or the overall structure of the new system which will achieve the desired results. These blocks will be a combination of manufacturing functions and their relationships, together with necessary controlling functions. At this level, higher-level specifications must be developed for the function and data design. In addition, the long term production capacity to be achieved will also have been specified in terms of average static capacity levels and levels of variation which reflect the dynamic requirements (Schroeder 1993; Tanner 1985; Vanderspek 1993).

The completion of the conceptual design of the system creates a system model in terms of related functions. It contains a set of inter-related manufacturing functions each with a related list of products. There is also a hierarchy of control systems which will process key information associated with the effective performance of these functions. In summary, the detailed design phase takes the conceptual model of the selected manufacturing systems options and transforms this into a detailed specification which can be used later for implementation. *Evaluation* and *decision* phases take place after the conceptual modeling and the detailed design. At this level the design team selects the best available business option with regard to their objectives.

MSD is a formal method or a systematic approach that is used to design a manufacturing system. The CAMSD objective is to produce a framework to support manufacturing engineers in the design of production operations. The two main characteristics of such a framework are: Firstly, it must be flexible enough to address the variety of situations encountered in manufacturing system design, while being systematic, pragmatic, and practical. Secondly, it should be presented to engineers in a form that they will be willing to use in practicing, focusing, and guiding their activities (CAMSD 1994; Fritz et al. 1994).

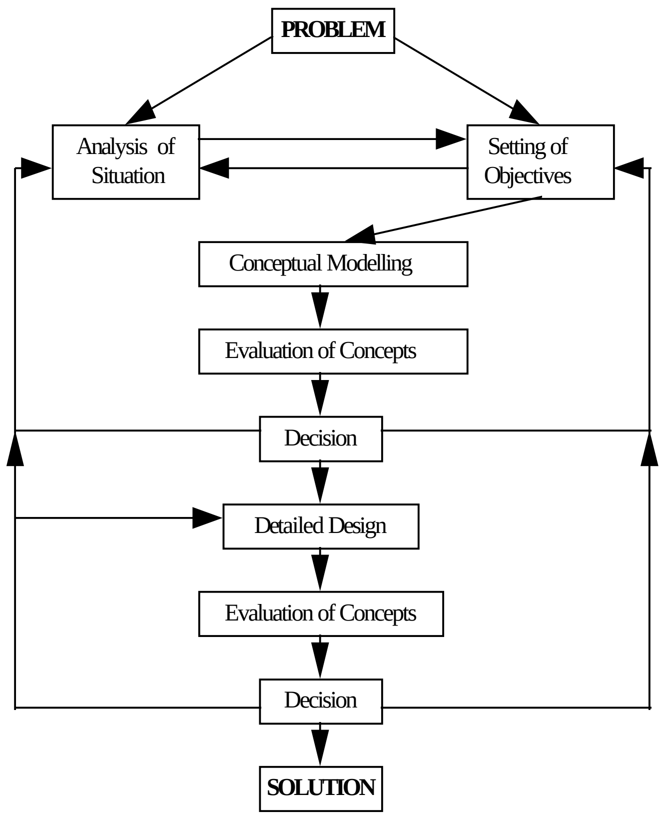

Figure 1: MSD Framework (Wu, 1992a)

3 SIMULATION IN MANUFACTURING

Kochhar (1989) states that the simulation technique has been used since the sixties as a tool to investigate the underlying behavior of many different types of manufacturing systems. During the last three decades there has been a dramatic increase in the use of simulation to design and optimize manufacturing and warehousing systems (Hollocks 1992). There are many reasons for the increase in use of simulation in manufacturing. *Firstly*, increasing competition as a result of greater emphasis on automation to increase productivity, quality, and reduce costs, has led to an increased complexity which can be analyzed only by simulation. *Secondly*, there has been a large scale reduction in the cost of computer hardware required to run the simulation models, in addition to the availability of advanced simulation software. *Thirdly*, the introduction of animation has resulted in a greater understanding of simulation by non-simulationists such as managers and manufacturing engineers.

4 SIMULATION SOFTWARE

During the early days of the application of simulation techniques, many simulation models were created using high level programming languages such as FORTRAN and Pascal, or general purpose simulation languages such as GASP, GPSS, SIMSCRIPT, SLAM, and SIMULA (Kochhar 1989). However, during the last two decades, many simulation software tools have become commercially available, which require less or no programming effort and experience to use. Examples of these tools include SIMFACTORY II.5, ProModel for Windows, AutoMod II, and WITNESS.

Law and Kelton (1991) present the features desired in simulation software, which can be summarized as follows:

- Generating random numbers from the uniform probability distribution.
- Generating random values from a specified probability distribution.
- Advancing simulation time.
- Determining the next event from the event list and passing control to the appropriate block of code.
- Adding records to, or deleting records from, a list.
- Collecting and analyzing data.
- Reporting the results.
- Detecting error conditions.

4.1 Types of Simulation Software

The features mentioned in the previous section are needed in a simulation package to reduce the time spent in programming and to ease the process of model building. Generally, simulation packages have a wide variety of features and characteristics. For example, some packages require coding effort, while some others need little or no programming effort. There are many different methods of classifying simulation software into different types or groups. However, this paper concentrates on one method of classification, where simulation software tools are classified into three categories: *general purpose simulation languages*, *data-driven simulators,* and *program generators* (Hlupic and Paul 1993).

4.1.1 General Purpose Simulation Languages

A general-purpose simulation language is a simulation package which is used for modeling different types of systems with different characteristics. Law and Haider (1989) state that some of these languages may have special features for manufacturing such as workstation and material handling modules. For example, AutoMod II is specifically directed towards material handling and manufacturing problems. Another example of such a package, SIMAN/CINEMA IV, has special material handling features, such as forklifts and conveyors (Pegden, Shannon, and Sadowski 1990). In this research, we choose SIMAN/CINEMA IV to represent this category.

4.1.2 Data-Driven Simulators

Law and Haider (1989) define a data-driven simulator as a computer package that allows the modeller to model systems with little or no programming. Many data-driven simulators are domain-specific (Pidd 1992). They are used to model systems with specific features (e.g. cellular manufacturing systems). There are simulators currently available for certain types of manufacturing, computer, and communication systems. Examples of simulators which are dedicated to manufacturing simulation are: SIMFACTORY II.5, WITNESS, ProModel for Windows, and Xcell+. A graphical user interface is a fundamental part of simulators, which is used for modeling as well as for running the model. Most of these simulators employ a network in their underlying concept (Pidd 1992). Thus entities are assumed to flow through a network from node to node. At these nodes they may be delayed as they engage in activities with entities and resources placed on the nodes. The resources placed on the nodes may also be engaged in endogenous activities. For example, in a manufacturing application such entities can be machines which occasionally fail. In this research, SIMFACTORY II.5 is selected to represent this type of software (SIMFACTORY II.5 1993).

4.1.3 Program Generators

Program generators are used as another way of making simulation more accessible to non-computer specialists. A program generator is a computer program which itself generates another program. Unlike a compiler, which takes a source program written in a problem-oriented language and produces machine code, a program generator takes a system description and produces source code. This generated source code may then be compiled or interpreted to present a computable simulation model (Pidd 1992). Examples of program generators are CAPS/ECSL, VS7, and DRAFT. Program generators are usually interactive and accept a description of a conceptual model such as an activity cycle diagram (ACD) (Carrie 1988). The ACD is a graphical representation which depicts the cycle of each type of entity. Each life cycle is a closed cycle of alternating activities and queues (Paul and Balmer 1993). Most program generators require definition of the model entities. activities, queues, attributes, and priorities. Thus the user starts modeling by drawing an 'ACD,' and then describes the components of the diagram to the program generator. Generally, features of program generators lie between those of simulation languages and data-driven simulators. VS7 is taken in this research as a representative of this software category.

5 ASSESSMENT CRITERIA

The method of analysis followed in the research was an assessment of how well the investigated packages can meet certain criteria, with regard to detail variability during model building and running. This section identifies the requirements from simulation packages to model both the *conceptual* modeling level and *detailed* design level of MSD. The first sub-section presents the assessment criteria for the conceptual level while the second one provides criteria for the detailed modeling.

Table 1: Assessment Criteria for Modeling Conceptual Model and Detailed Design Levels

Conceptual Modeling Features	Detailed Modeling Features
Quick and Simple Model Building	Detailed Model Building
Running-Speed	Automatic Batch Run
Low-Level Animation	High-Level Animation
Total Output	Manufacturing Features
Cell Utilization	Total Output
Parts Life-Time	Statistics of Queuing Lengths and Times for Cells and Transporters

5.1 Conceptual Modeling Criteria

The conceptual level design needs to be quick and simple. It contains the building blocks of the system without involving many details. It is used for high-level analysis and long-term planning. Therefore, the modeling ability assessment criteria, shown in Table 1 above, at this stage of design can be characterized using the following points:

- *quick and simple model building*:

At this stage the conceptual model is supposed to be simple as possible, as it contains the major blocks of the system. Also, modeling needs to be quick in order to carry out the design process in further detail without delay. Regarding simulation packages, this feature (quick modeling) may be affected by factors such as the time required to learn the simulation package in use, the complexity of model building in such a package, compilation, and debugging.

- *running-speed*:

This feature is important because it gives a clearer idea about the system in the long term, which helps in strategic planning. In this case, the simulation package is supposed to be able to run models with long durations of simulated time in less amounts of real time.

- *low-level animation*:

Animation in simulation is generally used for model debugging, verification and validation, and analysis. At this stage of the MSD, low-level animation is suitable. Low-level animation, or level I animation (Johnson and Poorte 1988), can be characterized as a two-dimensional animation which includes only color changes as indicators for changes in the state of entities. Icons used in this type of animation are abstract or basic geometric shapes. Generally, this level of animation displays the logic of the model without including physical layout, such as the real positioning of cells.

- *total output*:

Total output is desirable at this stage to measure the performance of the system as a whole. It is used to compare the different configurations of the conceptual model. That is, to examine which configuration produces the highest output amongst the others, where all configurations are assigned the same types of resources.

- *cell utilization*:

This feature is needed to examine the individual performances of different cells. It could be used as a comparison factor between different configurations. For example, if total outputs are equal for every configuration, the analyst may refer to cell utilization results to determine which configuration could produce the same output with lower utilization.

- *parts life-time*:

This feature is considered a very important method for measuring the performance of the system, as it is used to measure the time spent by a part in the system from the arrival time until all its corresponding processes are finished. This gives a good view about the process routing and helps in comparing different configurations of the system.

The above points represent the modeling features of a conceptual level of the MSD Framework. The first three points represent the modeling requirements, while the last three points represent the desirable results for analyses at this stage.

5.2 Detailed Modeling Criteria and Analysis

The detailed design level is needed to decide the detailed layout of the plant by selecting and allocating the required equipment. Modeling ability assessment criteria, shown in Table 1 above, at this level can be characterized below.

- *detailed model building*:

The main feature of a model at this level is that it should contain all details about the system specifications. The model must be as precise as possible, including all details about cell components and transportation behavior. The detailed model is supposed to be extended from the conceptual model by adding details such as variation of machines number in each cells, number of transporters, loading times, and unloading times.

- *automatic batch run*:

Generally, the detailed design is concerned with the daily or weekly performance of the factory. For this reason, simulation run-length is supposed to be a short period, usually between two weeks to one month. As the model at this stage is supposed to be precise, a number of independent replications should be made. This is to ensure that the results are accurate by quantifying variation due to sampling fluctuations.

- *high level animation*:

At this stage animation is important for analysis as well as communication with other people who are not expert in simulation, such as managers, manufacturing engineers, or clients. Animation of detailed models should be more detailed and it must be meaningful, so that anyone can make sense of it. Usually the type of animation used here is level II animation (Johnson and Poorte, 1988). Level II animation is characterized as a two-dimensional system with movements of objects. Icons and symbols in some way depict the real parts.

- *manufacturing features*:

It is very important at the detailed design level to have manufacturing facilities within the package used. Such features enhance the process of modeling a manufacturing system in less time. These features could be machines, transporters, AGV, and conveyors (Cheng 1985).

- *total output*:

Total output is required at this stage to compare the performances of different configurations of the same detailed model. For example, to examine the effect of variation in the numbers of machines and transporters on each configuration.

- *statistics of queuing lengths and times*:

This feature is needed to detect which cells have large rates of accumulation of parts. This helps in identifying bottlenecks in the system, and in deciding where to build storage points and which cells need more machines. In addition to that, this feature is required to add or delete one or more transporters based on the queuing statistics for transporters.

The above points represent the modeling features of the detailed design level of the MSD Framework. The first four points represent the modeling requirements, while the last two points represent the desirable results for analysis at this stage.

It is worth noting that this research concentrates on the criteria for selecting manufacturing simulation packages that can model problems with different levels of detail such as the Conceptual Model level and the Detailed Design level of the MSD. The criteria outlined in this section may be added to other groups of selection criteria in order to select a suitable package. The reader is referred to (Hlupic and Paul 1995; Hlupic and Paul 1996; and Hlupic 1997) for other evaluation and selection criteria.

5.3 Results and Analysis

Findings and results of modeling for each package were collected and matched with the previously identified modeling features for both levels of details. In this analysis, the higher the ability of a package to meet *both* groups of modeling features, the higher will be its ability to offer modeling variability. Tables 2 and 3 show summary results of the analysis of the three simulation packages with respect to both levels of detail of the MSD, respectively. The ability of each package to match each of the modeling criteria is represented by the numbers 1, 2, and 3: '1' represents absence or poor matching with the particular feature; '2' represents fair quality of matching; '3' represents excellent quality of matching. A case study is used for this analysis which was based on a multi-level manufacturing problem taken from Wu (1992b).

It can be seen from Tables 4 and 5, as a result of our case study, that there is no perfect match between the packages and the features of variable detail modeling. Generalizing these results it can be said that some of the simulation packages can efficiently be used for conceptual level modeling, but inefficiently expand the models into greater detail. On the other hand some packages might be too complex when used for modeling conceptual levels, but after that they can easily expand the model into more detail. However selecting an appropriate package may depend upon the modeling requirements. Section 6 presents some guidelines which could be helpful for building models with detail variability and ease the process of model building with existing simulation packages.

Table 2: Results of Analysis of the Simulation Packages with Respect to the Conceptual Modeling Features

Packages	SIMAN/CINEMA IV	SIMFACTORY II.5	VS7
Quick and Model Building	1	3	2
Running-Speed	3	1	2
Low-Level Animation	1	3	2
Total Output	2	3	2
Cell Utilization	2	2	3
Parts Life-Time	2	3	1

Table 3: Results of Analysis of the Simulation Packages with Respect to the Detailed Design Features

Packages	SIMAN/CINEMA IV	SIMFACTORY II.5	VS7
Detailed Model Building	3	1	2
Automatic Batch Run	3	3	1
High-Level Animation	3	1	2
Manufacturing Features	3	2	1
Total Output	2	3	2
Statistic of Queuing Lengths and Times for Cells and Transporters	3	3	2

Table 4: Distribution of Qualities of Matching with the Conceptual Modeling Features

package	SIMAN/CINEMA IV	SIMFACTORY II.5	VS7
Low (1)	2	1	1
Medium (2)	3	1	4
High (3)	1	4	3

Table 5: Distribution of Qualities of Matching with the Conceptual Modeling Features

package	SIMAN/CINEMA IV	SIMFACTORY II.5	VS7
Low (1)	0	2	2
Medium (2)	1	1	4
High (3)	5	2	0

6 MODELING DETAIL VARIABILITY

This section focuses on the process of modeling the different levels of detail of the MSD and how such models may be built in a manner that helps the already existing simulation packages to build a conceptual model and then flexibly extend it to the detailed design model. A simple method is introduced here which is based on some guidelines to ease the process of modeling systems with variable details. The following subsections describe the main steps for this method.

6.1 Modeling the Conceptual Level

1. *Identification and Classification of Model's Entities:*

Considering the conceptual level of the MSD, the first major step in building such a model should be the identification of the building blocks of the system, such as the different types of cells. When starting to identify the basic components of the conceptual model, the modeller must bear in mind that this model is to be extended into more detail later in a flexible manner without the need to create a new detailed simulation model from scratch. Therefore, the major components of a conceptual model might be classified as separate, preferably non-overlapping, blocks or entities regardless of their internal structures and details.

2. *Assigning Entities' Activities*:

After the identification of the main entities of the model, the second step is to assign the behavior of each entity. Generally, the modeller, when developing the conceptual model, must avoid including any unnecessary details that may overcomplicate the conceptual model. On the other hand, forgoing any other important components at this level will increase the problem of complication in the more detailed stage. If a simulation model of the conceptual level is built correctly, it will provide the required results and at the same time it will be a well-established base for detailed design. This can easily be extended with more details and complexity. At this stage the modeller may assign equal numbers for each entity or resource. For instance, he/she may assign the same number of machines for each cell. Another example might be the assigning of equal speed of transportation between any two cells. This is to eliminate the effect of such details on the simulation results. Generally speaking, the model at this stage is not necessarily 'valid' or typical of the real system.

6.2 Modeling the Detailed Level

3. *Entering Model's Details*:

At the detailed design level, the third step is to enter the new details within the boundaries of the blocks, which are already created at the conceptual level. That is, each block of the conceptual model is expanded separately from the rest of the model. Detailed data of a cell block could be the number of machines in the cell, the process duration of each machine, rate of failure for each machine, and maintenance time. Sometimes it can be expanded into an internal network of activities. For example, in a ‘painting-cell’ block parts may be queued for cleaning, then after cleaning they are transferred to another queue for painting. Some details might be entered as interactions between different entities such as, physical positioning, distances, and directions between cells within the system.

Table 6: Summary of the Steps of the Data Classification Method

Steps	Summary Procedures
Conceptual Level	
Step 1	Identification and classification of the main blocks or entities of the system separately, to be extended into more details later.
Step 2	Assigning averages and assumptions of real data to the established blocks and not entering much detail.
Detailed Level	
Step 3	Adding more extensive details (entities and activities) needed to build the final model including all necessary factors such as physical layout.
Step 4	Reassigning the model’s behavior by entering the real data into those blocks then fine-tuning the model to achieve the required results .

4. *Re-Assigning Entities’ Details*:

At this level, information assigned at the conceptual level is to be reassigned by introducing the real values to each entity before fine-tuning it to achieve the best results. The detailed design level can be considered as a network of blocks, each block containing all its corresponding details and other necessary details which represent interactions with other blocks. In addition, it gives the real physical layout. It is worth noting at this stage the model validity is very important, that it should represent all the details that make up the system as accurately as possible.

Generally, this method of classification will ease the process of model building and reduce the chance of error, as all necessary modifications are to be made from within the entity’s boundaries with no subsequent effects on other parts of the system. This reduces the time needed for any changes to the model, as a change of one entity will not affect the rest of the model. This method can be considered useful for effectively building flexible models with variable levels of detail.

The above discussion gives an overview of how system components can be classified in for flexible modeling of detail variability at the conceptual level, then how data is arranged and reassigned at the detailed level to be entered into the simulation package. Table 6 gives a summary of the steps, mentioned above, for data classification.

7 SUMMARY AND CONCLUSIONS

The research investigated three simulation packages, each one representing a simulation software category. The investigation focused on the ability of these packages to handle problems concerning MSD Framework, which involves different levels of detail.

The objectives of this research were: firstly, to investigate the existing manufacturing simulation software environments that may offer variable detail modeling, secondly, to classify models’ entities related to the levels of detail and to develop mechanisms in order to increase the level of detail of models effectively.

Findings from this research have suggested that there is no perfect simulation package that can fulfill the first objective. It can be seen from Tables 2, 3, 4, and 5 that most packages tend to meet some of the features of variable detail modeling, identified in section 5. Generally, there are simulation packages, particularly those devoted to manufacturing modeling, which are suitable for the conceptual level, while the others are most suitable for detailed design. For example, SIMAN/CINEMA IV is suitable for ‘detailed modeling,’ whilst SIMFACTORY II.5 is more suitable for ‘conceptual modeling.’

Some existing simulation packages can be used for modeling systems with different levels of detail. The problem of modeling variable detail mostly depends on the methodology of modeling itself (Toncich 1992). That is how the model’s entities are classified. The method introduced in section 6 identifies a suitable way to perform entity classification with regard to the levels of detail of the MSD Framework. This method of entity classification involves the classification of the model’s entities and the development of a mechanism to increase the level of detail effectively.

REFERENCES

CAMSD Research *CAMSD Newsletter*, 1: March/April 1994. London: Department of Manufacturing and Engineering Systems, Brunel University.

Carrie, A. 1988. *Simulation of Manufacturing Systems*. Chichester: John Wiley & Sons.

Cheng, T. C. E. 1985 Simulation of Flexible Manufacturing Systems. *Simulation*. vol. **45** (6): pp 299 - 302.

Fritz, S., Schmid, F., Wu, B. and Donnell, J. M. 1994. *Survey on Manufacturing Systems*: *Current Practice and Development in Computer Aided Manufacturing Systems Design*. London: Department of Manufacturing and Engineering Systems, Brunel University.

Johnson, M. E. and Poorte, J. P. 1988. A Hierarchical Approach to Computer Animation in Simulation Modeling. *Simulation*. vol. **50** (January): pp 30 -36.

Hlupic, V. 1997. A Comparative Evaluation of Ten Manufacturing Simulation Packages. Forthcoming in *Journal of Computing and Information Technology*.

Hlupic, V. And Paul, R. J. 1993. Simulation Software in Manufacturing Environments: A Users' Survey. *Journal of Computing and Information Technology*. vol. **1** (3): pp 205 - 212.

Hlupic, V. And Paul, R. J. 1995. A Critical Evaluation of Four Manufacturing Simulators. *International Journal of Production Research*. vol. **33** (10): pp 2757 - 2766.

Hlupic, V. and Paul, R. J. 1996. A Methodological Approach to Manufacturing Simulation Software Selection. *Computer Integrated Manufacturing Systems*. vol. **9** (1): pp 49 - 55.

Hollocks, B. 1992. A Well-Kept Secret?: Simulation in Manufacturing Industry Review. *OR Insight*. vol. **5** (October - December): pp 12 - 17.

Kochhar, A. K. 1989 Computer Simulation of Manufacturing Systems - 3 Decades of Progress. In the *Proceedings of the 3rd European Simulation Congress*. Editors: D. Murray-Smith, J. Stephenson, R. N. Zobel, San Diego, the Society for Computer Simulation. pp 3 - 9.

Law, A. M. and Haider, S. W. 1989. Simulation in Manufacturing - Selecting Simulation software for Manufacturing Applications: Practical Guidelines & Software Survey. *Industrial Engineering*, vol. **31** (May): pp 33 - 56.

Law, A. M. and Kelton, W. D. 1991. *Simulation Modeling and Analysis*, 2nd edition. Singapore: McGraw-Hill, Inc. International Editions.

Paul, R. J. And Balmer, D. 1993. *Simulation Modelling*. Bromley: Chartwell Bratt.

Pegden, C. D., Shannon, R. E., and Sadowski, R. P. 1990. *Introduction To Simulation Using SIMAN*. New Jersey: Systems Modeling Corporation and McGraw Hill, Inc.

Pidd, M. 1992. *Computer Simulation in Management Science*. 3rd edition. Chichester: John Wiley & Sons.

Schroeder, R. G. 1993. *Operations Management: Decision Making in the Operation Function*. Singapore: McGraw-Hill, Inc. International Edition.

Tanner, J. P. 1985. *Manufacturing Engineering: an Introduction to Basic Functions*. New York: Marcel Dekker, Inc.

Toncich, O. J. 1992. Multilevel Simulation for Advanced Manufacturing Systems. *Advanced Manufacturing Technology*. vol. **7**: pp 178 - 185.

Ülgen, Onur, and Sanjay S. Upendram. 1997. Productivity Simulation in Automotive Industry. *In Simulators International XIV*, Editors: Maurice Ades and Roy Fray. pp 199 - 204.

Vanderspek, P. G. 1993. *Planning For Factory Automation*. London: McGraw-Hill, Inc.

Wu, B. 1992a. *Manufacturing Systems Design and Analysis*. London: Chapman & Hall.

Wu, B. 1992b. *Manufacturing Systems: Advanced Manufacturing Systems - M.Sc. Program*. London: Department of Manufacturing and Engineering Systems, Brunel University.

AUTHOR BIOGRAPHIES

TILLAL ELDABI is a Research Assistant, and Secretary to the Centre for Applied Simulation Modelling, at Brunel University, UK. He received a B.Sc. in Econometrics and Social Statistics from the University of Khartoum, and an M.Sc. in Simulation Modeling from Brunel University. His doctoral research is in aspects of simulation output analysis, concentrating on multiple output analysis and the detection of output interrelationships. Alongside his Ph.D. research, he is working on applying simulation modeling to health services and health economics, looking at the possibilities of identifying standard methodologies for conducting the simulation modeling process in the health area.

RAY J. PAUL is a Professor of Simulation Modeling, Director of the Centre of Applied Simulation Modelling, and Head of the Department of Information Systems and Computing, all at Brunel University, UK. He received a B.Sc. in Mathematics, and an M.Sc. and a Ph.D. in Operational Research from Hull University. He has published widely, in books, journals and conference papers (over 100 publications), many in the area of the simulation modeling process and software development. He has acted as a consultant for a variety of United Kingdom government departments, software companies, and commercial companies in the tobacco and oil industries. He is a co-editor of the Springer-Verlag Applied computing book series. His research interests are in methods of automating the process of discrete event simulation modeling, and the general applicability of such methods and their extensions to the wider arena of information systems. He is currently working on aspects of simulation in the social sciences, in particular in health management.