Int. J. Engng Ed. Vol. 10, No. 4, pp. 361–372, 1994
Printed in Great Britain.
0742–0269/91 $3.00+0.00

A Hypertext Approach to Discrete Event Simulation: The Development of a Computer-based Learning Tool*

T. A. SPEDDING
R. DE SOUZA
Design Research Center, School of Mechanical and Production Engineering, Nanyang Technological University, Singapore 2263

This paper presents an interactive simulation system in a hypertext environment in an attempt to develop an awareness of the potential of simulation techniques both in an educational environment and for the practising engineer. The hypertext medium used here is Toolbook which runs on an IBM PC compatible computer under the Windows environment. The system provides a hierarchical combination of text, graphics and animation to illustrate simulation analysis. The software develops a unified and structured approach to simulation so that the user can work through the information at several levels, depending on their expertise or particular requirements. At the centre of the system is a fully developed simulation of a surface mount technology (SMT) line, which is accessible to the user from any point in the system. The system is therefore designed to respond to different levels of ability and experience, thus alleviating one of the classic problems of engineering education which is difficult to address using conventional media. This paper illustrates the potential of hypertext as an educational tool in engineering. The conceptual design of the system can be adopted in any discipline which requires a high level of visual and cognitive interaction to gain a thorough understanding of engineering principles and their applications.

SUMMARY OF THE EDUCATIONAL ASPECTS OF THE PAPER

The paper describes software applications useful in the following engineering disciplines:

Manufacturing systems, discrete event simulation

The paper is suitable for teaching/classwork/self-study for engineering students at the following level:

Undergraduate

What aspects of your contribution are new?

Using Hypertext to present theory and applications of simulation in manufacturing

How is the material as presented to be incorporated in engineering teaching?

Useful for lecturing aid and self-study programme

Have the concepts presented been tested in the classroom or in project work?

Yes

What conclusions have been drawn from the experience?

Helpful in allowing students to visualize concept of simulation

Other comments on the benefits of your approach for engineering education:

Computer-aided learning is useful in both classroom and self-study by providing another natural medium for conceptualizing data.

INTRODUCTION

HYPERTEXT and multimedia systems offer great potential in engineering education through their facility to present concepts and applications in a highly interactive and visual environment. Traditional education techniques are often limited in providing effective communication of engineering ideas to both undergraduate and practising engineers.

In its simplest form, hypertext can be thought of as a book (the screen of the computer being analogous to the page of a book); the user can read text and view diagrams in the usual manner. The user can also read through the pages serially or jump to different topics and find more information about specific items as required. Consequently whilst developing the system much of the emphasis was placed on cultivating a self-paced learning approach. Hypertext systems also have the facility to animate diagrams so that two-dimensional or

* Paper accepted 25 May 1994.

even three-dimensional representation can be 'brought to life'. This feature in particular is excellent in putting over engineering concepts, working features, movements or work flow which are difficult to describe in words or through static diagrams. More complex systems provide interfaces to proprietary animation systems and provide full multimedia facilities such as video and sound.

Simulation is an ideal subject for computer-based learning. Primarily, an animated simulation of a manufacturing system, provides the user with an insight into how manufacturing works as an entity. The user obtains a better appreciation of system logistics such as flow of material, queuing, bottlenecks, etc. and is able to explore these concepts by changing the parameters of the model to perform simple 'what-if' type analysis. Not only does the user form a better appreciation of manufacturing, they also learn about simulation—as a modeller and its potential as a problem solving technique. The system proceeds to explain how a simulation is written in terms of the basic concepts and how to analyse the results and the statistical output of a simulation.

Simulation represents a powerful technique which may be used to model and analyse a manufacturing system. However, one of the fundamental problems of simulation is the initial difficulty in understanding it. This has possibly limited its application in industry. Current simulation systems offer a greater power and flexibility but initially require a fundamental understanding from the end user. The development of a hypertext system for the teaching of simulation could be used as a self-paced learning tool in both the educational and industrial environment. The system could also be used as an on-line teaching aid for specific examples in the lecture theatre or tutorial room. The system provides an introduction to simulation concepts and techniques, showing how simulation models are constructed and illustrates how they can be used for analysis and problem solving. This approach not only provides the user with an introduction to simulation but also gives an excellent overview of the 'mechanics' of a manufacturing system.

The centre of the system is a small but complete generic simulation model consisting of five processes, written completely within the hypertext system. The actual example is based on a surface mount technology (SMT) line but can be applied to any manufacturing system consisting of five (or less) processes. The user can either run the system using default values or interactively, through a system of input controls. The rest of the system takes the form of a standard hypertext book consisting of screens showing text, diagrams and animation demonstrating the concepts of simulation, etc. The user can work through the system in a straightforward manner, finally ending with the complete system with which they can apply their new-found knowledge. However, if the user prefers to try out particular aspects of the model as they learn new principles, this can be achieved by moving to and from the simulation model from anywhere in the system. This facility also provides for users of differing abilities who may wish to experiment with a particular aspect of simulation or manufacturing logistics. In this way the system provides an open-learning experience, offering self-paced learning, a flexibility in style or approach, depending on the needs of the individual and the facility to concentrate on specific aspects of manufacturing or simulation.

The hypertext medium used here is Toolbook which runs on an IBM PC compatible computer under the Windows environment. The main reasons for choosing this particular software was that it is inexpensive, a run-time version can be freely distributed with Windows 3.0/3.1 and it has a powerful and structured programming language capable of the task.

SIMULATION AND SURFACE MOUNT TECHNOLOGY

Simulation models are used to develop conclusions that provide insight into real-world systems. The two most common types of computer simulation are continuous and discrete event simulation. Continuous simulation is the modelling of a set of equations that represent a system over time. Discrete event simulation, on the other hand, is characterized by the passage of blocks during which nothing happens and punctuated by events that change the state of the system. The occurrence of the discrete events may be either deterministic or stochastic in nature. Manufacturing systems are generally modelled as discrete systems.

The basis of the system is the simulation of an SMT line. The model shows the assembly of a single-sided printed circuit board (PCB) using a five-stage process. SMT is the technique used to mount electronic components on the surface of PCBs or substrates. This technique is gaining popularity and has, to a large extent, replaced the conventional technology of inserting components through holes in the board. However, SMT cannot totally replace conventional technology because up to now, not all components are suitable. Hence, it is common to find mixed technology in PCBA lines. Management of this PCBA process can be complex, particularly in the subcontractor sector of the industry where there is uncertainty about incoming orders, quantities and mix. Even where certainty in production does exist, it is often necessary to compare alternatives for future equipment and layouts and to compare work flow alternatives such as changing batch sizes.

Design, planning and experimentation through simulation models is ideal as one can plan or anticipate production requirements without undue material handling of the PCBs and downtime of expensive equipment.

Discrete event simulation is rapidly becoming a popular tool to address these manufacturing issues

([1] and others) in scheduling, control and management [2], analysis of new process technologies to support the formulation of assembly, test and repair strategies [3], inventory analysis, line balancing and cost justification [4] and to study grouping of equipment into autonomous manufacturing cells dedicated to a specific type of product or to a family of products [5].

Simulation software provides the facility to build a simulation model of a manufacturing system. Typically such software provides an interactive environment and a natural framework in which to develop and execute simulation models. Systems usually provide a variety of data processing and modelling tools and include graphics and animation.

Once the generic simulation model has been developed however, a high level of expertise is required to ensure that the model characterizes a realistic environment. Further, it is important to ensure that the results of simulation are correctly interpreted so that the most appropriate remedial action may be executed.

Hypertext offers an alternative interpretation to simulation modelling and analysis which effectively solves many of the problems of interpreting the output of simulation models. The hypertext approach provides an interlinked combination of text, graphics and animation using non-sequential computer screens or scrolling windows.

SYSTEM DETAILS

The SMT line

A standard process flow for all types of surface mount assemblies does not exist. Industry has evolved three main types of surface mount design. These are commonly known as type I, type II and type III design [6–8]. The definition for the different types of board design is by no means universal. The type II design is actually a combination of types I and III.

The simulation model is based on the production of the type I design. For simplicity only single-sided boards are considered. First the PCB screen is printed with solder paste, components are placed and the assembly is baked in a convection or infrared oven to drive off volatiles from the solder paste. The assembly is then soldered by the reflow solder process and cleaned with solvent. Finally, the assembly is visually inspected before loading into the outgoing pallet.

The above sequence is animated on a single computer screen and although the animated sequence is specific to the SMT line, a generic model is easily adapted which can be used for any manufacturing or assembly process which consists of five machines or less.

Toolbook

Toolbook [9] is a mouse-driven, hypertext system for the windows environment of an IBM PC computer. Toolbook uses the book metaphor to define an application. Essentially it allows the programmer to develop a system of text, graphics and animation or a combination of all three on each page. By defining icons, buttons can be specified to move to another screen (or page). The 'book' does not have to be read in sequence. The basis of this software construction set is Openscript, toolbook's scripting language which is an object-oriented, block structured language.

Buttons fields and graphics can be developed on the screen using tools provided within the system. Alternatively graphics and text can be imported into the system via the windows clipboard facility, allowing the inclusion of graphics created in other systems.

Toolbook and animation

Animation plays an important role in simulation because of its ability to communicate the essence of a simulation model. In a computer animation, the user can see the key elements of a system (e.g. machines and parts, etc.), represented by icons, change or move over time. In Toolbook, animation is produced concurrently with program execution, in one of three different ways.

1. Once an object has been defined the *move* command in the Openscript language can be used to move that object to any coordinate position on the screen.
2. Toolbook's script recorder allows the programmer to create scripts without using the scripting language by recreating the actions on the screen. Animation, for example, can be produced by recording the movement of a graphics object as it is dragged across the screen by the mouse. This is automated method of the animation technique described in 1 above, and can produce a similar Openscript code which can then be edited if necessary.
3. The classic approach to animation may also be developed by creating a sequence of screens with an object placed at slightly different positions and then moving quickly (flipping) through the screens.

The choice of animation technique depends on the particular application and requirements.

THE SIMULATION MODEL

This section describes the simulation model which is the focus of the hypertext system. As previously mentioned the user has the flexibility to access the simulation model from anywhere in the system. The illustrations in the following sections are actual screens from the learning module and serve as an example of the type of information provided by the system. It should be noted however that many are static diagrams from animated sequences.

Event scheduling

A discrete event simulation is stochastic in nature, using random samples from a probability distribution to drive the model through time. There are three different approaches which can be adopted [10] to drive the model through time. These are

(1) next event scheduling,
(2) activity scanning,
(3) process orientation.

These approaches are based on the concept of generating random numbers for each process, collecting the statistics and looking for the occurrence of a key event. The hypertext system uses next event scheduling (Fig. 1), because of limitations in Openscript. This approach is typically adopted for simulation coded in general purpose programming languages because these languages do not offer ready-made constructs which are found in simulation languages.

Next event scheduling advances along a time-scale until an event is encountered and depending on the type of event, appropriate conditional or unconditional action is taken. The technique begins by selecting the first event on the time-scale and performs all actions including placing any newly generated events in their proper order on the time-scale (i.e. updating the time-scale events). The simulation will then move forward to the next event on the time-scale and all actions relating to this event will be performed. This next event time advanced mechanism is repeated until a desired simulation period is covered or the required number of boards are assembled.

The event scheduling routine uses a combination of variables and arrays created in the openscript language including the following.

1. *System state*: a collection of state variables used to describe the system at a particular time.
2. *Simulation clock*: variables giving the current value of simulated time.
3. *Event list*: a list containing the next time when each type of event will occur.
4. *Statistical counters*: a subroutine to check for any simulation violations, prompting the user accordingly.
5. *Initialization routine*: a subprogram to initialize the simulation model according to the method chosen.
6. *Time advance routine*: a subprogram that determines the next event from the event list and then advances the simulation clock at the time when the event occurs.
7. *Event routine*: a subprogram that updates the system state when a particular type of event occurs.

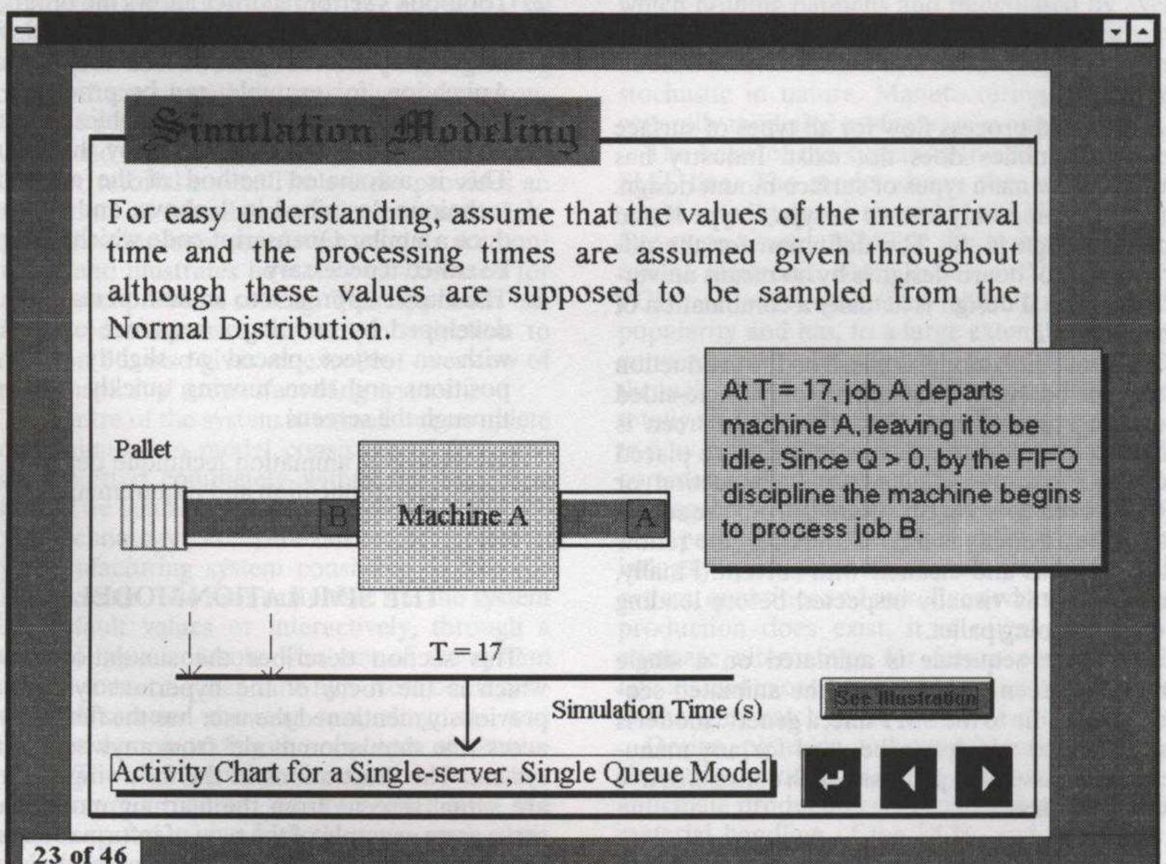

Fig. 1. Next event scheduling.

8. *Random number generator*: a subprogram to generate random numbers from the normal distribution.
9. *Report routine*: a subprogram that computes estimates (from the statistical counters) of the desired measures of performance. At the end of the simulation, individual summary reports for various workstations and an overall system report is generated.

Random number generation

The generation of random numbers is the essence of any stochastic simulation. The system generates arrival times and processing times using random numbers (Fig. 2). On start-up the model uses default values for the mean and standard deviations for the various processes, however, the user may specify whatever values they require. Note that by specifying zero standard deviation a deterministic process may be assigned to any process. Random numbers are generated using 'random()' an intrinsic function within Openscript which generates uniformly distributed random numbers. These are then transformed to the Gaussian distribution using the Box Muller transformation [11]. The Gaussian distribution was chosen because it represents the most typical distribution encountered in real-life systems. However, a future enhancement of the system, will provide the user with a choice of distributions (exponential, Weibull, gamma, etc.) so that it is possible to investigate the effects of their differing characteristics.

Queuing

One of the advantages of simulation models is their ability to model the behaviour of queues which build up within the manufacturing system (Fig. 3). This has many benefits including the identification of bottlenecks and the optimal design of buffers, etc. A queuing system can be described by its calling population, the system capacity, the nature of the arrivals and services and the queuing discipline [12].

The calling population for the modelled system is assumed to be infinite with a limited capacity. As such, it is assumed that the buffer area in front of each workstation can only store ten PCBs. Whenever the buffer size exceeds this figure the simulation will terminate and the user will be prompted by the occurrence of a bottleneck in the line. All inter-arrival times are assumed to be stochastic and follow the Gaussian distribution. The arrival occurs one at a time and serviced by a single server at all workstations except the inspection stage. In this last workstation, it is assumed that there are multiple servers manning the station. Finally, the first-in, first-out (FIFO) queue discipline is adopted.

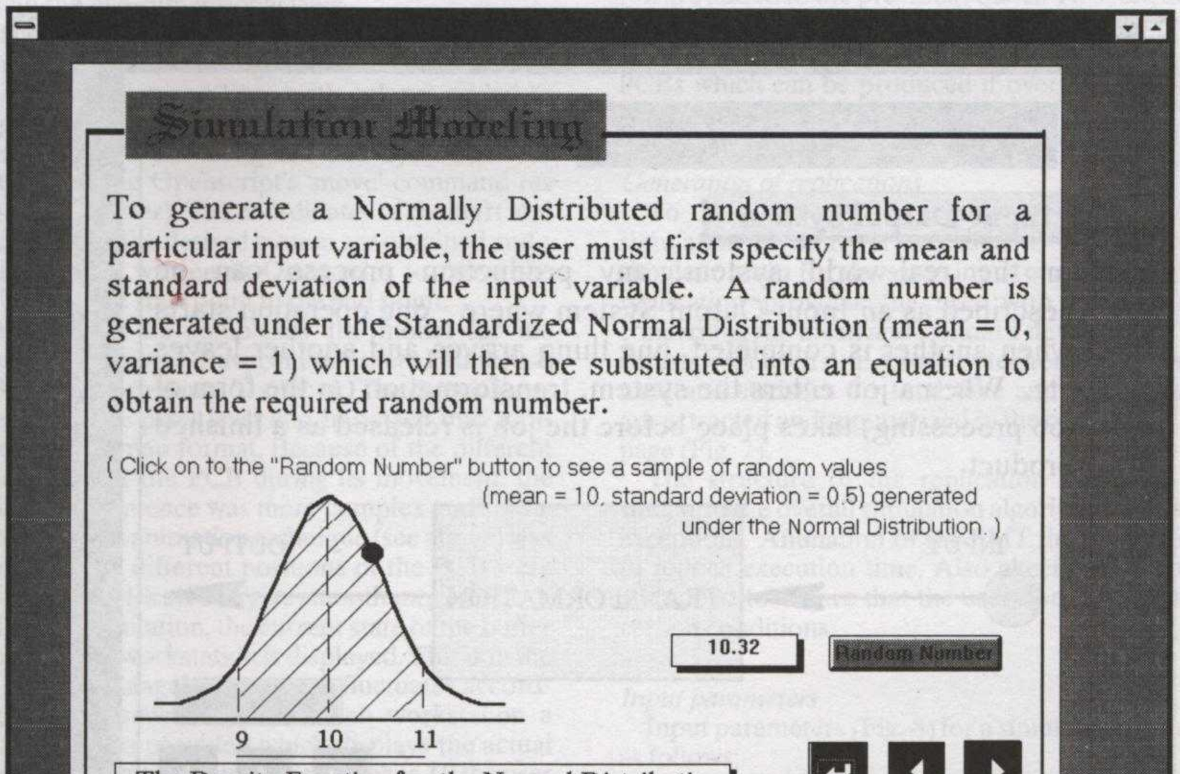

Fig. 2. Generation of random numbers.

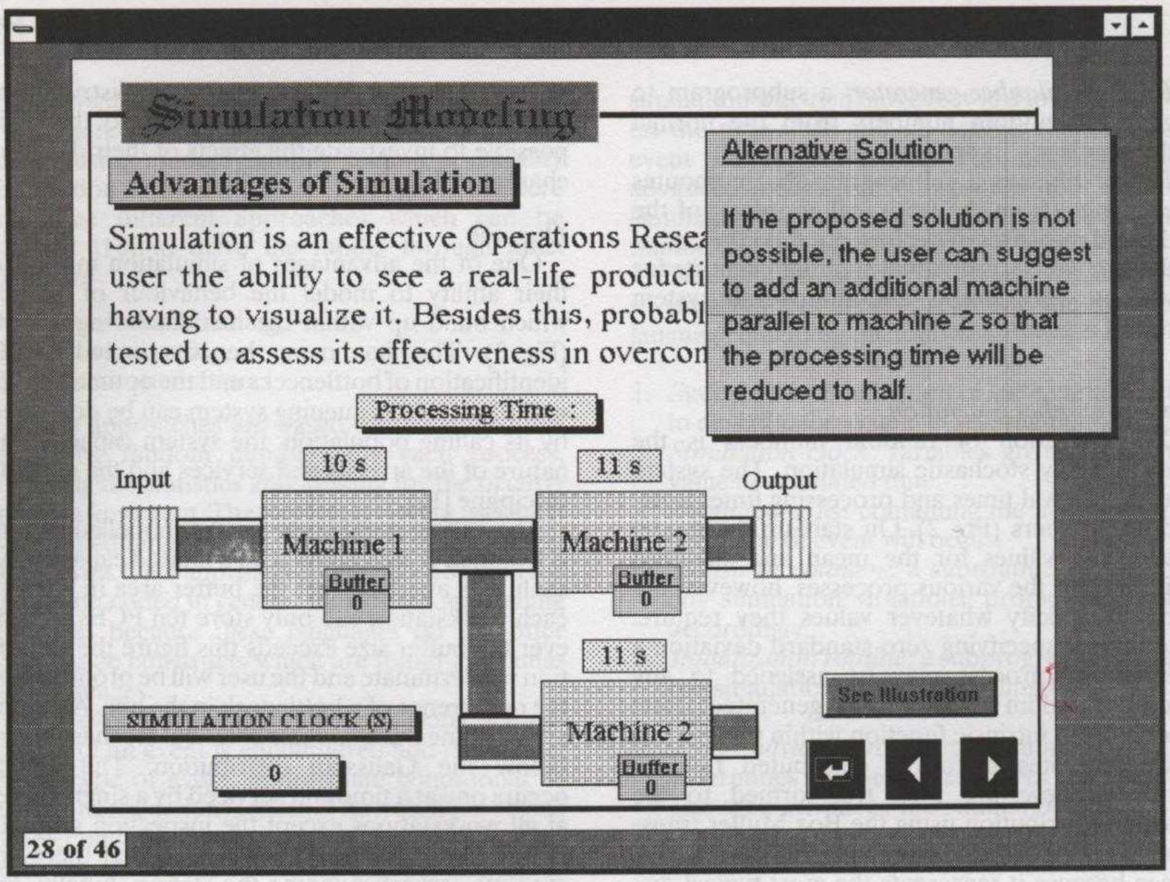

Fig. 3. Using simulation to identify bottlenecks.

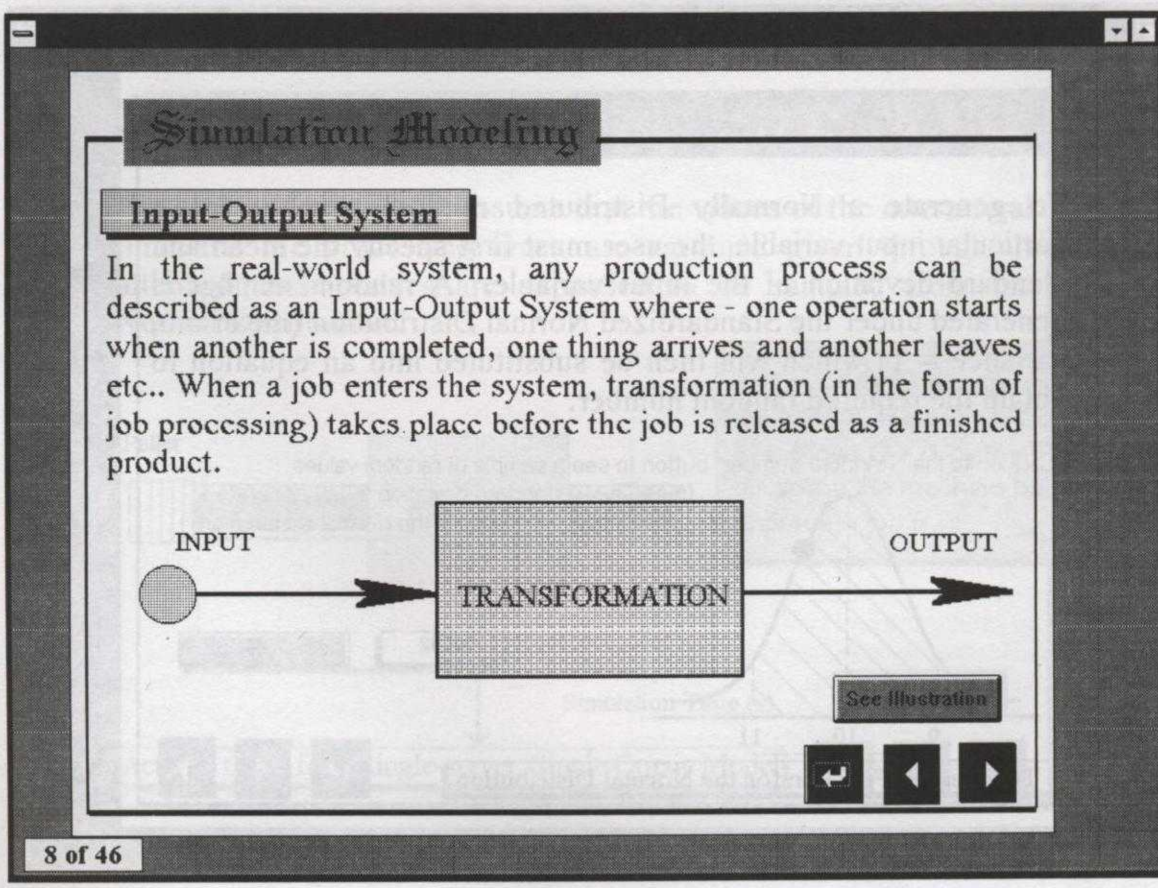

Fig. 4. Representing a simulation model as a transformation system.

The complete model

In its simplest form a manufacturing system can be represented as an input–output process where raw material is introduced into it, subjected to a transformation and finally exits as a finished product (Fig. 4). Each process within the system can also be considered as smaller input–output systems. As a result, a manufacturing system is made up of an organized network of input–output systems where individually assigned tasks are performed. Parts arriving at a server will be processed if the machine is available. Otherwise it will wait in a queue until the machine is free.

In terms of modelling, the SMT line can be considered as a dynamic, stochastic discrete model. Dynamic because time is the critical factor in the performance of the system, stochastic because the arrival times and processing time of the machines exhibit variation and discrete as the system state only changes at discrete changes in time.

The SMT line represented in the system consists of five machines. This is represented in graphical form on the screen. Either a two-dimensional (Fig. 5) or three-dimensional (Fig. 6) representation is available. However, the two representations are independent and thus there is no facility to move from one to the other during a simulation run. It was considered that whilst the three-dimensional representation was more realistic, it was easier to follow the progression of the PCB within the two-dimensional view.

The two-dimensional model was drawn entirely using Toolbook's drawing tools. The model is made up of rectangles and lines, with colours added to improve the appearance. Animation of the movement of the PCB along the conveyor belt was obtained using Openscript's 'move' command (as described earlier). The coordinates of the start and end points of the desired motion are obtained and a finite loop is written in Openscript to move the PCB from the start point to the end point.

Because of its complexity the model for the isometric view was drawn in Publishers Paintbrush [13] and imported into Toolbook. The drawing was created at the pixel level and imported into toolbook in a bitmap format. Because of the different orientation of the PCB during its movement, the animation sequence was more complex and so the flipping pages animation technique (see above) was adopted. The different positions of the PCB were painstakingly located to give smooth movement.

During simulation, the current state of the buffer area for each workstation is displayed. This is in the form of a rectangular bar which fluctuates according to the queue length. At each workstation a counter is also provided which displays the actual queue size. Once the queue size reaches 10 the user will be prompted that a bottleneck has occurred as previously described.

FEATURES OF THE SIMULATION MODEL

Simulation speed

The user is provided with buttons to slow down or speed up the simulation run. This is useful, for example, when certain details of the simulation need to be analysed visually. Where this is not required the simulation can be run at full speed. For the two-dimensional animation the two speeds were obtained by defining two loops for the same PCB movement, the difference in the loops being the incremental step. For the isometric view the animation speed was defined by the difference in distance between the PCBs on consecutive pages.

Simulation method

The model allows the user to choose simulation either by batch or production time. To determine the total time to process a fixed number of PCBs, the user is required to enter the batch size and the simulation will determine when the batch has been completely processed.

Alternatively to determine the throughput over a particular period the user is required to specify the period.

Continue simulation

The continue simulation function allows a simulation run to be restarted with all the previous results maintained. Results for the new simulation run are added to the previous results. This function is particularly useful where, for example, it is necessary to estimate the additional number of PCBs which can be produced if overtime is to be scheduled.

Generation of replications

To investigate the integrity of a particular simulation a facility is provided to replicate the simulation run for the same input conditions. The replication provides the user with information concerning the mean and standard deviation of the statistics for each run. A total of ten replications are used and statistics for total time and total output are extracted and summarized in the overall results page (Fig. 7).

The structure of the replication algorithm is similar to the overall simulation algorithm with two exceptions. Animation of the SMT line is omitted to reduce execution time. Also checking routines are added to ensure that the user does not violate certain conditions.

Input parameters

Input parameters (Fig. 8) for a simulation run are as follows:

(1) mean and standard deviation for the processing time of the five workstations and the interarrival time for each PCB pallet,
(2) the simulation method, i.e. either batch by batch or production time,

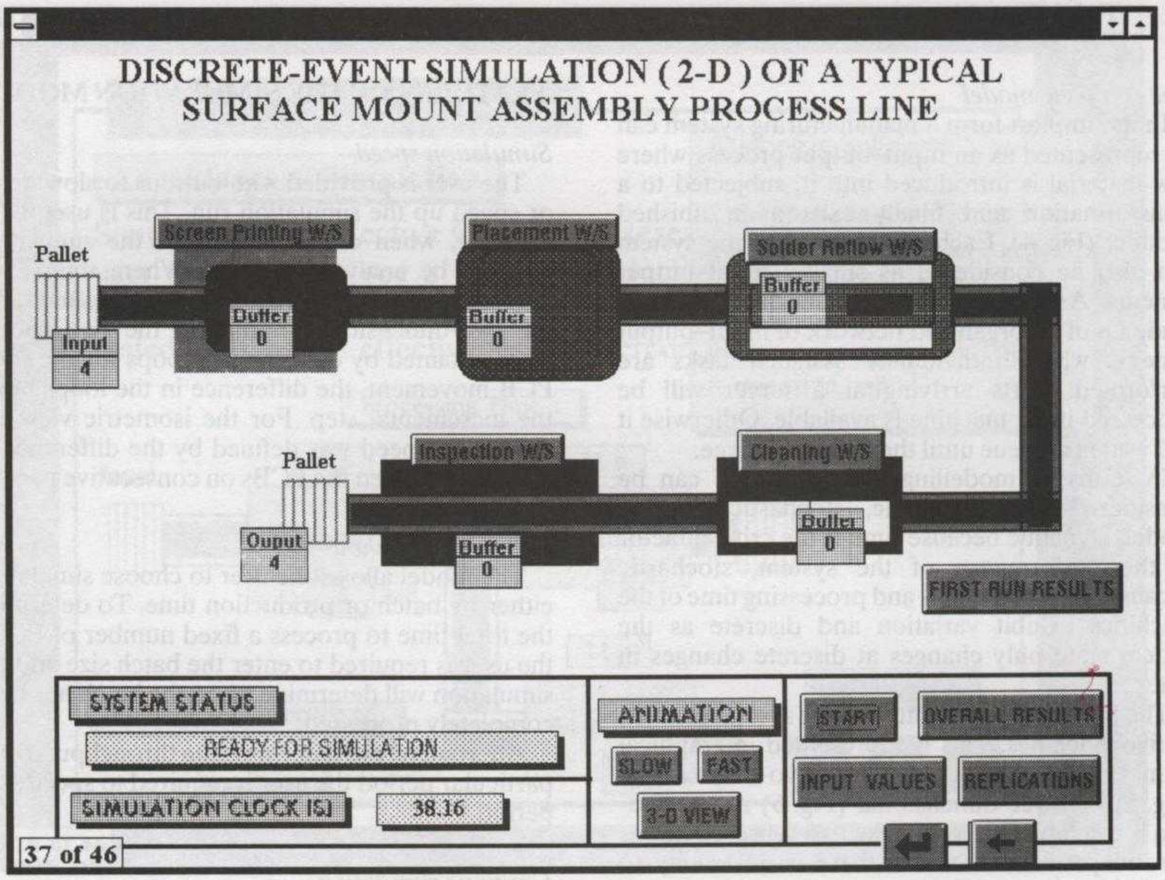

Fig. 5. Two-dimensional simulation model of SMT line.

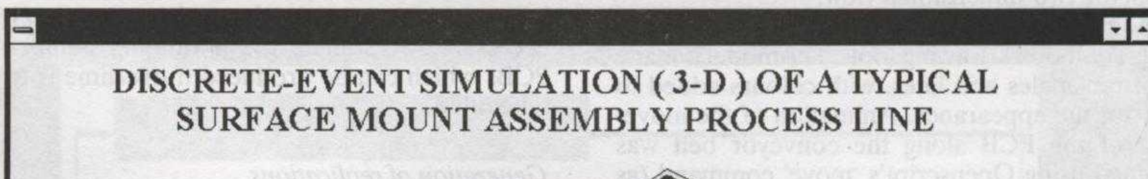

Fig. 6. Three-dimensional simulation model of SMT line.

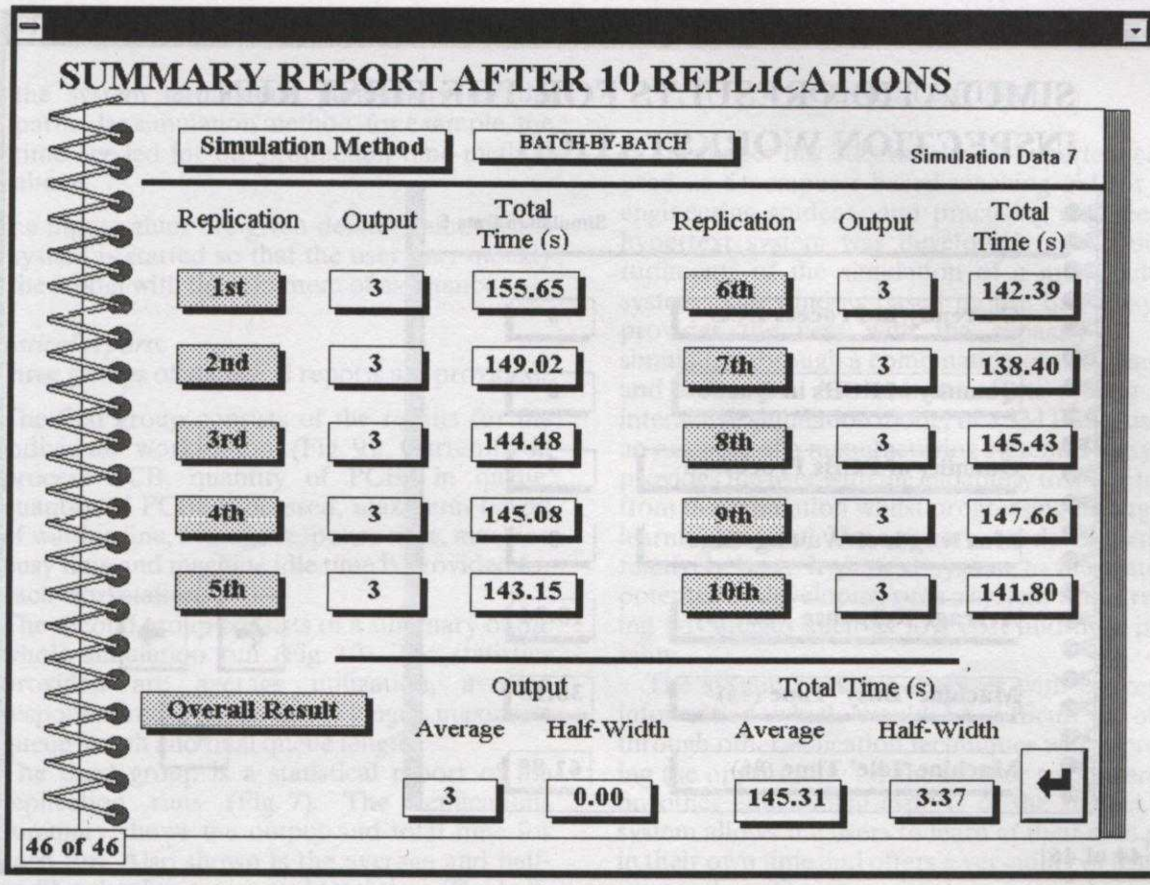

Fig. 7. Summary report of simulation after ten replications.

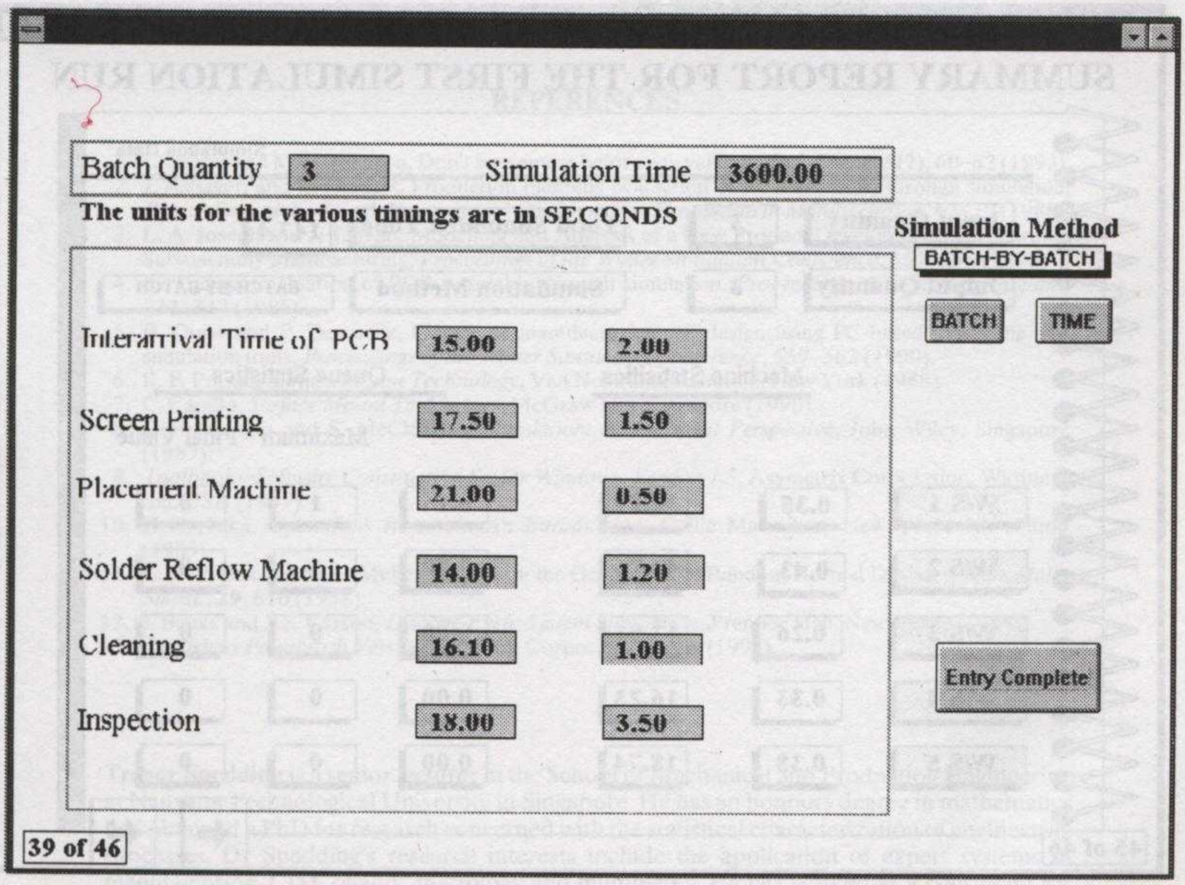

Fig. 8. Input parameters for simulation model.

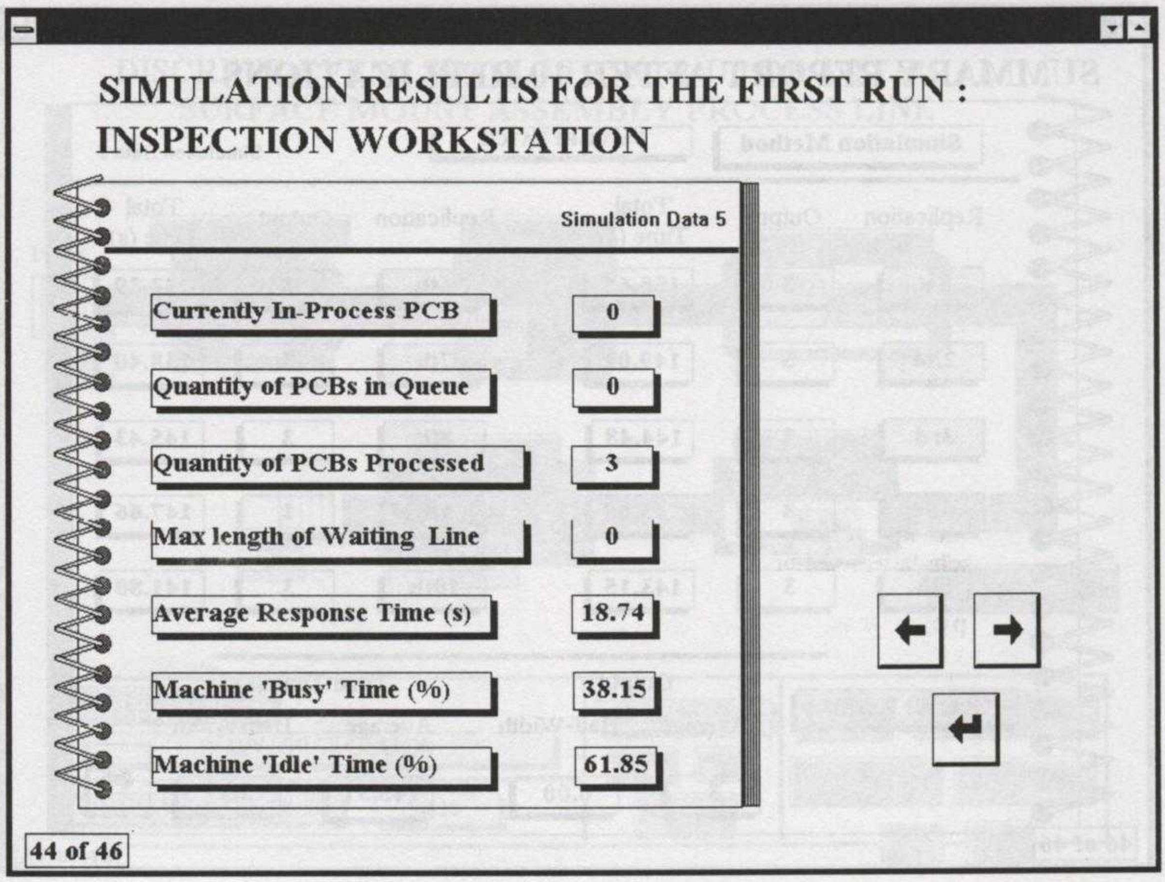

Fig. 9. Simulation report for the inspection workstation.

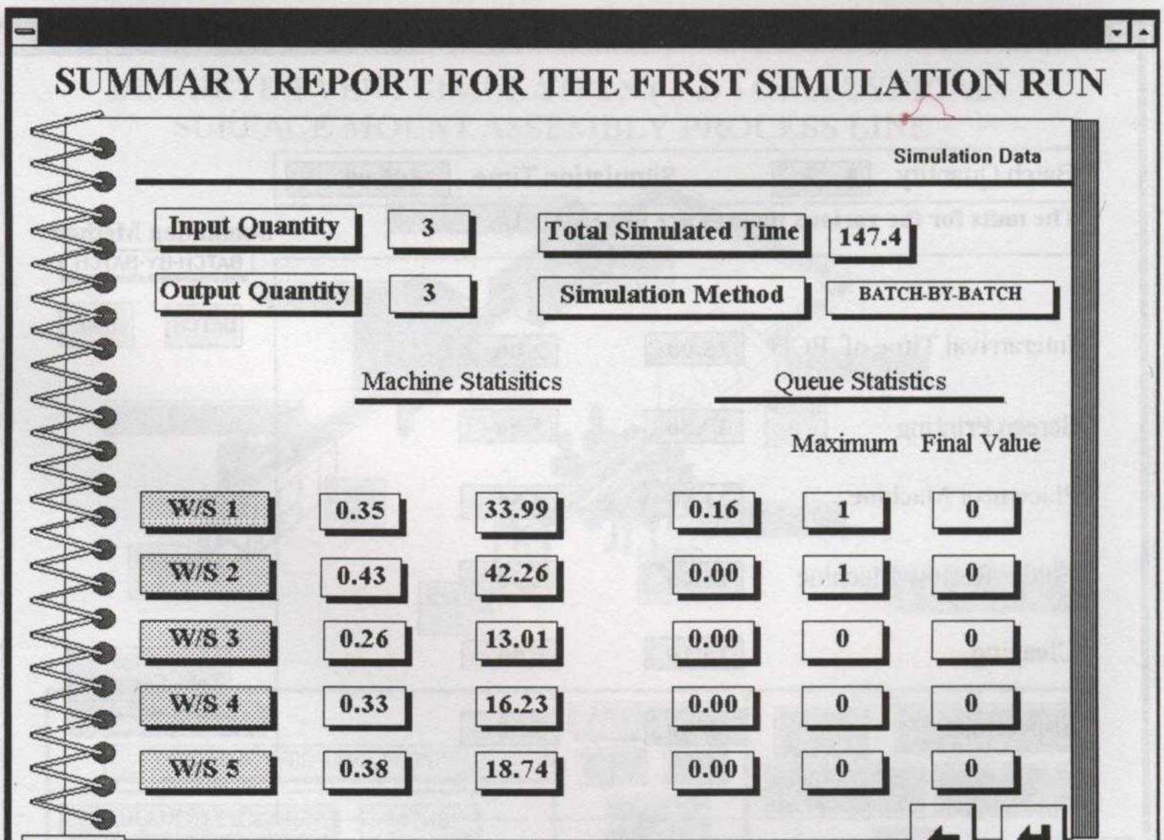

Fig. 10. Summary report for the first simulation run.

(3) the system termination parameters for the particular simulation method, for example, the time needed for the production time method above.

The input values are given default values when the system is started so that the user may quickly use the model with the minimum of assistance.

Statistical reports

Three groups of statistical reports are provided.

1. The first group consists of the results for the individual workstation (Fig. 9). Currently in process PCB, quantity of PCBs in queue, quantity of PCBs processed, maximum length of waiting line, average response time, machine busy time and machine idle time is provided for each workstation.
2. The second group consists of a summary of the whole simulation run (Fig. 10). The statistics provided are average utilization, average response time, average queue length, maximum queue length and final queue length.
3. The third group is a statistical report of the replication runs (Fig. 7). The replications summary shows the output and total time for each run. Also shown is the average and half-width value for output and total time. The half-width value, based on 95% confidence shows a measure of the precision of the point estimate of the average.

CONCLUSIONS

This paper has illustrated how hypertext can be used as a computer-based teaching aid for both engineering students and practising engineers. A hypertext system was developed to explain the rudiments of the simulation of a manufacturing system. The window-based mouse driven system provides the user with the capacity to learn simulation through a combination of text, graphics and animation. At the heart of the system is an interactive simulation model of a SMT line, used as an example of a manufacturing system. The system provides the user with the capability to move to and from the simulation whilst progressing through the learning material. The software was developed on a relatively basic hypertext system to illustrate the potential of developing such a system whilst retaining advantages such as low cost and high portability.

The system provides the user with conceptual information which would be difficult to obtain through other education techniques whilst providing the opportunity for the teacher to concentrate on other educational aspects of the subject. The system allows the users to learn at their own pace, in their own time and offers a versatility of learning approaches. The conceptual design of the system can be adopted in any discipline which requires a high level of visual and cognitive interaction to gain a thorough understanding of engineering principles and their applications.

REFERENCES

1. L. Baxter and M. E. Johnson, Don't implement before you validate, *Ind. Engng*, **5(2)**, 60–62 (1993).
2. J. Ashayeri and L. Gelders, Production planning evaluation of PCB assembly through simulation, *Proceedings of the Fourth International Conference on Simulation in Manufacturing*, 63–70 (1988).
3. L. A. Joseph and J. T. Watt, Modelling and Analysis of a New Product Development in Electronic Subassembly Manufacturing, *Proceedings of the Winter Simulation Conference*, 559–563 (1990).
4. K. Tumay, Evaluation of PCB Assembly through simulation, *Proceedings of Technical Program*, 337–342 (1985).
5. R. Guiar and P. Desruelle, Electronic manufacturing cell design using PC-based modelling and simulation tools, *Proceedings of the Winter Simulation Conference*, 559–563 (1990).
6. R. P. Prasad, *Surface Mount Technology*, Van Nostrand, Reinhold, New York (1989).
7. C. Capillo, *Surface Mount Technology*, McGraw Hill, Singapore (1990).
8. C. H. Mangin and S. McClelland, *Simulation: A Statistical Perspective*, John Wiley, Singapore (1987).
9. *Toolbook—Software Construction Set for Windows, Version 1.5*, Asymetrix Corporation, Washington, USA (1989).
10. H. A. Taha, *Operations Research: An Introduction*, Collie Macmillan, New York, 4th edition (1989).
11. G. E. P. Box and M. E. Muller, A Note on the Generation of Random Normal Deviates, *Ann. Math. Statist.*, **29**, 610 (1958).
12. J. Banks and J. S. Carson, *Discrete Event System Simulation*, Prentice Hall, New Jersey (1984).
13. *Publishers Paintbrush Version 2.0*, Zsoft Corporation, USA (1991).

Trevor Spedding is a senior lecturer in the School of Mechanical and Production Engineering at Nanyang Technological University in Singapore. He has an honours degree in mathematics and obtained a PhD for research concerned with the statistical characterization of engineering processes. Dr Spedding's research interests include the application of expert systems in manufacturing, CIM, quality, metrology and multimedia. He has worked as a consultant for several companies in the UK and Singapore.

Dr Robert de Souza is presently a lecturer in the School of Mechanical and Production Engineering at Nanyang Technological University (NTU) in Singapore. Robert also leads the Smart Manufacturing Group in the Schools Design Research Centre. Prior to joining NTU in early 1991, Robert was a senior research engineer in the Department of Manufacturing Engineering at Loughborough University of Technology (UK). His current research interests are in process planning for electronics assembly and discrete event simulation modelling and analysis. Robert has done case study and consulting work with prominent industrial organizations in Singapore and in the UK. He has coauthored a book and over 50 journal, conference and technical papers. He holds a PhD, MSc and BSc and is a chartered engineer and corporate member of IIES, ASME, IEE and SMTA.