Int J Adv Manuf Technol (2005) 25: 1254–1265
DOI 10.1007/s00170-003-1957-7

ORIGINAL ARTICLE

Q. Wang · C.R. Chatwin

Key issues and developments in modelling and simulation-based methodologies for manufacturing systems analysis, design and performance evaluation

Received: 23 June 2003 / Accepted: 19 September 2003 / Published online: 23 June 2004

Abstract Discrete event simulation (DES) has been widely applied to modelling and simulation of computer and engineering systems and is an active field of research that has now evolved from 2D to 3D discrete event simulation. This paper attempts to address several key issues in a successful implementation of DES models based on our own and the previous experiences of others. It describes the common basis, which forms the core for the application of modelling and simulation methodologies that are available to support manufacturing systems analysis, design and performance evaluation. Through a comprehensive literature survey, this paper summarises and compares the most widely used optimisation techniques for simulation of manufacturing systems; an overview of the recent and popular simulation languages and packages available for the modelling and simulation community and the classification of their utility for modelling and simulation of manufacturing systems is also given. Finally, this paper summarises and reports the latest development in the most exciting world wide web (www)-based simulation techniques that represent a future that may completely change the nature and future exploitation of modelling and simulation technology in industry.

Keywords Internet · Manufacturing systems · Modelling · Optimisation · Simulation

1 Introduction

Among specialists, it is widely accepted that mathematical or analytical modelling techniques are not sufficient if a detailed analysis is required of complex manufacturing systems [1–9], the major weaknesses in using mathematical or analytical methodologies are:

1. When analysing a complex system, stochastic elements cannot be accurately described by a mathematical model and cannot be evaluated analytically as modern manufacturing systems consist of many discrete operations that occur randomly and nonlinearly. Therefore, the objective function may not be expressible as an explicit function of the input parameters; hence, mathematical models or other methods are impractical.
2. Dynamic systems involve randomness that changes with time, such as an assembly line, where the components being assembled change with time. The modelling of complex dynamic systems theoretically requires too many simplifications, and the emerging models may not therefore be valid.
3. Purely analytical methods are often insufficient for optimisation because a mathematical model can only be built based on simplifying assumptions; therefore, accuracy often becomes a major problem for system optimisation.

In some cases, one must resort to simulation even though in principle some systems are analytically tractable; that is because some performance measures of the system have values that can be found only by running the simulation model or by observing the actual system. Consequently, the analytical effort required to evaluate the solution may be so formidable that computer simulation is the only realistic option.

Instead of using experts to build an extensive mathematical model by using the analytical approach, where the method of analysing the system is purely theoretical, computer-based simulation is used. Computer-based simulation is seen as an integral business tool giving flexibility and convenience in designing, planning and analysing complex manufacturing processes and/or systems. This is because the computer-based modelling and simulation method has the capability of representing the complex static structure as well as the dynamic behaviour of manufacturing systems [10–12]. As shown in Fig. 1, modelling and simulation for manufacturing systems is the technique of building an abstract logical model that represents a real system,

Q. Wang
Department of Mechanical Engineering, University of Bath,
Bath BA2 7AY, UK
E-mail: Q.Wang@bath.ac.uk

C.R. Chatwin (✉)
School of EIT, University of Sussex,
Falmer, Brighton BN1 9QT, UK
E-mail: C.R.Chatwin@sussex.ac.uk

and describes the internal behaviour of its components and their interactions including stochastic variability. This model, which is represented by a computer program that gives information about the system, can be used to mimic the operation of a real system, such as the day-to-day operations of an assembly flow line in a factory, and to predict the behaviour of complex manufacturing systems by calculating the movement and interaction of system components.

Almost any type of manufacturing system can be modelled as a discrete event system. Discrete event systems are dynamic systems, which evolve in time by the occurrence of events at regular and irregular time intervals; examples are flexible manufacturing systems, production assembly lines and traffic transportation systems. Although there are a number of modelling and simulation tools available for analysis and evaluation of manufacturing systems, when dealing with complex manufacturing systems it is often difficult to find the most effective way to describe the functions that must be performed and the relationships between these functions. In simulating manufacturing systems, we are concerned with systems in which performance is principally affected by competition for resources (machines, workers, material-handling devices, etc.). There are several basic problems when trying to model these systems: determining the resources and their characteristics that most affect performance; formulating a model or description representing these resources and their relationships; and determining the values of the performance measures of interest under given scenarios.

Hence, computer-based simulation of manufacturing systems focusses on:

1. Design and analysis of factory layouts, equipment decisions, alternative operating policies, problem evaluation, etc. These traditional management requirements are frequently assisted by computer-based simulation models.
2. Scheduling, particularly with automated systems. This allows the decision-maker to explore and plan changes to the existing schedule and/or to find the optimal schedule starting with current conditions. For example, current conditions may include the fact that a particular piece of equipment has broken down. The model would then generate an alternative schedule that would be used until the equipment has been repaired [7, 11, 13–21].

A typical stochastic system has a large number of control parameters that can have a significant impact on the performance of the system. A simulation model that explicitly tries to capture the important random components of the system is called a random or stochastic model. Discrete event simulation models (which represent the system) typically have stochastic components that mimic the probabilistic nature of the system.

Almost all manufacturing systems are stochastic rather than deterministic. Common sources of randomness in manufacturing systems are:

1. Arrival times of entities, i.e., parts or raw materials moving throughout the system
2. Processing or assembly times at each workstation or machine for different types of parts

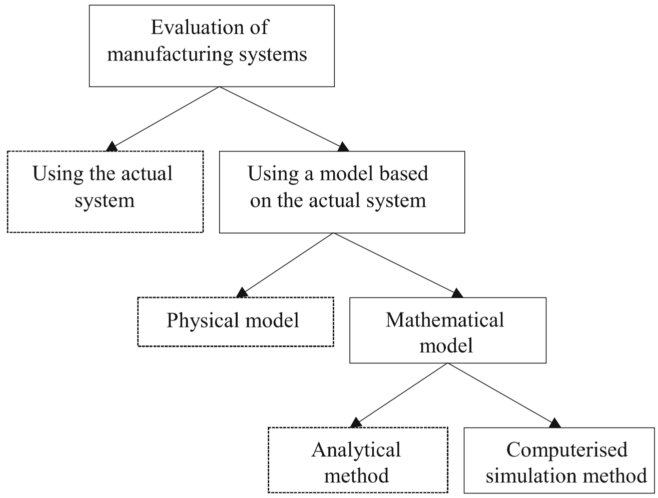


Fig. 1. Simulation methods to evaluate systems

3. Operation times for each workstation or machine without failure or breakdown
4. Repair times for system failures or breakdown
5. Set-up times for systems

Stochastic models typically depend upon various uncertain and uncontrollable input parameters that must be estimated from existing data sets. Therefore, the inputs for the models are critical elements and must be determined properly. Examples of such input include the arrival rate of entities to the system, the processing times required at various machines, reliability data (e.g., the pattern of breakdowns of machines), the time needed to repair machines, etc. Thus, input data analysis is one of the most important procedures in simulation tasks.

The input data analysis involves modelling an element (e.g., arrival, process and service times) in a discrete event simulation given a data set collected on the element of interest. This stage performs intensive error checking on the input data, including external policy, random and deterministic variables. Successful input data requires a close match between the input data model and the true underlying probabilistic mechanism associated with the system. To establish a basic knowledge of the behaviour of a system under variation of input parameter values and to estimate the relative importance of the input parameters, sensitivity analysis applies small changes to the nominal values of input parameters. Sensitivity analysis is concerned with evaluating sensitivities (gradients, Hessian, etc.) of performance measures with respect to parameters of interest. It provides the guidance for design and operational decision and plays a pivotal role in identifying the most significant system parameters as well as bottleneck subsystems [4, 8, 10, 12, 14, 20].

2 Modelling system randomness

Figure 2 presents an example of statistical procedures using an ARENA 'input analyser' facility to analyse and process the external modelling (empirical) data in terms of a histogram to fit and to form a standard distribution for model uses. The right window shows the input data and the left window displays the entire shape of the histogram that conforms to a normal distribution. The bottom window displays a summary report of the recommended distribution.

The main idea of statistical inference is to take a random sample from a population (i.e., the entire group from which we may collect data) and then use the information from the sample to make inferences about particular population characteristics such as the mean (measure of central tendency), the standard deviation (measure of spread) or the proportion of units in the population that have a certain characteristic. A sample is generally selected for study because the population is too large to study in its entirety. The sample should be representative of the general population. This is best achieved by random sampling.

Because a sample examines only part of a population, the sample mean will not exactly equal the corresponding mean of the entire population. Thus, an important consideration for those planning and interpreting sampled results is the degree to which the sample produces an accurate estimate of reality. In practice, a confidence interval is used to express the uncertainty in a quantity being estimated. Inferences are based on a random sample of finite size from a population or process of interest. Therefore, one gets different data (and thus different confidence intervals) each time [4, 10, 12, 20, 22].

The sampling distribution is the probability distribution or probability density function of the statistic. It describes probabilities associated with a statistic when a random sample is drawn from a population. If the parameter in a system varies continuously then it is possible that it conforms to one of the standard statistical probability distributions, such as: uniform, normal, exponential, or Poisson. Thus, this behaviour can be sampled from a distribution.

For instance, operation times at a workstation can be sampled from a distribution. First, the type of distribution must be determined, and its parameters must be calculated. To do that, the actual operation times are studied and plotted as a frequency distribution. If the shape of the distribution suggests that it does conform to one of the standard distributions, the 'goodness of fit' of the observed data can be assessed and the parameters for that distribution can be computed. If the frequency distribution of the actual times do not conform to a standard distribution, the observed data can be expressed as a histogram and samples drawn from that. It could also be sampled from the histogram giving

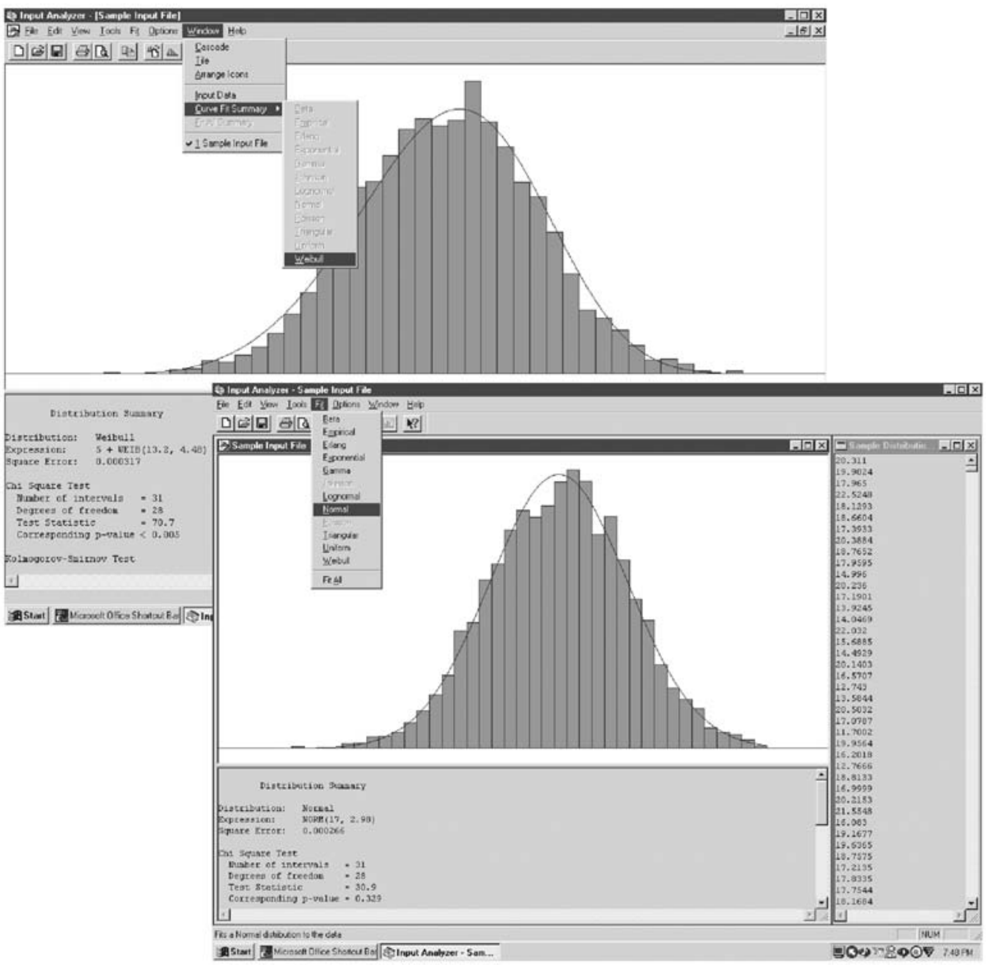


Fig. 2. Fitting empirical data as a sample distribution using 'Input Analyser'

the probability of an operation being performed at each workstation [7, 22–24].

2.1 Input data acquisition and analysis for stochastic system models

The essence of this procedure is abstraction and simplification; the real difficulty in modelling is to determine which elements should be considered and included in the model [24–26]. For establishing a flexible manufacturing system (FMS) model, these inputs could be abstracted by considering:

1. The basic configuration of the FMS, and its production scheduling, which defines the entities and activities involved in the model and the logic sequences that occur for each activity.
2. The number of workstations or machines that should be included in the simulation model.
3. How many types of processed parts need to move through the FMS; do they have similar processing requirements or not?
4. Buffer capacities for each machine.
5. Transport: conveyor or AGV and their track.
6. Profile of operations allocated to each workstation or machine.

Once these elements, together with logical functional relationships and their relevant descriptive information (descriptive variables), are determined, the simulation model can be built as a logical flow block (or pseudo-code) to describe and represent the real system to be investigated [12, 16, 27–29].

The authors believe that the input data collection and analysis play a key role in successful implementation of simulation model construction and simulation execution. Typically, more than one third of project time is spent on identification, collection, validation and analysis of input data. Although very little research work has paid attention to the development of systematic approaches to input data gathering, a number of researchers have raised issues surrounding data collection [10]. Basically, the quality of available data is a key factor in determining the level of detail and accuracy of the model.

Stochastic models typically depend upon various uncertain parameters that must be estimated from existing data sets if available; otherwise, if the data does not exist they can be sampled directly from theoretical probabilistic distributions. With manufacturing systems, there is no standard method for collecting the required information [24]. Data resources can possibly be collected from a literature survey, interviews with domain experts, industrial data reviews and state of the art assessments.

System design documentation includes data such as: drawings, specifications, production records and so on; it is important that such data reflects the current configuration of the system. Although these resources are usually reasonably accurate, they may be inaccurate or insufficient as historical records often do not represent the performance of the current system. Even though there is frequently copious data from reliable sources, simulation experts always argue over how we should use the data. If we sample directly from the empirical data, we may faithfully replicate the past but no values other than those experienced in the past can occur. If we fit the data to a theoretical distribution and then sample from it, the simulation may give values either bigger or smaller than the historical data, so the accuracy of representing the system is in doubt. This debate still continues, and an appropriate solution is still unclear.

If empirical data is to be used, it is input in the form of a cumulative probability distribution, which can be plotted by appropriate tools such as the so called 'Input Analyser' which arranges data in ascending order, grouping identical values and computing their relative frequencies. To organise raw data, first the collected data can be summarised and grouped into classes or categories so that we can determine the number of individuals belonging to each class. The observed number is called the class frequency. We can then form frequency distributions by determining the largest and smallest numbers in the raw data, thereby defining the range and breaking the range into a convenient number of equal class intervals. Next, we can determine the number of observations falling in each class interval to find out the class frequencies, and then the frequency distribution can be graphically plotted as a histogram, which represents a relative frequency distribution. Several excellent software packages, including ARENA, can perform these functions. These packages can simplify manual tasks in selecting and evaluating a distribution for model input data [4, 10, 14, 20].

The most difficult case in simulation studies is when the data for modelling systems does not exist either because the system does not exist or because it is not possible to obtain the data. Nevertheless, there are a number of possibilities to get data input for systems' models: estimation or theoretical distributions.

Vendors, designers and modellers can make the estimations. This greatly depends on factors from different people who have different experiences and use different measurement systems. The research has shown that people are very poor at estimating events even though they are very familiar with the systems. Therefore, the input data based on estimations may be highly unreliable; also in many cases it is hard to estimate. Instead, more popularly, we can choose a probability distribution based on theoretical considerations, i.e., using well-known statistical knowledge, so that we only need to determine how close this distribution is to reality by specifying the appropriate parameter values associated with the specific system [4, 28, 29].

One of the important skills of a simulation expert is to know how to summarise the data, to simplify the modelling process and to minimise the sensitivity of the results to errors in data estimates. Thanks to past studies of the industrial engineering statistics, we already know many statistical distribution functions that can be used particularly to 'represent' (or generate) various types of activity in industrial processes. For instance, it is already known among simulation experts that for a random process, inter-arrival times of customers (assembled parts) normally follows the exponential distribution, represented as EXPO. (ParamSet), which is thus often used to model random arrival times of events (and breakdown processes), but it is generally inappropriate for modelling process delay times. Also, the exponential distribution is typically not a good choice for representing service times, as most service processes do not exhibit the high variability that is associated with the exponential distribution. The normal distribution is used for the processing times when the mean is at least three standard deviations above zero. The uniform distribution is used when all values over a finite range are considered to be equally likely, which is generally used to represent 'worst case' results.

Each distribution has one or more parameter values (mean, standard deviation, etc.) associated with it. However, the parameter values associated with relevant distributions are also based on statistical estimations that often depend on the phenomena being represented. For example, the mean value of inter-arrival times can be estimated, if the times vary independently and randomly, and the estimated value is not large, then the time between arrivals can be modelled as an exponential distribution. This estimation can be considered reasonable [4, 14, 22].

As discussed above, the determination of what data to use is a very difficult and a time-consuming task. Regardless of the method used to collect the data, the decision of how much to collect is a trade-off between cost and accuracy. Perera [10] has summarised and ranked a number of factors that affect accuracy of analysis and identification of the collected data, namely:

1. Poor data availability
2. High-level model details
3. Difficulty in identifying available data sources
4. Complexity of the system under investigation
5. Lack of clear objectives
6. Limited facilities in simulation software or packages to organise and manipulate input data
7. Wrong problem definitions

2.2 Simulation model programming translation, validation, verification and execution

The development of an appropriate conceptual, logical simulation model by programming is one of the major tasks in simulation model construction. Although there are many simulation languages commercially available and there are hundreds of other locally developed languages being used by companies and universities, the trend for simulation software development has been an emphasis on an integrated simulation environment to provide ease of use [30–32].

Figure 3 shows an example of part of the logic program to build a model of a printed circuit board assembly (PCBA) system based on ARENA using two approaches. As shown in Fig. 3, an ARENA model that is constructed by placing and connecting modules, which have already been developed individually as integrated 'blocks or modules' using the SIMAN simulation language to represent distinct process modelling functions in the model window. The appropriate input data can be entered through the modules' dialogues. A model is constructed by selecting standard modules from the available set. The blocks are arranged and linked in a linear logical sequence, based on their functional operation and interaction, to depict the process through which the entities move in the system.

The definition of the model boundary is usually a trade-off between accuracy and cost. However, a valid model should include only those aspects of the system relevant to the study objectives.

Model verification is a process of determining the computer code of a model to ensure that the simulation program is a correct implementation of the model. This process does not ensure that the model appropriately represents the real system; it only ensures that the model is free of errors. Validation is concerned with the correspondence between the model and reality, i.e., model validation is a process of determining that a model is a sufficiently adequate approximation of the real system that the simulation conclusions drawn from the model are correct and applicable to the real-world system.

Although most simulation tools can automatically detect certain types of errors introduced by a programmer and may be able to display intentional errors in a model's logic, it cannot automatically correct or debug the errors. It is also unable to find errors of the model to represent the system, even though its program is correct. Furthermore, a manual verification process is used to avoid common errors, such as: data errors, initialisation errors, errors in the units of measurement, flow control, blockages and deadlocks, arithmetic errors, overwriting variables and attributes, data recording errors and language conceptual errors. It is found to be very useful to detect and expose such errors by running animation as a verification aid; such direct observation of errors in model execution, speeds the debugging process.

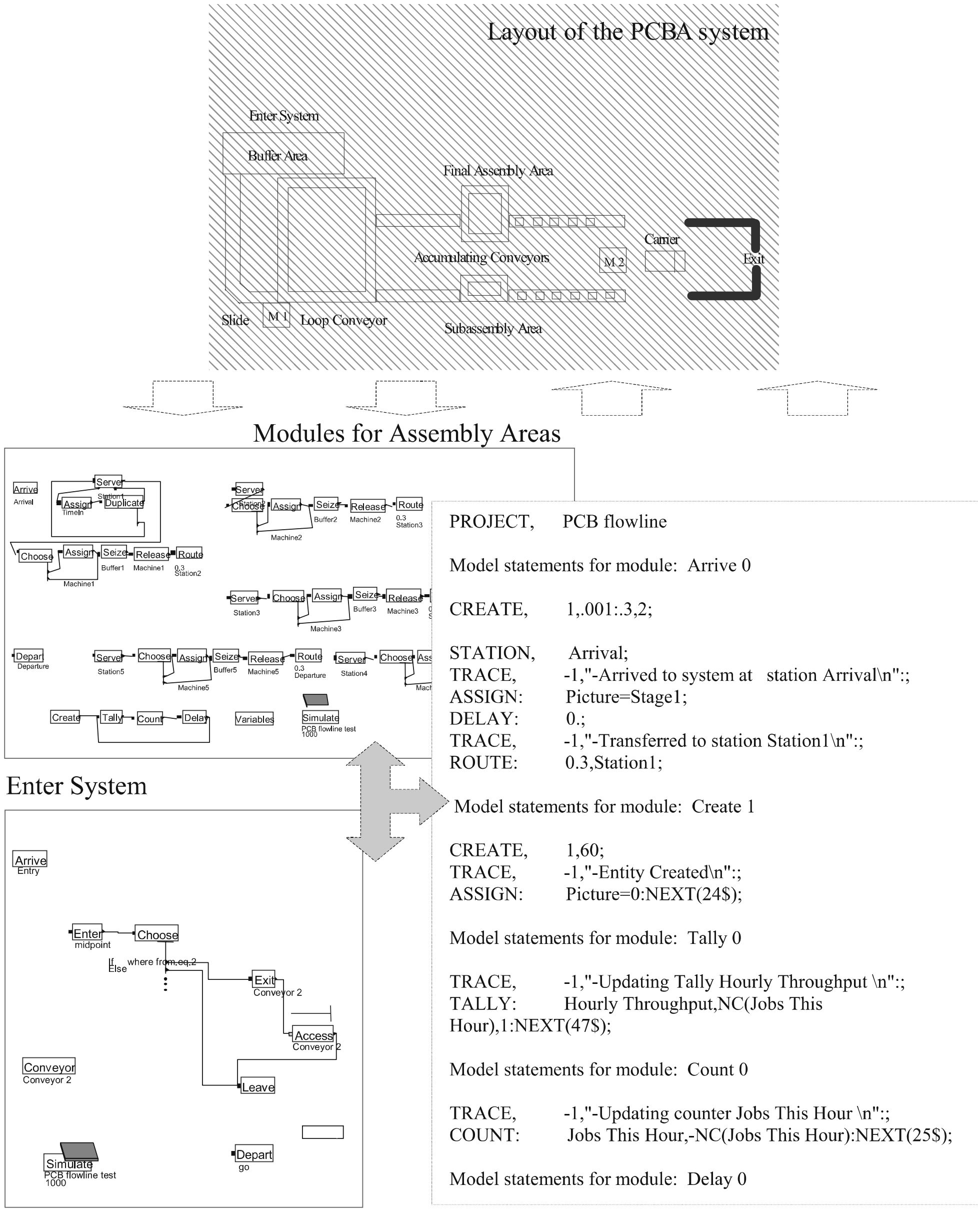
The increasing size of the systems and designs requires more efficient simulation strategies to accelerate the simulation process. At present, parallel and distributed simulation approaches seem to be promising moves in this direction. Topics currently subjected to intensive investigation are: synchronisation, memory management, randomised and reactive or adaptive algorithms, partitioning and load balancing [4, 12].

3 Simulation-based optimisation techniques

The purpose of simulation is not only performance evaluation but also optimisation. Although computer-based simulation modelling has emerged as a powerful tool for the analysis of complex systems and processes, it is not easily used to optimise processes. Because the objective functions (throughput, machine utilisation, etc.) of simulation models are not explicitly expressed in terms of the decision variables but rather as outputs of simulation replications, any attempt to find an optimal solution is complex and this is without having to consider the additional complexity of the stochastic nature of the simulation output.

For instance, discrete event simulation is the primary analysis tool for designing complex systems; however, it often needs to be linked with optimisation techniques to be used effectively for systems design. For a simulation model and for each set of feasible input parameters to its simulation experiment, the output, which is stochastic, is not optimised. Hence, optimisation techniques, such as genetic algorithms (GA), need to be employed to optimise objective functions. Genetic algorithms have been widely applied to the optimisation of asynchronous, stochastic automatic as-

Fig. 3. Two approaches for ARENA model programming



sembly systems, such as assembly line balancing problems in a flow-type production line that manufactures a variety of parts by optimising the buffer capacity between each machine tool [33].

The following is based on a literature survey of some of the above techniques that are mostly used for optimisation of manufacturing systems [12, 34–38].

3.1 Heuristic search techniques

The heuristic search technique is widely used along with mathematical analysis in optimising manufacturing systems performance, such as in optimisation of the assembly system performance. It is also the least sophisticated scheme mathematically, and it can be thought of as an intuitive and experimental approach.

The analyst determines the starting point and stopping rule based on previous experience with the system. After setting the input parameters (factors) to levels that appear reasonable, the analyst makes a simulation run with the factors set at those levels and computes the value of the response function. If it appears to be a maximum (minimum) to the analyst, the experiment is stopped. Otherwise the analyst changes parameter settings and makes another run. This process continues until the analyst believes that the output has been optimised. However, if the analyst is not intimately familiar with the process being simulated, this procedure can turn into a blind search and can expend an inordinate amount of time and computer resources without producing results commensurate with the effort. Heuristic search can be ineffective and inefficient in the hands of a novice.

3.2 Pattern search techniques

Pattern search techniques assume that any successful set of operations used in searching for an approximated optimum is worth repeating. These techniques start with small steps; then, if successful, the step size is increased. Alternatively, when a sequence of steps fails to improve the objective function, this indicates that shorter steps are appropriate, so we may not overlook any promising direction.

These techniques start by initially selecting a set of incremental values for each factor. For instance, starting at an initial base point, they check whether any incremental changes in the first variable yield an improvement. The result of improved settings becomes the new intermediate base point. One repeats the process for each of the inputs until one obtains a new setting where the intermediate base points act as the initial base point for the first variable. The technique then moves to the new settings. This procedure is repeated, until further beneficial changes cannot be made with the given incremental values. Then, the incremental values are decreased, and the procedure is repeated from the beginning. When the incremental values reach a pre-specified tolerance, the procedure terminates, and the most recent settings are reported as the solution. Pattern search techniques include conjugate direction search, steepest ascent (descent), and Tabu (Taboo) search technique, etc. Among them, the most effective technique in achieving local optimality for discrete optimisation is the Tabu Search technique.

3.3 Genetic techniques

In today's short cycle time production environments genetic techniques (GT), also called genetic algorithms (GA), are most frequently used to optimise manufacturing systems' effectiveness while simulation serves as a system performance evaluation tool (such as using ARENA). As a powerful and broadly applicable stochastic search and optimisation technique, GT have successfully been applied in various areas of industrial engineering in manufacturing, such as production scheduling and sequencing, reliability design, vehicle routing and scheduling, group technology (GT), transportation and many others. However, to evaluate these complex systems, this technique must be used together with simulation modelling techniques. Therefore, this combined method is also called the GA-enhanced simulation technique.

Genetic techniques, first proposed by Holland (1975), are heuristic search and optimisation techniques that imitate the principle of natural selection and genetic biological evolutionary processes. Furthermore, genetic algorithms are optimisers in that they use evolutionary techniques (computationally) to optimise a system that is too difficult for traditional optimisation methods. Evolutionary techniques are robust optimisers as they utilise nature's optimisation mechanisms to find acceptable solutions to intractable problems.

Precisely, a GT is an adaptive search algorithm that operates with a population of 'individuals'; each individual is assigned a 'goodness or fitness score' and represents a potential solution according to how good a solution it is to a given problem. It seeks to produce superior (fitter) individuals (solutions) by combining the better of the existing ones (through the mechanics of natural selection and genetics). As an example, the fitness score for manufacturing systems optimisation can be the utilisation and the buffer sizes required at each machine, where each machine is represented as an individual in the population. Therefore, the factors that are used to define the goodness of each solution are crucial for developing the genetic algorithms, which should aim at the incorporation of various performance variables in the manufacturing system. They are such factors as machine utilisation, queue length, buffer size at each of the machines and material handling costs while satisfying the total demand.

GT differs from traditional optimisation procedures in that GT works with a coding of the decision parameter sets, not the parameters themselves. GT searches a population of points, not a single point. GT uses objective function information, not derivatives or other auxiliary knowledge and finally; GT uses probabilistic transition rules, not deterministic rules. GT are probabilistic search optimising techniques that do not require mathematical knowledge of the response surface of the system which they are optimising.

GT is well suited for qualitative or policy decision optimisation such as selecting the best queuing disciplines or network topologies. It can be used to help determine the design of the system and its operation and find an optimal solution. The areas of application of GT involve inventory systems, job-shop, and computer time-sharing problems. GT does not have some of the shortcomings of other optimisation techniques, and it will usually result in superior optima to those found when using the traditional techniques. It can search a response surface with many local optima and find (with a high probability) the approximate global optimum. One may use GT to find an area of potential interest and then resort to other techniques to find the optimum.

GT-enhanced simulation is now being used in such tasks as machine learning, job scheduling, engineering design and assembly line planning. Assembly line design in particular has attracted much attention over the years. Figure 4 shows how the genetic algorithm is used to optimise simulation results in optimisation of a semi-automatic assembly line for compressors. The system variables of interest are encoded as genes of chromosomes, which are decoded and input into the simulation model along with the mandatory simulation parameters. Therefore, GA-enhanced simulation techniques have the ability to optimise the operation of the assembly line in such tasks as machine utilisation, throughput and tardiness.

Another similar technique called simulated annealing (SA) borrows its (evolutionary technique) basic ideas from statistical mechanics. SA as an optimisation technique was first introduced to solve problems in discrete optimisation, mainly combinatorial optimisation. Subsequently, this technique has been successfully applied to solve optimisation problems over the space of continuous decision variables. SA is a simulation optimisation technique

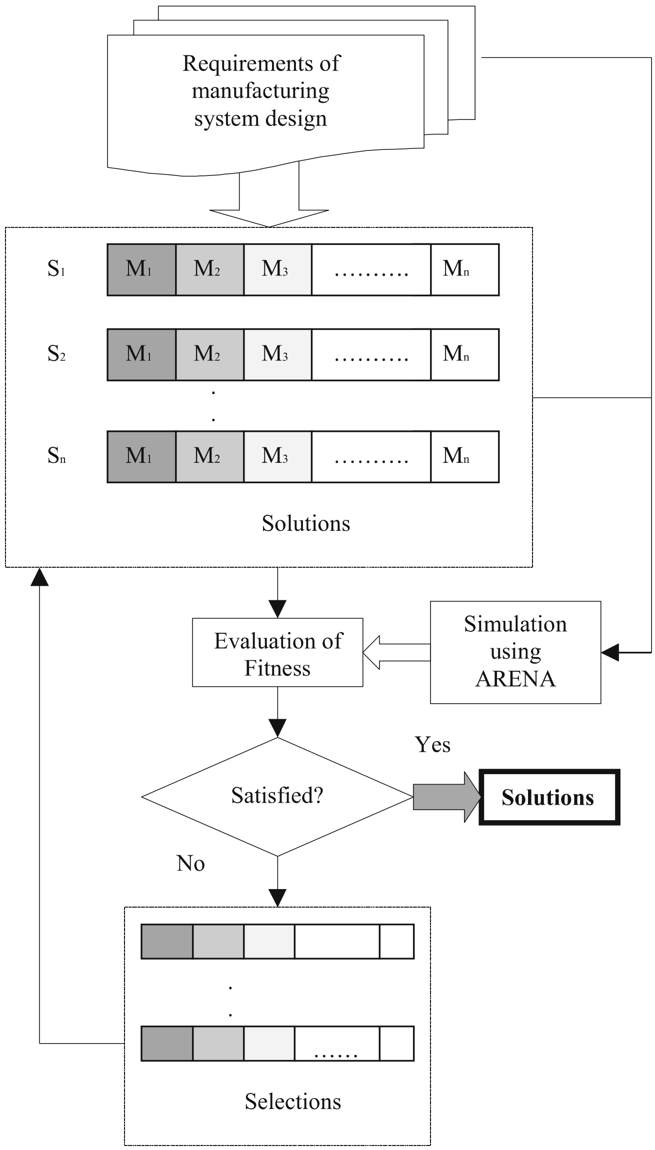


Fig. 4. Genetic algorithm-enhanced simulation technique

that allows random ascent moves in order to escape the local minima, but a price is paid in terms of a large increase in the computational time required. It can be proven that the technique will find an approximated optimum. The annealing schedule might require a long time to reach a true optimum.

4 Software selection and web-based simulation techniques for evaluating manufacturing systems

Analysts using computer-based simulation methods may develop their simulation models for manufacturing systems or industrial processes by means of general-purpose computer language at different levels: FORTRAN, C, C++, Java and MATLAB, etc., or advanced simulation languages: GPSS/H, SIMAN, Visual SLAM (SLAM II), and Simscript II.5. All these general algorithmic languages are capable of expressing the desired model. They are languages developed for discrete events or combined discrete and continuous simulations.

4.1 Simulation software selection and classification

Nevertheless, a vast amount of well-developed commercial tools and simulation packages for discrete event simulation are available for users in the market today. The simulation model is automatically created using high level modelling languages and notation that allows the user to validate and optimise the line performance, including throughput, bottlenecks, resource utilisation and buffer sizing. These models permit evaluation of different manufacturing scenarios and maximisation of their throughput potential [20]. Some major simulation software tools, which are widely used for modelling and simulation of manufacturing systems, are summarised below through a comprehensive survey.

The most popular simulation packages in recent years include: SIMAS II, which is devoted to the simulation of industrial mass production installations using automated assembly lines. WITNESS provides a graphical environment to design discrete event simulation models. It allows simulation experiments to optimise material flows across the facilities and generates animated 3D virtual reality models. SIMUL8 is mainly used for discrete event simulation. By providing a user-friendly visual interface, SIMUL8 allows the user to pick from a predefined set of simulation objects and statistical distributions to create the system's model; it also allows hierarchical modelling. Taylor ED (Taylor enterprise dynamics) is an object-oriented software application used to model, simulate, visualise and monitor business processes, whether the process is manufacturing, material handling, logistics or administration. ShowFlow is designed to model, simulate, animate and analyse processes in logistics, manufacturing and material handling. It provides powerful visualisation and reporting tools, in particular for simulation animation. The model is facilitated by the availability of many simulation components ready to run. ARENA is developed based on the SIMAN modelling language, which has an object-oriented design and the ability to be tailored to any application area. The original version of ARENA was called SIMAN. Many of the basic concepts included in SIMAN are based on the previous work of other simulation language developers such as GPSS at IBM in the process-orientation and GASP at US steel. SIMAN also contains features from SLAM. The SIMAN language was originally designed to be a general purpose modelling language, but the design also includes many special purpose manufacturing features to make the language particularly useful in modelling large and complex manufacturing systems. Compared to others, ARENA is a simulation package that is specially used to model and simulate manufacturing environments. It combines most features of the other packages referred to [39, 40].

All the above packages are specially dedicated software for modelling, simulating and evaluating the manufacturing environment. They can help determine plant capacity, measure utilisation of resources, balance manufacturing production lines, identify and manage bottlenecks, solve inventory and WIP problems, verify designs of manufacturing systems and test new scheduling practices, optimise production rates or processes and justify capital expenditures.

However, each of these packages has its own user interface for building models, consequently: building, running and analysing simulation models by using different tools can be a very time-consuming and error-prone process. The National Institute of Standards and Technology (i.e., NIST) in the manufacturing systems integration division (i.e., MSI) in the USA has proposed the development of neutral libraries of simulation components and model templates, which contain detailed formal information models of all commonly used simulation components (queues, machines, transportors, and so on). Each of these component models will form different modelling templates such as an equipment simulation, a material flow simulation, a supply chain simulation and so on [24, 41]. This work may simplify the model-building process, reduce complexity, enable component-based modelling, and speed Internet-based simulation services in the future.

4.2 Web-based modelling and simulation techniques

The authors believe that the aim of concurrent engineering systems can only be fully realised when the protagonists can apply internet-based technologies, which are able to link the client's remote workplaces directly to the hub facilities that allow data exchange between remote independent systems so that any changes on new product designs from CAD/CAM databases may be immediately implemented. When this technology is fully implemented, web-based simulation will play a key role in the manufacturing design environment. Figure 5 depicts an implementation framework for web-enabled concurrent engineering.

Web-based simulation (or WebSim) represents the marriage of web technologies and simulation science; it is a convergence of computer simulation methodologies and applications within the World Wide Web (WWW). The major benefit of web-based modelling and simulation is that it provides an open channel between a simulation service provider and a simulation client. Hence, the service provider can offer a simulation and application software warehouse (models and tools) on its web servers. Clients can work with and modify models or make new models. The models can run applets on the client machine or on the provider's server; this mode of operation is called distributed simulation. In another scenario, the service provider can build a customised model and carry out a complete simulation study. This is the most commonly used approach associated with the term web-based simulation. This includes both the remote execution of existing simulations from a browser through HTML forms and CGI scripts and the development of mobile code simulation (e.g. applets) that run on the client side.

There are many possible future application areas for simulation over the web. Web-based simulation will not only deliver 'distributed simulation' or 'simulation documentation' for manufacturing and business. The introduction and widespread use of the web suggests that there are many new areas where web science and technology will utilise simulation to generate new application scenarios.

At present, there are many successful applications of e-commerce or e-business, where consumers and businesses use the Internet for e-mail, searching, advertising, selling and buying. Nevertheless, there is evidence from the first implementations of business-to-business (B2B) developments, that failures have vastly outnumbered successes, many applications using internet-based techniques are still in development and are focussed on practical applications. The development of applications for rapid manufacturing using internet-based technologies, i.e., e-manufacturing, represents a typical example that is involved in internet-supported integration of design, manufacturing and supply to achieve agile manufacturing and resource optimisation; this technology will ultimately include web-based workflow modelling and simulation methods for manufactur-

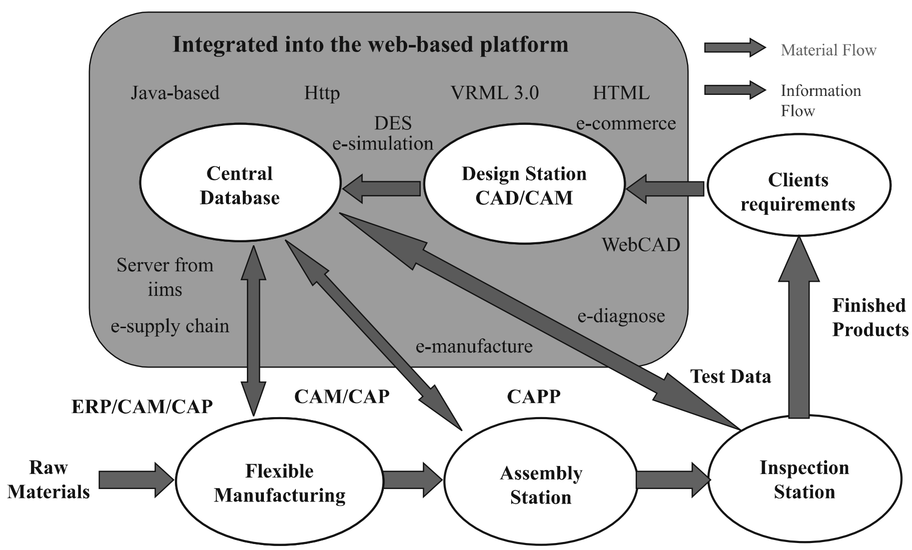


Fig. 5. Web-enabled concurrent engineering

ing products or systems design etc. In e-manufacturing, the manufacturing data from CAD/CAM systems enable separate design engineering tasks to be performed concurrently by the updating of common design databases remotely controlled by computers.

Thanks to advances in technology, in only a few years web-based (Internet) simulation has quickly emerged as an area of significant interest for both simulation researchers and simulation practitioners. This interest in web-based simulation is a natural outgrowth of the proliferation of the world wide web and its attendant technologies, e.g., HTML, HTTP, CGI, etc., and the surging popularity of computer simulation as a problem solving and a decision support systems tool.

The appearance of network-friendly object-oriented programming languages, such as XML, Java and C++, for internet applications and of distributed object technologies, like the 'common object request broker architecture' (CORBA) and the 'object linking and embedding or component object model' (OLE/COM), have had a major impact on the state of simulation practice. For instance, CORBA provides a support platform for the building of distributed systems. It allows a user to invoke objects located in other machines, handles all information conversions required and allows the communication between heterogeneous machines.

A number of discrete event simulation tools based on Java have been developed recently including Simjava, Silk, DEV-Java, JSIM, JavaSIM, JavaGPSS etc. A successful example in development for web-based discrete event simulation tools is called Silk (developed and marketed by ThreadTec Inc.). Silk is a general purpose simulation language based on the Java programming language with object-oriented language features, i.e., it is a Java-based modelling tool for the simulation, study and improvement of computer, manufacturing and industrial systems and is a modelling tool that merges the process description modelling methodology within an object-oriented language. In the Silk simulation language, models are developed directly in the java programming language using a package of classes consisting of a relatively few powerful process-oriented modelling features. The key feature that makes Silk very attractive is that it allows modellers to develop domain specific simulation objects using the JavaBeans methodology. JavaBeans are software components written in Java; these components are self-contained, reusable software units that can be visually composed into applets or applications using visual application builder tools. JavaBeans provide a Java platform that has opened up an entirely new world of opportunities for building fully portable network-aware applications. The JavaBeans component architecture is a platform-neutral architecture for the Java application environment. The JavaBean architecture extends 'write one, run anywhere' capability to reusable component development, this facilitates the development of web-based simulation.

In the Silk environment, the simulation entities, resources and queues are provided as classes. The modeller develops problem-specific entities, queues and resources by directly extending and inheriting these classes from Silk. Silk also provides animation features by providing a set of JavaBeans that have animation features. Entities, queues and resources defined by the modeller can be linked to these animation JavaBeans to provide process animation. In addition, other Silk classes can be used to perform other routing simulation tasks such as random distribution generation, statistics collection and output generation. Silk has been designed in such a manner that it can be extended and customised for a particular domain such as manufacturing industry. This can be accomplished by developing JavaBeans and integrating them into a simulation model developed using Silk.

Due to the above features of Java-based Silk, it can be chosen as a support tool to implement 'virtual reality applications', such as a virtual factory, which is a system that allows the modelling and simulation of a factory based on a virtual reality environment and simulation tools within a web-based simulation environment. The continuous development of this technology is still ongoing; a potential solution for a standardised virtual model is the most exciting development of VRML, i.e., the Virtual Reality Modeling Language, a new addition to the World Wide Web. The first version of VRML (VRML 1.0) was developed by a consortium of computer graphics professionals and became available on the World Wide Web in 1995. It quickly gained broad support and, after redefining the language, the specifications for VRML 2.0 followed in 1996. This version, with a few minor differences, became an international standard (ISO/IEC 14772) in 1997 under the name VRML97. While HTML, the Hyper-Text Markup Language, is the current standard for authoring home pages, VRML supports the distribution of three-dimensional models over the Internet. These models have all the characteristics of virtual models (as described above). They are based on a polygonal representation and can be animated, they can include functionality and dynamic behaviour and can be interactively controlled by the user.

VRML was not originally specifically designed for engineering applications in design, manufacturing and operational simulation etc. Using VRML on the World Wide Web provides an excellent tool for sharing virtual models with remote users and for supporting collaborative work and concurrent engineering. It is extremely cost effective since the required infrastructure (networked computers) exists almost everywhere and the viewing software (VRML plug-in) is available to everyone. Today's limitations are dictated by network capabilities (download times for large VRML files describing complex virtual models) and the speed of the user's local computer (responsible for real-time rendering and interactions). The current development trend towards high capacity networks like the Internet and more powerful desktop and laptop computers with 3D graphics acceleration will remove theses limitations in the near future.

Although there are obvious advantages, the challenge comes in fully implementing the system including details of product data management (PDM), network security, bandwidth and overall system reliability. A key problem with the use of the WWW as a platform for distributed simulation model execution is transparency, i.e., the problem of transparent access to net resources.

Currently, researchers in the field of web-based modelling and simulation are focussed on the following issues:

- Java-based modelling and simulation
- Methodologies for web-based model development
- Collaborative model development over the Internet
- Distributed modelling and simulation using web technologies
- Education and training with web-based simulation
- Multimedia-enriched simulation modelling

However, analysts generally agree that web-based simulation applications may be successful only if it provides real value for industry. It seems to be a trend that, in the near future, it will be increasingly difficult to draw a line between techniques using web-based simulation, i.e., e-simulation, and traditional simulation [39, 42–56].

5 Discussion and conclusions

Clearly, simulation has become one of the most popular techniques applied to the analysis of complex manufacturing systems at both the justification phase and the design phase within the manufacturing life cycle. Therefore, many researchers continue to explore a range of modelling methodologies and tools, which contribute to established modelling frameworks. Nevertheless, these techniques are developed based on a common foundation or principle, which has been presented and discussed in this paper.

Furthermore, with the growing use of computer modelling and simulation, the scope of simulation domains must be extended to include much more than traditional optimisation. Optimisation techniques for simulation must also account specifically for the randomness inherent in estimating the performance measures and satisfy the constraints of stochastic systems for manufacturing systems' design, analysis and performance evaluation. This review has reported on the latest developments in optimisation techniques for manufacturing systems and summarised their merits or pitfalls.

A comparison based on several studies indicates pattern search techniques are more effective for constrained problems. Genetic techniques are robust and can produce near-best solutions for large problems. The pattern search technique is most suitable for small sized problems with no constraints; it requires less iteration than genetic techniques. The heuristic search technique is widely used with mathematical analysis in optimising manufacturing systems.

However, no single technique works effectively and/or efficiently in all cases.

Finally, we have reported on the most popular simulation tools available and their strengths for use for different aspects of modelling and simulation of manufacturing systems. Web-based (also may be called Internet) simulation techniques will become increasingly important for modelling and simulation of manufacturing systems.

References

1. Doumeingts G, Vallespir B (1995) Methodologies for designing CIM system: a survey. Comput Ind 25:263–280
2. AL-Ahmari AMA, Ridway K (1997) Computerised methodologies for modelling computer integrated manufacturing systems. Proceedings of 32nd International MATADOR conference, Manchester, pp 111–116
3. Mansharamani R (1997) An overview of discrete event simulation methodologies and implementation. Sadhana 22(5):611–627
4. Peden CD, Shannon RP, Sadowski RP (1995) Introduction to simulation using SIMAN, System Modelling Corporation, McGraw, NY
5. AL-Ahmari AMA, Ridway K (1999) An integrated modelling method to support manufacturing system analysis and design. Comput Ind 38:225–238
6. O'Kane JF, Spencekley JR, Taylor R (2000) Simulation as an essential tool for advanced manufacturing technology problems. J Mater Proc Technol 107:412–424
7. Chan FTS, Jiang B, Tang NKH (2000) The development of intelligent decision support tools to aid the design of flexible manufacturing systems. Int J Prod Econ 65:73–84
8. Guest Editorial (2001) Modelling, specification and analysis of manufacturing systems. Int J Prod Res 39(2):159–162
9. Kim CH, Weston R (2001) Development of an integrated methodology for enterprise engineering. Int J Comput Integr Manuf 14(5):473–488
10. Perera T, Liyanage K (2000) Methodology for rapid identification and collection of input data in the simulation of manufacturing systems. Simul Pract Theory 646–656
11. Borenstein D (2000) Implementation of an object-oriented tool for the simulation of manufacturing systems and its application to study the effects of flexibility. Int J Prod Res 38(9):2125–2142
12. Arsham H (2002) Systems simulation: the shortest path from leaning to applications. Teaching handout. EOF@1995–2002, University of Baltimore, USA
13. Zeigler BP, Chao TH, Rozenblit JW (1996) A knowledge-based simulation environment for hierarchical flexible manufacturing. IEEE Trans Syst Man Cybernetics 26(1):81–90
14. Tobaben F (1997) WITNESS modelling, simulation, and statistical analysis of an assembly line for Rolls-Royce Motor Cars Ltd. Project Report, Technical University of Braunschweig
15. Chryssolouris G, Anifantis N, Karagianis S (1998) An approach to the dynamic modelling of manufacturing systems. Int J Prod Res 38(90):475–483
16. Chou YC (1999) Configuration design of complex integrated manufacturing systems. Int J Adv Manuf Technol 15:907–913
17. Choi SH, Lee SL (2000) The design architecture and implementation of an integrated manufacturing system. Int J Comput Integr Manuf 10:232–244
18. Balduzzi F, Giua A, Seatzu C (2001) Modelling and simulation of manufacturing systems with first-order hybrid Petri nets. Int J Prod Res 39(2):255–282
19. Ikbal M, Rondeau E, Divoux T (2001) Industrial systems communications: design and integration. Int J Comput Integr Manuf 14(6):545–559
20. Kelton WD, Sadowski RP, Sadowski DA (20002) Simulation with Arena, McGraw College, NY
21. Cunha PF, Dionisio J (2002) An architecture to support the manufacturing systems design and planning. Proceedings of the 1st CIRP(UK) Seminar on Digital Enterprise Technology, Durham, UK, pp 237–240
22. Ostle B, Turner KV, Hicks CR, McElrath GW (1996) Engineering statistics: the industrial experience, Wadsworth, Stamford, CT, USA
23. Brooks-Roger J, Tobias AM (2000) Simplification in simulation of manufacturing systems. Int J Prod Res 38(5):1009–1027
24. Lee YT (2001) An overview of information modelling for manufacturing systems integration. Manufacturing Systems Integration Division and National Institute of Standards and Technology. Gaithersburge, MD PP. 20899–0001, USA
25. Fokkert JVDZ (1998) Process flow design for repetitive assembly. The BETA Research Institute

26. Baines TS, Harrison DK (1999) An opportunity for system dynamics in manufacturing system modelling. Prod Plan Control 10(6):542–552
27. Spathopoulos MP, Ridder MD (1999) Modelling and distributive control design of a flexible manufacturing system. Comput Ind 38:115–130
28. System Modelling Corporation, ARENA user guides (1994–1998), 1998
29. Chan FTS (2001) The effects of routing flexibility on a flexible manufacturing system. Int J Comput Integr Manuf 14(5):431–445
30. Mak KL, Lau HYK, Wong STW (2000) An object-oriented prototyping tool for automated manufacturing systems. Comput Ind 43:43–60
31. Little D, Yusuf YY (2000) Manufacturing control systems-moving towards the enterprise model. 5^{th} International Conference on FACTORY 2000, pp 388–395
32. Bernard A, Perry N (2002) Fundamental concepts of product/technology/process informational integration for process modelling and process planning. Proceedings of the 1st CIRP(UK) Seminar on Digital Enterprise Technology, Durham, UK, pp 237–240
33. Lee SG, Khoo LP (2000) Optimising an assembly line through simulation augmented by genetic algorithms. Int J Adv Manuf Technol 16:220–228
34. Rao HA (1997) Design methodology and integrated approach for design of manufacturing systems. Integrated Manufacturing Systems, 8/3 (1997), MCB University Press (ISSN0957-6061) pp 159–172
35. Grunow M (2000) Simulation-based performance analysis and optimisation of electronics assembly equipment. Int J Prod Res 38(17): 4247–4259
36. Azadivar F, Wang J (2000) Facility layout optimisation using simulation and genetic algorithms. Int J Prod Res 38(17):4369–4383
37. Dimopoulos C, Zalzala AMS (2000) Recent developments in evolutionary computation for manufacturing optimisation: problems, solutions, and comparisons. IEEE Trans Evol Comput 4(2):93–113
38. Zeigler BP, Chao TH, Rozenblit JW (1996), A knowledge-based simulation environment for hierarchical flexible manufacturing. IEEE Trans Syst Man Cybernetics 26(1):81–90
39. Nikoukaran J (1999) Software selection for simulation in manufacturing: a review. Simul Pract Theory 7(1):1–4
40. Perera T (2002) Enhancing the future of simulation UK Ltd. Proceedings of the 1st CIRP(UK) Seminar on Digital Enterprise Technology, Durham, UK, pp 237–240
41. Son YJ (1999) Automatic generation of simulation models from neutral libraries: an example. The Pennsylvania State University
42. Bateman R, Cheng K (2002) Web-based distributed manufacturing and customisation. Proceedings of the 1st CIRP(UK) Seminar on Digital Enterprise Technology, Durham, UK, pp 237–240
43. Maropoulos PG (2002) Digital enterprise technology: defining perspectives and research priorities. Proceedings of the 1st CIRP(UK) Seminar on Digital Enterprise Technology, Durham, UK, pp 3–12
44. Cantamessa M, Fichera S (2002) Process and production planning in manufacturing enterprise networks. Proceedings of the 1st CIRP(UK) Seminar on Digital Enterprise Technology, Durham, UK, pp 187–190
45. Scholz B, Hohns H (2002) Integrated software agents: enabling technology for collaborative e-logistics and e-business. Proceedings of the 1st CIRP(UK) Seminar on Digital Enterprise Technology, Durham, UK, pp 237–240
46. Fourie CJ (2002) Intelligent remote analysis and control for adaptive manufacturing. Proceedings of the 1st CIRP(UK) Seminar on Digital Enterprise Technology, Durham, UK, pp 247–249
47. Vigano G, Liao Z (2002) To build object-oriented virtual factory based on virtual reality and simulations tools. Proceedings of the 1st CIRP(UK) Seminar on Digital Enterprise Technology, Durham, UK, pp 225–228
48. Catamaran RH, Diego S (1998) What is web-based simulation? International Conference on Web-based Modelling and Simulation, California, USA
49. Cunha PF, Dionisio J (2002) An architecture to support the manufacturing system design and planning. Proceedings of the 1st CIRP(UK) Seminar on Digital Enterprise Technology, Durham, UK, pp 129–134
50. Sawhney A, Manickmam J, Mund A (1999) Java-based simulation construction process using Silk. Proceedings of the 1999 Winter Simulation Conference, USA, pp 985–991
51. Page E (1999) Special issues on web-based modelling and simulation. ACM Transactions on Modelling and Computer Simulation
52. Huang CY, Nof SY (1998) Formation of autonomous agent networks for manufacturing system. Int J Prod Res 38(3):607–624
53. Santos ETP, Fialho SV (2000) A web-based tutor for Internet communication protocols, Wiley, NJ, USA
54. Medeiros DJ, Damle A, Nanda J. Web-based simulation, IE 522-Discrete Event Computer Simulation, Pennstate, Potsdam, Germany, March 29 – April 4, 2000
55. Beier KP, Web-based virtual reality in design and manufacturing application. Proceedings of the 1st International EuroConference on Computer Applications and Information Technology in the Maritime Industries, Potsdam, Germany, March 29 – April 4, 2000
56. Lei PW, Lo LK, Chatwin CR, Young RCD, Heywood MI (2003) Security and Trust of Online Auction Systems in E-Commerce, In S. Nansi (Eds), Architectural Issues of Web-Enabled Electronic Business, Idea Group Publishing, ISBN 1-59140-049-X, pp 362–375