J. Automation & Systems Engineering

Regular paper

Simulation Software Evaluation and Selection: A Comprehensive Framework

Rajesh Verma, Ashu Gupta and Kawaljeet Singh

In a period of continuous change in global business environments, organizations, large and small, are finding it increasingly difficult to deal with, and adjust to the demands for such change. To accomplish business process change, most companies use different methods and tools which integrate components for static and dynamic modelling and measuring the performance of business processes. An increase in the use of simulation as a modelling and analysis tool has resulted in a growing number of simulation software products on the market. Companies are seeking advice about the desirable features of software for manufacturing simulation, depending on the purpose of its use. Because of this, the importance of an adequate approach to simulation software evaluation and selection is apparent. This paper presents a comprehensive framework which can be used for evaluation of simulation software packages. The evaluation criteria are grouped according to their nature, and they can be of practical use to anyone involved in simulation software evaluation and selection.

Keywords: Evaluation criteria, Simulation, Simulation software.

1. INTRODUCTION

One of the major uses of simulation is in the planning of new manufacturing operations. Wild and Otis (1987) describe the case of a company that considered the implementation of 77 new machine tools, for a new production line. However, when the operation was simulated, they found that 4 machines were actually needed, presenting a total saving of $750 000. Such savings are not unusual when simulation is used.

Simulation modelling is being applied to many areas such as manufacturing, material handling, health care, military, transportation, business process re-engineering and computer system performance. As a result, simulation is often used to make current operations more efficient, resulting in substantial savings. It is often used to help production systems become more efficient, with better on-line deliveries and less investment. Simulation allows many variables to be considered simultaneously. It can be used to solve problems that are too complex to analyze otherwise. Therefore, modelling the system, through its simulation, and then experimenting with it, leads to a better understanding of its operation (Rice, 1988). Simulation modelling can also be used as a tool for employee training and troubleshooting. As described by Wilkinson (1988), Polaroid's line supervisors are taught how to use simulation for identifying possible, or probable problems, thus allowing the development of appropriate corrective action. However, managements are often hesitant to accept the results of simulation. They are more comfortable with more familiar methods and may not believe that the simulation accurately represents their

Assistant Professor, Lovely School of Business, Lovely Professional University, Phagwara (Punjab) India. email: dr.rajeshverma@rediffmail.com

Senior Lecturer, Department of Computer Applications, Apeejay Institute of Management, Rama Mandi-Hoshiarpur Road Jalandhar-144023 India. email: guptashu1@rediffmail.com.

Director, University Computer Centre, Punjabi University Patiala, Punjab India. email: singhkawaljeet@rediffmail.com.

production system. Simulation in the past presented results in the form of large amounts of data, however, with modern use, simulation can be presented to senior management in understandable terms, for example, animated graphics.

Advances in technology, have allowed simulation to become a more widely available tool. In part, this is facilitated by the price of simulation software, which is more affordable to a greater number of companies. Also, simulation languages are more user-friendly. However, users need advice on how to select an adequate simulation software tool and which software features to examine during the evaluation process.

T his paper presents a comprehensive evaluation criteria derived for the evaluation of simulation packages, grouped according to their nature. In addition to an evaluation framework, a review of research studies in simulation software evaluation criteria is provided. The main findings of these studies are analyzed and their context for this research presented. The possible use of the derived criteria is briefly discussed and the methodology for simulation software selection is presented. The main findings of this research are summarized and conclusions subsequently drawn.

2. REVIEW OF LITERATURE

Haider and Banks (1986) addressed the issues related to the choice of simulation software products for the analysis of manufacturing systems and established the following desirable features for simulation software: input flexibility; structural modularity; modelling flexibility; macro capability and hierarchical modelling; materials handling modules; standard statistics generation; data analysis; animation; interactive model debugging; micro/mainframe compatibility; the support provided by the supplier; and the cost of simulation software.

Bovone *et al.* (1989) have proposed a simple method for the evaluation of simulation software, and attempt to determine whether or not commercial software tools are better than home-made products, and whether or not to prefer general purpose simulation software to an application oriented simulator. They proposed several criteria to be used for software evaluation including flexibility, ease of use, speed of modelling, debugging, portability, reliability and execution speed.

Ekere and Hannam (1989) described several criteria such as programming features, model development characteristics, experimental and reporting features, and commercial and technical features for the evaluation of manufacturing simulation software.

Law and Haider (1989) provided a simulation software survey and evaluation on the basis of information provided by vendors. Both simulation languages and simulators such as FACTOR, MAST, WITNESS, XCELL + and SIMFACTORY 11.5 are included in this study. Instead of commenting on the information presented about the software, the authors conclude that there is no simulation package which is completely convenient and appropriate for all manufacturing applications.

Kochhar and Ma (1989) addressed the essential and desirable features of simulation software for its effective use in manufacturing environments, providing criteria which should be used for the selection of manufacturing simulation software tools. These criteria relate to: modelling assistance; interactivity; graphics; a data handling capability; the time scale for model development; the learning curve and the required skills for the use of software; ease of model editing; portability; simulation speed; and interfacing the simulation package with external systems.

Breedam et al. (1990) described a study in which 11 evaluation criteria were identified and a sample of simulation packages assessed on these criteria. The resulting assessments provided the necessary input for a cluster analysis. The results of this cluster analysis were used to obtain some idea of positioning in the market of identified simulation packages. They said that this may help users to choose appropriate packages.

Holder (1990) recommended a feasibility study out of which the extensive software evaluation criteria can be constructed. Some of the groups of criteria derived are ease of coding, dialogue design, debugging, graphics, flexibility, processing, run-time considerations, costs and survivability of the software.

Banks *et al.* (1991) evaluated SIMFACTORY 11.5, XCELL +, WITNESS and ProModelPC by modelling two manufacturing systems. The main results of the evaluation revealed that SIMFACTORY 11.5 and XCELL + did not have robust features, while WITNESS and ProModelPC had most of them. Such conclusions were obtained on the basis of twenty two criteria.

Williams and Trauth (1991) ranked 30 manufacturing software packages. Each criterion was weighted from 0.1 to 1 and the features of each package were scored a number from 1 to 10. Finally Analytic Hierarchy Process (AHP) was used to find the best of the top three packages found from previous evaluation. AHP showed that their previous evaluation technique has been consistent for the top three packages.

Law and Kelton (1991) described the main characteristics and building blocks of AutoMod 11, ProModel, SIMFACTORY 11.5, WITNESS and XCELL +, with a limited critical evaluation based on featured criteria.

Mackulak and Savory (1994) carried out a questionnaire survey on the most important simulation software features. The most important features identified include: a consistent and user friendly user interface; database storage capabilities for input data; an interactive debugger for error checking; interaction via mouse; a troubleshooting section in the documentation; storage capabilities for simulation models and results; a library of reusable modules of simulation code; and a graphical display of input and output.

Nikoukaran *et al.* (1998) presented a comprehensive list of criteria structured in a hierarchical framework for evaluating and selecting simulation software. Issues related to criteria for simulation software evaluation and selections are categorized into seven main groups and several sub-groups. The hierarchy can be used for obtaining a better view of the features of simulation software and as a guide to test and analyze simulation modelling packages.

Hlupic and Paul (1999) presented criteria for the evaluation of simulation packages in the manufacturing domain together with their levels of importance for the particular purpose of use. They suggested general guidelines for software selection. They pointed that to expect a particular package to satisfy all criteria. However, it is indicated which criteria are more important than others, according to the purpose of software use.

Tewoldeberhan *et al.* (2002) proposed a two-phase evaluation and selection methodology for simulation software selection. Phase one quickly reduces the long-list to a short-list of packages. Phase two matches the requirements of the company with the features of the simulation package in detail. Different methods are used for a detailed evaluation of each package. Simulation software vendors participate in both phases.

Hlupic and Mann (2002) developed a software tool (SimSelect) that selects simulation software given the required features. It is evident form the material presented within this research that simulation modelling is the "cost-effective" method of exploring "what-if"

scenarios quickly, and finding a solution to or providing a better understanding of the problem, as this method is supported by a number of software tools (similar to Simul8) that provide a graphical representation of the business processes through executable models.

Seila *et al.* (2003) presented a framework for choosing simulation software for discrete event simulation. By evaluating about 20 software tools, the proposed framework first tries to identify the project objective, since a common understanding of the objective will help frame discussions with internal company resources a well as vendors and service providers. It is also prudent to define long-term expectations. Other important questions deal with model dissemination across the organization for others to use, model builders and model users, type of process (assembly lines, counter operations, material handling) the models will be focused, range of systems represented by the models etc.

3. SIMULATION SOFTWARE EVALUATION CRITERIA

The criteria derived can be applied to the evaluation of any general or special purpose simulation package. For this study four main groups are defined to develop the framework for the evaluation. Features within each group are further classified into subcategories, according to their character. The main categories are:

1. *Hardware and software considerations*: pedigree, coding aspects, software compatibility, user support, financial and technical features;
2. *Modelling capabilities*: general features, modelling assistance;
3. *Simulation capabilities*: visual aspects, efficiency, testability, experimentation facilities, statistical facilities;
4. *Input/Output issues*: input and output capabilities, analysis capabilities.

Owing to the comprehensiveness of the evaluation framework, individual criteria within each group are merely listed, and generally described in the context of a particular group. According to the type of each criterion, the classification determines whether, for example, a certain feature exists in the package, determines the quality of features provided, or lists types of alternatives available within a particular feature.

3.1. Criteria for Hardware and Software Considerations

3.1.1 Pedigree:

Criteria included in this group refer to the origin of the package and its importance. They also evaluate how widely the package is used, and judge the reputation of the software supplier. A supplier's reputation is a criterion which depends on many factors such as the length of time the supplier has been present in the software market, the number of employees and representative offices the supplier has, and the type and level of user support that is provided.

Table 1: Items for Pedigree

1.1.1	Sources of information about the package	O Literature			O Supplier			O Other users	
		Very High		High	Medium		Low		Very Low
1.1.2	Sale of Software	O		O	O		O		O
1.1.3	Reputation of supplier	O		O	O		O		O

3.1.2 Coding Aspects:

The possibility of additional coding might be a very important feature of a package. This feature determines the flexibility and robustness of the software, which is especially valuable when complex systems are to be modelled. Criteria included in this group determine compilation efficiency, the programming concepts supported, logic builder availability etc.

Table 2: Items for Coding Aspects

		Very High	High	Medium	Low	Very Low
1.2.1	Quality of the support for programming	O	O	O	O	O
1.2.2	Efficiency of Compilation	O	O	O	O	O
		Very Good	Good	Average	Poor	Very Poor
1.2.3	Built-in logic builder	O	O	O	O	O
1.2.4	Program Generator	O	O	O	O	O
1.2.5	Snippet code help	O	O	O	O	O
		Very Large	Large	Medium	Small	Very Small
1.2.6	Built-in functions	O	O	O	O	O
		Very Easy	Easy	Moderate	Tough	Very Tough
1.2.7	Ease of entering text/code	O	O	O	O	O
		Possible				Not Possible
1.2.8	User defined functions	O				O
1.2.9	Writing comments/notes in model building activity	O				O
1.2.10	Creation of macros and arrays	O				O
		Provided				Not Provided
1.2.11	Global variables	O				O
1.2.12	Interface to user written programs	O				O

3.1.3 Software Compatibility:

These criteria evaluate whether the package can be interfaced to other software systems, in order to exchange data with these systems. This feature can considerably enhance the capabilities of the package, especially when complex real systems are modelled.

Table 3: Items for Software Compatibility

1.3.1	Integration with spreadsheet packages	O Excel	O Lotus	O Other _____
1.3.2	Integration with statistical packages (curve-fitting tools)	O SPSS	O Stat Fit	O Other _____
1.3.3	Integration with computer-aided software	O AutoCAD		O Other _____
1.3.4	Integration with database management systems	OSAP	O Oracle	O Other _____
1.3.5	Integration with manufacturing requirements planning software	O Possible		O Not Possible
1.3.6	Is it possible to do broad level scheduling with Simulation S/W	O Yes		O No

3.1.4 User Support:

These criteria evaluate the type and quality of user support provided by the software supplier, which can facilitate learning and using the package. These criteria not only include technical support in the form of documentation, and demo disks, but also include a variety of services provided by the software supplier which ease the use of the package and keep the user informed about plans for future software improvements.

3.1.5 Financial and Technical Features:

Criteria included in this group examine features of the package related to its costs and technical characteristics. Some of the issues considered here are: how expensive is it to purchase a certain package, to install and maintain it, whether any additional hardware would have to be purchased for installation of the package, etc.

Table 4: Items for User Support

		Very Good	Good	Average	Poor	Very Poor
1.4.1	Quality of manuals	O	O	O	O	O
1.4.2	Tutorial	O	O	O	O	O
1.4.3	Run-time help	O	O	O	O	O
1.4.4	Software maintenance facility	O	O	O	O	O
1.4.5	Training course	O	O	O	O	O
1.4.6	Web based support	O	O	O	O	O
1.4.7	Troubleshooting facility	O	O	O	O	O
1.4.8	Quality of documentation	O	O	O	O	O
1.4.9	Demo models	O	O	O	O	O
		Very Frequent	Frequent	Average	Rare	Very Rare
1.4.10	User group meetings	O	O	O	O	O
1.4.11	Frequency of training courses	O	O	O	O	O
		Provided				Not Provided
1.4.12	Discussion groups on the internet	O				O
1.4.13	User community web page	O				O

Table 5: Items for Financial and Technical Features

		Very Large	Large	Medium	Small	Very Small
1.5.1	Types of contracts available	O	O	O	O	O
		Very Frequent	Frequent	Average	Rare	Very Rare
1.5.2	Frequency of update	O	O	O	O	O
1.5.3	Frequency of upgrade	O	O	O	O	O
		Very High	High	Medium	Low	Very Low
1.5.4	Life cycle maintenance costs	O	O	O	O	O
1.5.5	Price of training course	O	O	O	O	O
1.5.6	Comprehensiveness of update	O	O	O	O	O
1.5.7	Price of Software	O	O	O	O	O
1.5.8	Installation costs	O	O	O	O	O
1.5.9	Cost of Hardware required	O	O	O	O	O
		Very Good	Good	Average	Poor	Very Poor
1.5.10	Free technical support	O	O	O	O	O
1.5.11	Availability of free evaluation S/W	O	O	O	O	O
1.5.12	Hierarchical modeling capability (Model/Submodel Merge feature)	O	O	O	O	O
		Very Easy	Easy	Moderate	Tough	Very Tough
1.5.13	Ease of installation	O	O	O	O	O
1.5.14	Ease of modeling	O	O	O	O	O
		Possible				Not possible
1.5.15	Simulation of material handling systems (AGV etc.)	O				O
1.5.16	Model package creation	O				O
1.5.17	Is it possible to use software in a network	O				O
		Provided				Not provided
1.5.18	Availability of package on standard operating systems	O				O
1.5.19	Add-on facility	O				O
1.5.20	Fly-through mode	O				O

3.2. Criteria for Modelling Capabilities

3.2.1 General features:

Criteria included in this group describe general features of the package. Most of these criteria relate to modelling aspects such as the type of formal logic needed for modelling (if any), the method of changing the state of the model (process based, activity based, event

based, three phase, or a combination of these methods), type of simulation (discrete event, continuous or combined), the level of modelling transparency, etc. There are also some criteria that evaluate the level of experience and formal education in simulation required by the user, and examines how easy it is to learn and use the package.

Table 6: Items for General Features

2.1.1	Type of simulation	○ Discrete event		○ Continuous		○ Both
2.1.2	Purpose	○ General purpose		○ Manufacturing Oriented		○ Other
		Very High	High	Medium	Low	Very Low
2.1.3	Representativeness of models	○	○	○	○	○
2.1.4	User friendliness	○	○	○	○	○
2.1.5	Experience required for software use	○	○	○	○	○
2.1.6	Formal education in simulation required for software use	○	○	○	○	○
		Very Easy	Easy	Moderate	Tough	Very Tough
2.1.7	Ease of learning	○	○	○	○	○
2.1.8	Ease of using	○	○	○	○	○
		Very Good	Good	Average	Poor	Very Poor
2.1.9	Run-time interface capability for scenario creation	○	○	○	○	○
2.1.10	Conceptual model generator	○	○	○	○	○
2.1.11	Multiple branch decision making	○	○	○	○	○
2.1.12	Probabilistic branch decision making	○	○	○	○	○
		Possible				Not Possible
2.1.13	Distributed simulation on network environment	○				○
2.1.14	Cut, copy, paste of objects	○				○
2.1.15	Possibility to built near Real-time simulation models	○				○
		Provided				Not Provided
2.1.16	Easy to use templates	○				○
2.1.17	Customizable window environment	○				○
2.1.18	Splines, Polygon and orthogonal curve types	○				○

3.2.2 Modelling Assistance:

Criteria systematised in this group evaluate the type and level of assistance provided by the package during modelling. For example, these criteria examine the comprehensiveness of prompting, on-line help if it is provided, whether the package enables modular model development and writing the documentation notes (this feature enables the writing of documentation concurrently with the model development), and whether the model and data can be separated.

Table 7: Items for Modelling Assistance

		Very Good	Good	Average	Poor	Very Poor
2.2.1	Libraries and templates of simulation objects	○	○	○	○	○
2.2.2	Warning messages	○	○	○	○	○
2.2.3	Intelligent Prompting	○	○	○	○	○
2.2.4	Facility for designing reusable user defined elements	○	○	○	○	○
2.2.5	3D models library	○	○	○	○	○
2.2.6	Bubble help	○	○	○	○	○
2.2.7	Context sensitive prompt to facilitate model development	○	○	○	○	○
		Provided				Not Provided
2.2.8	Undo/redo commands	○				○
2.2.9	Facility to insert comments	○				○

3.3. Criteria for Simulation Capabilities

3.3.1 Visual Aspects:

Graphical presentations of simulation models and animation of simulation are very important characteristics of simulation software. Criteria included in this group relate to the type and quality of graphical facilities provided by the package. These criteria evaluate, for example, whether it is possible to perform an animation of the simulation experiments, the types of animation provided by the package, and whether it is possible to manipulate icons.

Table 8: Items for Visual Aspects

		Very Good	Good	Average	Poor	Very Poor
3.1.1	Shape libraries	O	O	O	O	O
3.1.2	3D-animator	O	O	O	O	O
3.1.3	Logical animation	O	O	O	O	O
3.1.4	Network animation	O	O	O	O	O
3.1.5	Scenario viewer	O	O	O	O	O
3.1.6	Antialias display	O	O	O	O	O
3.1.7	Dashboard facility	O	O	O	O	O
3.1.8	Customizable entity appearance	O	O	O	O	O
3.1.9	Customizable path appearance	O	O	O	O	O
3.1.10	Library for real-time simulations	O	O	O	O	O
3.1.11	Virtual reality animation	O	O	O	O	O
		Provided				Not provided
3.1.12	HotSpot Evaluator	O				O
3.1.13	Flowcharting Module	O				O
3.1.14	Animation of image changes	O				O
3.1.15	Facility for customizing the view of the model	O				O
3.1.16	Playback mode	O				O
		Provided				Not provided
3.1.17	Animation with visual clock	O				O
3.1.18	Zoom function	O				O
3.1.19	Panning	O				O
3.1.20	Print screen facility	O				O

Table 9: Items for Efficiency

		Very High	High	Medium	Low	Very Low
3.2.1	Robustness	O	O	O	O	O
3.2.2	Level of detail	O	O	O	O	O
3.2.3	Adaptability to model changes	O	O	O	O	O
3.2.4	Reliability	O	O	O	O	O
		Very Large	Large	Medium	Small	Very Small
3.2.5	Number of elements in the model	O	O	O	O	O
3.2.6	Number of queuing policies	O	O	O	O	O
3.2.7	Time scale for model building	O	O	O	O	O
3.2.8	Model execution time	O	O	O	O	O
		Very Good	Good	Average	Poor	Very Poor
3.2.9	Model Protection	O	O	O	O	O
		Possible				Not Possible
3.2.10	Model status saving	O				O
3.2.11	Multitasking	O				O
3.2.12	Model chaining (i.e. linking outputs from different models)	O				O
3.2.13	Editing partially developed models	O				O
3.2.14	Interactive handling of parameters during experimentation	O				O
3.2.15	Model reusability	O				O
		Provided				Not provided
3.2.16	Variable watches	O				O
3.2.17	Activity based costing	O				O

3.3.2 Efficiency:

Criteria classified in this group determine the effectiveness and the power of simulation software. Efficiency is expressed both by the capability of the software to model a variety of complex systems and by the characteristics which can save time needed for modelling, and improve the quality of modelling, such as model reusability, reliability, compilation and execution time and multitasking.

3.3.3 Testability:

This group comprises criteria that examine which facilities for model verification are provided by the package. These facilities include error messages, displays of the values of logical elements such as functions and variables, the possibility of obtaining special files for verification such as list, trace and echo files, provision of step function, etc.

Table 10: Items for Testability

3.3.1	Moment of error diagnosis	O Model entry		O Compilation		O Execution
		Possible				Not Possible
3.3.2	Display of attributes	O				O
3.3.3	Display of variables	O				O
3.3.4	Display of element's state	O				O
3.3.5	Replication of Run-length	O				O
3.3.6	Change in simulation speed	O				O
3.3.7	Execution trace	O				O
3.3.8	Logic checks	O				O
3.3.9	Runtime error viewer	O				O
		Possible				Not Possible
3.3.10	Explode function (showing a state of an element)	O				O
3.3.11	List of used elements	O				O
3.3.12	Backward clock	O				O
3.3.13	Step function (event to event jumping)	O				O
3.3.14	Display of parts flow tracking record collected during simulation run	O				O
3.3.15	Audible alarms	O				O
3.3.16	Rejection of illegal inputs	O				O
3.3.17	Syntax checker	O				O
3.3.18	Search & replace capability	O				O
3.3.19	Antithetic numbers	O				O
3.3.20	Multiple windows during simulation run	O				O
		Provided				Not provided
3.3.21	User Pause facility	O				O
3.3.22	OLE compatibility	O				O
		Very Good	Good	Average	Poor	Very Poor
3.3.23	Display of events on the screen	O	O	O	O	O
3.3.24	Display of the workflow path	O	O	O	O	O
3.3.25	Flow analysis	O	O	O	O	O
3.3.26	Interactive debugger	O	O	O	O	O
3.3.27	Line by line debugging	O	O	O	O	O
3.3.28	Interaction with model while running	O	O	O	O	O

3.3.4 Experimentation facilities:

Criteria classified in this group evaluate the variety and characteristics of experimentation facilities. These facilities are required for improving the quality of simulation results and for speeding up the process of designing experiments and of the experimentation itself.

Table 11: Items for Experimentation facilities

		Very Good	Good	Average	Poor	Very Poor
3.4.1	Quality of experimental design facility	O	O	O	O	O
		Very High	High	Medium	Low	Very Low
3.4.2	Warm-up period	O	O	O	O	O
		Possible				Not Possible
3.4.3	Automatic batch run	O				O
3.4.4	Restart from non-empty state	O				O
3.4.5	Stepwise simulation run	O				O
3.4.6	Resource variability	O				O
		Provided				Not provided
3.4.7	Independent replications of experiments for multiple runs	O				O
3.4.8	Breakpoints	O				O
3.4.9	Accuracy check	O				O
3.4.10	Automatic determination of run length	O				O
3.4.11	Shift editor	O				O
3.4.12	Scheduled execution of scripts	O				O
3.4.13	Sensitivity analysis	O				O

3.3.5 Statistical facilities:

Owing to the randomness that is present in the majority of simulation models, good statistical facilities are very important. Criteria included in this group examine the range and quality of statistical facilities provided by the simulation package.

Table 12: Items for Statistical facilities

		Very High	High	Medium	Low	Very Low
3.5.1	Quality of data analysis facility	O	O	O	O	O
		Very Large	Large	Medium	Small	Very Small
3.5.2	Number of theoretical statistical distributions	O	O	O	O	O
3.5.3	Number of different random number streams	O	O	O	O	O
		Provided				Not provided
3.5.4	Time dependent distributions	O				O
3.5.5	Ability to specify the random number seed	O				O
3.5.6	Random number generation by probability distributions	O				O
3.5.7	Distribution fitting	O				O
3.5.8	Goodness-of-fit tests	O				O
3.5.9	Output data analysis	O				O

4. CRITERIA FOR INPUT/OUTPUT ISSUES

4.1 Input and Output Capabilities:

Criteria included in this group investigate how the user can present the data to the package and the type and quality of output reports provided by the package. These criteria evaluate,

for example, whether the package has a menu-driven interface, whether static and dynamic output reports are provided, and how understandable these reports are.

Table 13: Items for Input/Output Capabilities

		Very Good	Good	Average	Poor	Very Poor
4.1.1	Static graphical output	O	O	O	O	O
4.1.2	Dynamic graphical output	O	O	O	O	O
4.1.3	Snapshot reports	O	O	O	O	O
4.1.4	Database maintenance for input/output	O	O	O	O	O
4.1.5	Dialogue boxes	O	O	O	O	O
4.1.6	Data Charting	O	O	O	O	O
4.1.7	Custom report generation	O	O	O	O	O
		Very High	High	Medium	Low	Very Low
4.1.8	Quality of output reports	O	O	O	O	O
4.1.9	Understandability of output reports	O	O	O	O	O
		Possible				Not Possible
4.1.10	Multiple inputs	O				O
4.1.11	Multiple outputs	O				O
4.1.12	Output export to excel	O				O
4.1.13	Printed report after each simulation run	O				O
4.1.14	Exchange data via internet	O				O
4.1.15	Task timeline report	O				O
4.1.16	Task execution report	O				O
4.1.17	Queue data collection report	O				O
		Provided				Not provided
4.1.18	Automatic rescaling of histograms and time series	O				O
4.1.19	Periodic output of simulation results	O				O
4.1.20	Writing reports to files	O				O
4.1.21	Summary reports for multiple run	O				O
4.1.22	Formattable result summary	O				O

4.2 Analysis Capabilities:

Table 14: Items for Analysis Capabilities

4.2.1	Capability to do What-if Analysis	O Yes	O No
4.2.2	Conclusion-making support	O Provided	O Not provided
4.2.3	Optimization	O Provided	O Not provided

5. METHODOLOGY FOR EVALUATION & SELECTION OF SIMULATION SOFTWARE

The methodology consists of the following six stages to be followed in the process of simulation software evaluation and selection:

Stage 1: Need for Purchasing Simulation Software

This stage represents the beginning of the software selection process. Once the need for purchasing simulation software has been established, several factors have to be considered. These factors include the intended simulation purpose, the existing constraints within the company, the main types of problems to be simulated, and information regarding the modellers and potential users. Other issues to be considered are organisational constraints. Financial constraints might include the hardware available for use by the simulation software, and the budget available for software purchasing, installation and maintenance costs, purchasing additional hardware, training of personnel, etc. Another constraint is the

time available for software evaluation, selection and implementation. A determination of the types of models that are likely to be developed can further help reduce the list of possible software candidates for evaluation. Three main types of model can be distinguished: discrete-event, continuous, or those that combine discrete and continuous elements. The issues relating to the persons involved in software selection, modelling, and the use of future models must also be addressed. Preferably, the same employees should be involved in the process of software selection and modelling.

Stage 2: Initial Software Survey

Following a determination of the above preliminary elements in Stage 1, an initial software survey has to be undertaken. The purpose of this is to shorten the list of software products that can be considered for evaluation and subsequent selection. If the initial elements are determined, the range of choice should already be narrowed. For example, if the systems to be simulated comprise both discrete and continuous elements, then all packages that are not suitable (i.e. cannot simulate both types of element) can be eliminated. At this stage, several other sources of information have to be consulted. Vendors of software products that seem to be candidates for software evaluation should be contacted and asked for assistance. They should provide as much information as possible, in addition to software demonstrations and written material. Other sources of literature related to software being considered ought to be examined and other software users contacted, if possible. The final decision for choosing software for evaluation should be influenced by the willingness of vendors to provide software for a free trial (at least a simplified version of the software) and appropriate documentation. Documentation should be well organised, indexed and written for an average non-technical user. The outcome of this stage is a short list of simulation software for evaluation.

Stage 3: Evaluation

Once it is decided which software products are to be evaluated and selected from Stage 2, the actual process of evaluation is performed. The main elements for this process are the evaluation framework presented in this paper, software to be evaluated, and documentation. Evaluation criteria must be chosen and the appropriate hierarchy determined for these criteria. The first step is to establish the most important software features according to their intended purpose, and after that investigate additional features within groups of criteria, according to available time and preferences. It is advisable to develop a preliminary model that is typical for the intended software purpose and type of system to be modelled. This practical work is necessary for evaluation, because it gives an impression about the software and actually tests software facilities (it is not impossible that certain features work on paper but not in practice). At a certain stage of evaluation, after some notes have been made and it is clear which features are important and which additional information is needed, it might be useful to meet a vendor representative. This ought to enable a discussion of present and forthcoming features of the particular software to take place with someone who has an adequate level of technical expertise.

Stage 4: Software Selection

The actions described in the previous three stages should lead to credible software evaluation results. On the basis of these results, it should be decided which software seems to be the most appropriate and which are suitable alternatives (if there are any). In an ideal situation, it should be possible to purchase more than one software product for a particular purpose. In this case, software tools should be chosen on the basis of evaluation results.

Stage 5: Negotiating Software Contract

Following software selection in Stage 4, the next step is to negotiate a software contract acceptable to both parties. The contract should specify what products and services are provided, where and when they may be used, how the licence may be transferred to other parties, and how long the product may be used. Dates and obligations should be specified precisely to avoid any future misunderstandings.

Stage 6. Software Purchase

In this stage, software is purchased and implemented if an acceptable contract is acquired in Stage 5. Where a suitable agreement cannot be achieved, nor an adequate level of support secured, the subsequent best alternative should be examined on the basis of evaluation results. This requires a return to Stage 4 for the selection of alternative software products, then proceeding to Stage 5 for the negotiation of another software contract, and finally to Stage 6 for software purchasing after the acceptable agreement has been achieved with the software supplier.

6. CONCLUDING COMMENTS

This paper provides a number of criteria derived for the evaluation of simulation software. More than 205 evaluation criteria are available within several groups of criteria. It would not be realistic to expect a particular package to satisfy all criteria. Nevertheless, some indication of which criteria are more important, according to the software purpose, is given. For each particular case of evaluation, it has to be decided which particular software features are more important than others. For this purpose, many other factors should be considered prior to software evaluation, such as financial and organisational constraints, individual preferences and experience of potential software users, types of systems that are to be simulated, etc. The evaluation framework and simulation software selection methodology presented in this paper can be used by anybody wishing to evaluate and choose adequate simulation software. This approach can improve the chance of effective simulation software selection and implementation. A specification of the initial factors to be considered once a need for purchasing simulation software has been established forms a basis for the initial software survey. This should enable instant elimination of many software products that would not be suitable for a particular circumstance, and subsequently result in a short list of software for evaluation. An overview of the desirable characteristics of simulation packages provided in this paper is more comprehensive than those reviews found in the literature. Therefore, these guidelines can be used both by users who are looking for a suitable simulator to buy, and by developers of such simulators to improve existing versions of simulators or perhaps to develop a new, better simulation package.

Acknowledgements

Much of the information in this paper was derived from extended conversations with software vendor personnel. The authors gratefully acknowledge the time investments in this project that were generously provided by Mr. Jai Shankar from ProModel Corporation, Mr. Mano from Flexsim Software Solutions., Mr. JC Rupe from CACI Inc., Mr. Vikram Ramnath and Mr. Jerry Banks from AutoMod Software, Mr. Beth Plott from Alion Science and Technology and Mr. Karen Raeburn from SIMUL8 Corporation.

References

[1] W.G. Wild and P. Otis, This video game is saving manufacturing millions, Business Week Vol. 20(4), pp. 82–84.

[2] J.W. Rice, Simulation in manufacturing, Proceedings of APICS, 31st Annual Conference, Las Vegas, USA, pp. 410–413.

[3] S. Wilkinson, Simulation gives clear picture of assembly line, PC Week, Vol. 28(8), pp. 36-40.

[4] S.W. Haider and J. Banks, Simulation software products for analyzing manufacturing system, Industrial Engineer, Vol. 12, pp. 98–103.

[5] M. Bovone, D.V. Ferrari and R. Manuelli, How to choose an useful simulation software, Proceedings of the 1989 European Simulation Multiconference, SCS, San Diego, pp. 39-43.

[6] N.N. Ekere and R.G. Hannam, An evaluation of approaches to modelling and simulating manufacturing systems, International Journal of Production Research, Vol. 27(4), pp. 599-611.

[7] A.M. Law and S.W. Haider, Selecting simulation software for manufacturing applications: Practical guidelines and software survey, Industrial Engineer, Vol. 34, pp. 33-46.

[8] A.K. Kochhar and X. Ma, Discrete event simulation software tools for the simulation of advanced manufacturing systems, Proceedings of the 1989 European Simulation Conference, Italy, San Diego, pp. 13–18.

[9] A.V. Breedam, J. Raes and K.V.D. Velde, Segmenting the simulation software market, OR Insight, Vol. 3 (2), pp. 9-12.

[10] K. Holder, Selecting simulation software, OR Insight, Vol. 3(4), pp. 19-24.

[11] J. Banks, E. Aviles, J.R. McLaughlin and R.C. Yuan, The simulator: new member of the simulation family, Interfaces, Vol. 21 (2), pp. 76-86.

[12] G.B. Williams and J. Trauth, Evaluation and selection of manufacturing software for a small business, Proceedings of The Seventh National Conference on Production Research, Hatfield, UK, pp. 102-106.

[13] A.M. Law and W.D. Kelton, Simulation Modelling and Analysis, 2nd ed., McGraw-Hill, Singapore.

[14] G.T. Mackulak and P.A. Savory, Ascertaining important features for industrial simulation environments, Simulation, Vol. 11, pp. 211–221.

[15] J. Nikoukaran, V. Hlupic and R.J. Paul, Criteria for Simulation Software Evaluation, Proceedings of the 30th Conference on Winter Simulation, Washington, D.C., United States, pp. 399 – 406.

[16] V. Hlupic and R.J. Paul, Guidelines for selection of manufacturing simulation software. IIE Transactions, Vol. 31, pp. 21-29.

[17] T.W. Tewoldeberhan, A.Verbraeck, E. Valentin and G. Bardonnet, An Evaluation and Selection Methodology for Discrete-Event Simulation Software, Proceedings of the 2002 Winter Simulation Conference, Vol. 1, pp. 67- 75.

[18] V. Hlupic and A.S. Mann, A System for Simulation Software Selection, Proceedings of 2002 Winter Simulation Conference, California, pp. 720-727.

[19] V. Hlupic and R.J. Paul, Guidelines for selection of manufacturing simulation software, IIE Transactions, Vol. 31, pp. 21-29.

[20] A.F. Seila, V. Ceric and P. Tadikamalla, Applied Simulation Modeling, Thomson Learning, Australia.