

Discrete-Event Simulation: From the Pioneers to the Present, What Next?

Author(s): S. Robinson

Source: *The Journal of the Operational Research Society*, Jun., 2005, Vol. 56, No. 6 (Jun., 2005), pp. 619-629

Published by: Palgrave Macmillan Journals on behalf of the Operational Research Society

Stable URL: <https://www.jstor.org/stable/4102035>

REFERENCES

Linked references are available on JSTOR for this article:

https://www.jstor.org/stable/4102035?seq=1&cid=pdf-reference#references_tab_contents

You may need to log in to JSTOR to access the linked references.

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at <https://about.jstor.org/terms>



JSTOR

Operational Research Society and *Palgrave Macmillan Journals* are collaborating with JSTOR to digitize, preserve and extend access to *The Journal of the Operational Research Society*



Discrete-event simulation: from the pioneers to the present, what next?

S Robinson*

University of Warwick, Coventry, UK

Discrete-event simulation is one of the most popular modelling techniques. It has developed significantly since the inception of computer simulation in the 1950s, most of this in line with developments in computing. The progress of simulation from its early days is charted with a particular focus on recent history. Specific developments in the past 15 years include visual interactive modelling, simulation optimization, virtual reality, integration with other software, simulation in the service sector, distributed simulation and the use of the worldwide web. The future is then speculated upon. Potential changes in model development, model use, the domain of application for simulation and integration with other simulation approaches are all discussed. The desirability of continuing to follow developments in computing, without significant developments in the wider methodology of simulation, is questioned.

Journal of the Operational Research Society (2005) 56, 619–629. doi:10.1057/palgrave.jors.2601864

Published online 22 September 2004

Keywords: discrete-event simulation; computing

Introduction

Discrete-event simulation is one of the most commonly used modelling techniques.^{1–3} Since the earliest days of computer simulation in the 1950s, the proliferation of simulation software and continued developments in computing have helped to place it near the top of the ‘modelling chart’. The past 50 years have seen many changes in the ways that simulations are developed and used, some brought on by developments within the field and others by external events. So what is the current state of discrete-event simulation? And what of the future, what will happen next in the simulation world?

The discussion that follows aims to answer these questions by providing a review of the current state and possible future for discrete-event simulation. This is done in three stages. First, there is a brief outline of the history of simulation from its inception to the end of the 1980s. Brief, because this history is well documented elsewhere. Following this, the current state of simulation is described by charting its progress through the 1990s to the present day. There is also a brief interlude in which the state of simulation research in the United Kingdom is discussed. The third part of the discussion centres on the future of simulation. Various observations are made about the possible ways in which simulations will be developed and used in the years to come.

All reviews are, of course, coloured by the world-view from which they are written. Therefore, before launching into a review of the history, current state and potential future

of simulation, the perspective from which this review is written should be clarified. The readers are then in a position to form their own opinions about the validity, or otherwise, of the discussion in relation to their own world-views.

This paper is written primarily in relation to discrete-event simulation (described as ‘simulation’ for convenience) as it is applied to the modelling of operations systems, that is, ‘a configuration of resources combined for the provision of goods or services’.⁴ Wild⁴ identifies four specific functions of operations systems: manufacture, transport, supply and service. There are, of course, many examples of such systems in business, the public sector and military organizations. Beyond this, because the author is based in the United Kingdom there is obviously a natural bias towards developments as seen from this geographical region. This is not to say that developments elsewhere in the world are ignored. Finally, the author’s prime interest is in the practice and application of simulation rather than in computing or statistical analysis. Again, this does not mean that developments in computing and statistical analysis have been excluded from the discussion that follows.

It is now left to the readers to determine how they would place themselves in relation to this world-view. This is particularly important once we start to debate the future where individual perspectives might influence our views most strongly.

The history of simulation: pioneering to revolution (1950–1980s)

In describing the history of simulation four distinct periods can be identified, referred to here as the periods of

*Correspondence: S Robinson, Warwick Business School, University of Warwick, Coventry, CV4 7AL, UK.
E-mail: stewart.robinson@warwick.ac.uk

pioneering, innovation, revolution and evolution. Each period roughly equates to a single decade, starting from the late 1950s. Since the history of simulation through to the end of the 1980s is well documented elsewhere, only a brief summary of these periods is given here, with a more detailed account of the story from 1990 on. For detailed accounts of the earlier history of simulation, see Pidd,⁵ Bell and O'Keefe,⁶ Paul,⁷ Bell⁸ and Hollocks.⁹ For an American perspective see Araten *et al.*,¹⁰ Banks *et al.*,¹¹ and Nance and Sargent.¹²

A key feature of the development of simulation is that it has closely followed the development of computing.⁵ As hardware and software have improved, so too have the capabilities of computer simulation. Indeed, this can be said right through to the end of the 1990s. However, as is explained later, more recently there appears to have been some departure from this conjoined path.

The pioneering period (1950–1960s)

In the late 1950s and the 1960s, pioneers in the field of simulation laid the foundations for future developments. Seizing upon the opportunity afforded by the first generation of computers, the simulations of the 1950s were developed in machine code. Meanwhile, the introduction of programming languages and more powerful and reliable computers in the 1960s greatly enhanced the potential for developing simulations. Advances were made in simulation methodology, for instance, Tocher¹³ published the three-phase simulation approach that is still used by a number of simulation packages today. The 1960s also saw the development of the first specialist simulation software, for example, GPSS¹⁴ around 1961 and SIMSCRIPT.¹⁵ SIMULA,¹⁶ developed in Norway, was a forerunner to modern object-oriented programming languages with its use of classes and inheritance. There is also evidence of the early use of animation.¹⁷

The period of innovation (1970s)

The 1970s represent a period of continued development and innovation. Computing technology continued to advance and with it simulation software continued to improve. Various new languages were developed, for instance, SLAM¹⁸ and GPSS-H.¹⁹ The late 1970s saw the introduction of the first microcomputers. Further to this, Hurriion²⁰ published his PhD thesis describing the potential for visual interactive simulation (VIS). Up to this point, simulations had provided little in the way of animation and interactive facilities. By the end of the 1970s, the bedrock had been laid for the 'revolution' that would take place in the 1980s.

The revolution (1980s)

Up until the 1980s, the commercial application of simulation was limited by the need for expensive hardware and

specialist computing skills. These were not readily available to most organizations. Two things were about to change all this. Firstly, reasonably powerful microcomputers started to become common place in organizations, not least because of IBM's entry into the market. Secondly, the development of VIS software, starting with SEE-WHY²¹ in 1979. Although VIS itself did not make model development much easier, it did enable the clients to become more involved in the modelling process and to use the models themselves for experimentation if that was required.

By the end of the 1980s, powerful microcomputers were available in most organisations (and even in the home) and many commercial VIS packages were available such as WITNESS,²² HOCUS,²³ GENETIK,²⁴ SIMAN/CINEMA^{25,26} and ProModel.²⁷ Many organizations, particularly in the manufacturing sector, were using simulation as a decision-aiding tool, although the take-up continued to be patchy and adoption was not widespread.²⁸

The history of simulation: evolution (1990s to the present)

This brings us to the early 1990s. In computing terms the decade that followed was typified by the ever-increasing power of the personal computer (PC), the falling price of the PC, the use of Windows technology and, of course, the worldwide web. Simulation certainly took advantage of all these, with the possible exception of the worldwide web as is discussed below. More powerful PCs enabled larger scale models to be developed and then to be run in a reasonable time-frame. The falling price of the PC made them ever more prevalent in both work and domestic life. Windows technology improved ease of use and created a common interface for computer users. The increase in computing power can be put in perspective when the speed of modern computers is compared to Bell's⁸ upbeat assessment in 1991 in which he states that '20 + MHz clock speeds will allow ... models to be executed at reasonable rates'!

After the revolution of the 1980s, in simulation terms the 1990s onwards represent a period of continued evolution, steady but not dramatic change. Key areas of change were in the development of visual interactive modelling, simulation optimization, virtual reality, software integration and simulation in the service sector. As before, these changes owe much to the development of computing. The one area where simulation has moved little beyond the desks of researchers is in its use of the worldwide web and distributed computing. All these areas are discussed in more detail below.

Visual interactive modelling

Although by the end of the 1980s, commercial software supported VIS, model development was still heavily reliant on the use of programming constructs. Towards the end of

the 1980s, WITNESS²² and ProModel²⁷ were among the first commercial simulation packages to break this mould with their use of a menu driven interface for model development. Many packages followed including (in a rough order of their introduction), Arena,²⁹ QUEST,³⁰ Taylor II,³¹ AutoMod,³² AweSim,³³ Micro Saint,³⁴ Enterprise Dynamics³⁵ and Flexsim.³⁶ Pidd³⁷ refers to these packages as visual interactive modelling systems (VIMS), recognizing that not only could a user interact with a visual simulation, but that the process of model development could also be carried out in a visual and interactive manner. As a result of the introduction of these packages, the development and use of a simulation model was no longer so reliant on specialist computing skills, although this does not mean that specialist modelling skills are not required. Because simulations involve an (simplified) imitation of a complex world, simple menu driven constructs are rarely sufficient to embody the level of detail required. As a result, powerful VIMS continue to include some form of programming interface and so it cannot be claimed that specialist computing skills can be completely dispensed with.

The early VIMS were certainly not cheap and today many of these packages continue to sell for more than £10,000. The price is often justified on the basis of the benefits that can accrue to an organization through the use of simulation. Albeit a reasonable argument, the requirement for a significant up-front investment may prove prohibitive for some organisations. Since the mid-1990s, however, a number of 'low cost' VIMS packages have become available, for instance, Simul8,³⁸ Extend³⁹ and more recently ShowFlow,⁴⁰ which is a re-launch of the Taylor II software. These are all sold for less than £1000.

These packages have met with some success and have certainly widened access to simulation. Interestingly, the emergence of these cheaper packages does not seem to have significantly eroded the market for the more expensive VIMS, although there is some evidence of price reductions and of consolidation through mergers and buy outs of software vendors. There are two possible explanations for this. First, the two types of software are working in quite different markets, the more expensive software claiming to be able to model much more complex problems and/or to provide a higher level of support. Second, the presence of 'low cost' VIMS has heightened the awareness of simulation, causing the total market for simulation to expand, benefiting all the software providers. Indeed, there is anecdotal evidence of organizations that have started using 'low cost' VIMS and have later moved to a more expensive package as their requirements change.

Simulation optimization

Law and McComas⁴¹ describe simulation optimization as 'arguably ... the most significant new simulation technology'. Certainly, in the past 10 years the majority of

simulation software packages have included some form of optimization facility. The argument may come over the extent to which the ready availability of optimization capabilities has changed simulation practice.

The problem is similar to that for any mathematical optimization problem. Given an objective function (expressed in terms of the simulation outputs), the aim is to find the optimum value of some decision variables (simulation model inputs), subject to a set of constraints (allowable range for the simulation model inputs). The difference from mathematical optimization is that no algorithms exist that guarantee an optimal solution will be found.

Simulation optimization has long been of interest within the research community and various approaches have been investigated over the years.^{42–44} One such approach is based on response surface methodology or metamodeling, in which the relationship between the simulation model inputs and outputs is represented, at least approximately. Regression is one such method for representing the relationship between the inputs (independent variables) and outputs (dependent variables). Alternatively, neural networks have been used.⁴⁵ Because the metamodel can be run almost instantaneously, a search of the response surface can be performed much more rapidly than with a simulation model, giving a greater probability of finding the optimum.

An alternative approach is to use metaheuristic search strategies, such as simulated annealing, genetic algorithms and tabu search. The 'optimizer' performs a series of simulation runs and based on the results decides what combination of model inputs to use for the next run. The simplest approach would be to adopt a hill-climbing algorithm, but this, of course, could lead to a local rather than a global optimum being found. The metaheuristic algorithms seek to overcome this problem by performing a more sophisticated search. The limitation is that because the simulation needs to be run for each input combination these methods are extremely time consuming, while they do not guarantee that an optimum solution will be found. There is also the problem that the output of most simulations is stochastic and so exact point predictions of performance cannot be given. The benefit of these algorithms is that they enable an intelligent search of the solution space to be performed without the intervention of the simulation user. Indeed, Robinson⁴⁶ argues that the term 'searchization' rather than 'optimization' more closely describes the function of these optimizers.

The majority of simulation software vendors seem to have adopted metaheuristic approaches, for example, WITNESS Optimizer,⁴⁷ SimRunner for ProModel⁴⁸ and AutoStat for AutoMod.⁴⁹ Also based on metaheuristics, the OptQuest software⁵⁰ has been adopted as the optimisation engine for a number of packages including Arena, Enterprise Dynamics and Simul8. Although these optimizers are readily available, anecdotal evidence and case studies do not suggest that their use is widespread, with simulation users seeming to rely on

more manual methods of experimentation. This may be because optimization is something of an anathema to VIS, which emphasizes the benefits of interaction with the model.

Fu⁴³ describes a number of other methods of optimization for simulation. In doing so, he notes a divergence between research and practice in simulation optimization. The software vendors, described above, have adopted general algorithms taken from the realm of deterministic optimization. Meanwhile, research has focused on the stochastic nature of simulations using algorithms that assure an optimum is found efficiently. The limitation of these algorithms is that many are problem specific, making commercial implementation problematic.

Virtual reality

Over the past 10 years the ability to generate 'virtual reality' (VR) animations of the running model have been included with many simulation packages, for instance, AutoMod,⁴⁹ Flexsim³⁶ and WITNESS VR.⁵¹ QUEST³⁰ was one of the first simulation packages to include such capabilities. The adoption of VR is an extension of the visual environment envisaged as part of VIS. It should be noted, however, that although these displays are described as VR, they are generally little more than three-dimensional animations that can be viewed from different perspectives.

There is a question over the benefit of such displays. Waller and Ladbrook⁵¹ admit that VR adds little or nothing to the solution of a problem with simulation, but argue that the benefits lie in the improved understanding of the system being modelled and in better communication, especially with senior management. As long as the VR display can be built quickly then these benefits can be realized at a reasonable cost. Meanwhile, Preddy and Nance⁵² point to the obvious benefits of VR for training simulations.

Software integration

At the beginning of the 1990s, most simulation models were stand alone applications, possibly with some links to files for reading data and recording outputs. There were a few exceptions to this, at least in the realms of research, for example, the linking of simulation models to artificial intelligence systems.^{53,54} Windows technology and more powerful computers have made the integration of simulation models with other software applications much more commonplace. Dynamic data exchange (DDE) and later object linking and embedding (OLE), along with multi-tasking operating systems have made this possible. It is probably true to say that all commercial simulation software packages now have capabilities for the exchange of data between applications.

The most common link made to/from simulation models is with Microsoft Excel. This provides a familiar and user friendly interface for data entry and reporting results. Links

to other software are also made, such as Microsoft Access, process mapping tools and CAD packages. Work continues in the realm of artificial intelligence, particularly for representing human interaction with a real system.⁵⁵⁻⁵⁷

Simulation in the service sector

The traditional base of simulation within the business environment has been in the manufacturing sector. During the 1990s there was a clear increase in usage in the service sector. Models were increasingly being developed of, for instance, airport facilities,⁵⁸ call centres,⁵⁹ business processes,⁶⁰ fast food restaurants⁶¹ and health care clinics.⁶² The business process reengineering 'fad' surely had some effect on the demand for simulation as the need for simulation as part of the re-engineering process was identified.⁶³ The move into service simulation no doubt mirrors the general growth in the service sector in most western economies over the past decade or so, along with the continued decline in manufacturing.

Simulation on the worldwide web and distributed simulation

The greatest single change in computing over the past 10-15 years has obviously been the growth of the worldwide web. But to what extent has this changed the way in which simulation is carried out? It would seem that there has been only a limited impact on the simulation community beyond that experienced by society as a whole through the use of email, information searching etc.

Some of the earliest discussions on simulation and the worldwide web took place at the 1996 Winter Simulation Conference.⁶⁴ Here the potential for sharing model components, complete models and simulation software was identified. There was also some discussion around distributed users interacting with a model. Two further papers discussed the use of the Java language for developing web-based simulations.^{65,66}

Since that time, researchers have suggested many potential applications for simulation on the worldwide web and closely allied to this the idea of distributed simulation (simulation across more than one computer). The execution of a large model might be split across many computers⁶⁷⁻⁶⁹ or separate models on several computers could be linked and run concurrently.^{70,71} Simulation models might be linked to remote third-party software during execution, such as databases, artificial intelligence software or real-time systems.^{72,73} The speed and range of experimentation could be helped by distributing replications across a number of computers^{74,75} or by distributing different experimental scenarios.^{76,77} Distributed users could interact with a gaming simulation from remote locations.^{78,79} The process of performing a simulation study could be helped by sharing models and software between distributed users.⁸⁰⁻⁸² Virtual

meetings could improve communication during model development and use.^{83,84} Model developers could search for and download model components over the worldwide web, reducing model development time.⁸⁵

Despite a whole range of opportunities and the prevalence of the worldwide web, there seems to have been very little impact on the simulation community, particularly among business users. It is hard to find applications reported in the business context, and simulation software vendors have not provided much support for distributed simulation. AutoMod is one of the few packages to claim some distributed simulation capabilities.⁴⁹ The main use of the world wide web in simulation seems to have come from sharing models and virtual meetings using groupware such as Microsoft's NetMeeting.⁸³ Outside the business context, the military, particularly in the USA, have invested heavily in distributed simulation, with varying degrees of success.

There may be various reasons for this slow uptake of the opportunities afforded by the world wide web. A key reason is that distributed simulation is difficult to implement. A major problem is synchronizing events that are being executed across a number of computers. Distributing multiple replications or experimental scenarios is not so difficult, but current software licensing arrangements for many simulation software packages restrict the potential for running multiple copies of a model. These arrangements also limit the potential for sharing software between multiple users. It may simply be that there is insufficient demand to make it viable for the simulation software vendors to invest in the development of these facilities. In terms of sharing model components, Pidd⁸⁶ points out that even if a demand exists, there is little incentive to encourage the supply of components on the web.

It would seem that with the expansion of the worldwide web, the conjoined path of simulation and computing has somewhat come adrift. At least simulation seems to be lagging some way behind.

Simulation research in the United Kingdom

The recent International Review of Operational Research in the United Kingdom⁸⁷ demonstrated the healthy state of simulation research. Work relating to discrete-event simulation in operational research is being carried out in 11 centres around the United Kingdom. These are Aston University, Bournemouth University, Brunel University, Canterbury Business School, Cranfield University (and the Royal Military College at Shrivenham), Lancaster University, University of East Anglia, University of Paisley, University of Southampton and Warwick Business School. This research covers a range of areas including:

- Simulation model development approaches, for example, distributed simulation, model design methods

- Output analysis, experimental design and optimization, for example, bootstrap methods, warm-up period determination
- Simulation practice, for example, surveys of simulation use
- Human performance modelling, for example, modelling worker performance and human decision-making
- Healthcare modelling, for example, informing policy and evaluating prevention and treatment interventions

There is also a stream of work investigating the links between discrete-event simulation and other fields such as knowledge management,⁸⁸ business processes⁶⁰ and complexity science.⁸⁹

Various networks exist for sharing ideas across the research and practitioner community. GROUPLIM, centred at Brunel University, has been looking into strategies and infrastructure for collaborative simulation modelling. Cranfield University has set-up HPM-net that is focusing on the modelling of human performance. Meanwhile, the High Level Architecture - Commercial-off-the-shelf Simulation Package Interoperability Forum (HLA-CSPIF) is an international standards group, based at Brunel University, that is looking to develop standards for the interoperability of commercial simulation software.

The Operational Research Society's Simulation Study Group has become a focus for the United Kingdom's discrete-event simulation community, holding regular 1-day and half-day meetings. Its biennial conference^{90,91} attracts researchers and practitioners from across the United Kingdom and overseas. The conference has also received official recognition from the Association for Computing Machinery's Special Interest Group for Simulation (ACM SIGSIM). Meanwhile, there is always a strong contingent of United Kingdom representatives at the annual Winter Simulation Conference,⁹² which is the major international event in discrete-event simulation.

What next?

So what will happen next? Or, indeed, what should happen next? Where will the next breakthrough in simulation come from? Or should we expect a continuation of the gradual evolution seen in recent years? It is almost impossible to predict breakthroughs, since they normally come from the most unexpected places. Therefore, it is probably safest to make predictions of an evolutionary nature and then to be pleasantly surprised when a breakthrough occurs.

The thoughts that follow come with the usual disclaimer that all forecasts are wrong. On the other hand 'where there is no vision, the people perish'⁹³ and so it seems right to at least think about what the future might hold, even if when we get there, the future looks quite different. All such discussions are, of course, founded in the opinions of the author, and this one is no exception!

A good place to start in discussing the future of simulation is to think about the nature of the future users of simulation. This might point to changing expectations and direct us towards a response.

The current users of simulation are predominantly from what might be described as the 'Space Invaders' generation (this can be interpreted literally or not depending on the user's exact age!). They are used to, and happy with, fairly simple graphics and low levels of interaction with their simulation models. The next cohort of simulation users will be taken from the 'Sims' generation. They are used to, and expect, a high level of graphics and interaction with their models.⁹⁴ Whether we agree with them or not, those brought up using powerful PCs and games consoles will probably not accept the types of simulations that are offered by current simulation software.

Thinking more on the development of simulation models, the current generation are comfortable with a fairly structured development processes and with writing their own code; code is interpreted here in the widest sense and includes model building using VIMS. The 'Sims' generation have been brought-up expecting to find what they require on the worldwide web, in other words to piece together someone else's 'code'.⁸⁵ What is more, they search in a fairly unstructured manner and expect to obtain that code for free. Surely, the expectations of the next generation are going to affect the way that simulations are both developed and used?

The discussion that follows explores the future of discrete-event simulation from four perspectives: model development, model use, the domain of application for simulation and integration with other simulation approaches. It focuses on a mix of speculation about what might happen and deliberation about what should happen. The ideas aim to take into account the current needs of simulation modellers and users as well as looking ahead to the next generation. The discussion, and the paper, concludes by asking whether the paths of simulation and computing will continue to be conjoined.

Model development

There does not seem to have been a breakthrough in how simulation models are developed since the introduction of VIMS in the late 1980s. Indeed, Henriksen states that 'there are a number of simulation products whose core technologies haven't changed in a decade';⁹⁵ the author would add the words '... or more'. The nearest to a significant change since that time might have been the adoption of the object-oriented approach,⁹⁶ but although there is much lip service to object orientation, it has hardly entered the mainstream of simulation software, at least in its pure form.

Where is a change in the development of simulation models needed? Ease-of-use is something that is constantly expressed as a need. VIMS have gone some way to

addressing this need and are certainly easier to use than their predecessors. Is there another level in ease-of-use that can be reached? Surely as computing technology continues to develop, simulation software will become easier to use. There is, however, a constraint on ease-of-use generated, not by the software, but by the nature of the systems that are being modelled.

Simulations are normally developed because a system is too complex to be represented in any other way. If the systems that are being modelled are complex, the simulations themselves must involve some level of complexity, although at a level of abstraction from the system. It may, therefore, not be the simulation tools that are difficult to use, but the systems that are being modelled that are difficult to represent. In this respect, 'difficulty-in-use' is almost to be welcomed since it means that simulation is succeeding in helping with difficult problems that could not otherwise be addressed.

Meanwhile, simulation model development requires more than just the use of software to code the model. Critical issues lie in the areas of conceptual modelling (deciding what to model) and validation (determining if the model is right). There is almost a complete dearth of literature on conceptual modelling as it is cast aside for more 'scientific' endeavours. This seems strange, since it is acknowledged to be among the most difficult aspects in performing a simulation study.⁹⁷ Validation is better served by researchers and practitioners,^{98–101} but it is still an area that could be improved upon. Perhaps it is in these two areas that an improvement in the model development process is most needed.

So where might a change in the development of simulation models come from? Model and component reuse seem to have some potential for helping to speed and ease the development of simulations, especially if the components could be shared between model developers. Indeed, this might satisfy the simulation modellers of the future who could find parts of their models on the worldwide web. At its lowest level, component reuse is already available in simulation packages as users select from a menu of basic components (eg machines or queues). Above this level, examples are harder to find.

Pidd⁸⁶ and Paul and Taylor⁸⁵ discuss the potential for model reuse. Although neither argue that the concept of reuse should be abandoned by the simulation community, they both express some concerns and barriers to reuse. First, there is the time required to learn, test and adjust someone else's model/component, which may be greater than the time to develop the simulation from scratch. Second, models/components built for one purpose may not be suited to another purpose, even if it is the same type of system that is being investigated. Finally, there is the problem that the costs fall upon the supplier of the model/component while the benefits go to the user. It may be that because of these issues reuse will always be limited to relatively low level

components and the potential for higher level reuse will be limited.

To counter this view, and to satisfy the simulation modellers of the future, Paul¹⁰² argues for an alternative model development process: grab-and-glue, run, reject, retry (G^2R^3). The modeller *grabs* bits of models from the web and *glues* them together. The model is then *run* to see if it provides insights into the problem being addressed. If it does, then the insights can be used to improve the problem situation. If it does not, then the modeller *rejects* the model and *retries* by returning to the grab and glue process. This idea certainly seems to match the expectations of the 'Sims' generation.

Another route for changing the development of simulation models is to automate (or at least guide) the process. The modeller could provide a description of the model and an interpreter would convert this into executable model code. This is not a new idea. Clementson's¹⁰³ CAPS/ECSL was motivated by this approach, as was Balmer and Paul's¹⁰⁴ investigations into a computer-aided simulation environment (CASM). More recently, researchers have turned their attention to the unified modelling language (UML) as a means for describing and then generating simulation code.¹⁰⁵

Ultimately, the development of simulation models might be improved most easily by providing better training for model developers. Although software vendors provide training in the use of their packages, in general they do not train their users to be simulation modellers. Simulation modelling is more than just using a package and requires modelling, statistical, communication and project management skills as well. In order to train modellers effectively, there needs to be a set of well-defined methods for developing simulation models. This returns us to the need for improvements in the areas of conceptual modelling and validation.

Model use

Hollocks¹⁰⁶ identifies the need for additional help in the experimentation stage of a simulation study. Optimizer software has provided some help in this respect, but there are still a great many (difficult) decisions left to the user. Issues such as determining the warm-up period, setting initial conditions, deciding on how long to run the simulation and determining the number of replications to perform all need to be addressed. Sensitivity analysis and experimental design procedures are also useful when there are uncertainties in the data and many scenarios to consider respectively. Methods exist that could be used to guide a user in all of these areas, but there is little evidence of these methods being embodied in commercial simulation software. AutoStat (part of the AutoMod software) is one of the few exceptions providing output analysis features including warm-up, confidence intervals, sensitivity analysis and design of experiments.⁴⁹

In a survey of simulation users, Hlupic¹⁰⁷ identified better experimental support as the highest ranked requirement. This could either be addressed by developing software to guide simulation users or even automating aspects of the experimental process. Without this, the 'Sims' generation may fall into the various traps that can occur when experimenting with simulation models, as indeed have the current generation of simulation users.

The domain of application for simulation

Simulation still remains a tool that is largely used in the design stage of a system, particularly as it is used in industry.¹⁰⁸ One way in which the use of simulation could be broadened is to look for new domains of application in the sectors where it is already being used. There remains a potential for simulation to be applied to areas such as:

- Emulation to aid the design of control systems
- Scheduling
- Predicting future performance, for example, the time to process an individuals insurance claim
- Real-time control
- Training

Another avenue to consider is the modelling of human behaviour and interaction with an operations system. The prime motivation here should not be to aim for more accurate models, as this is unlikely to succeed when data and understanding are limited. Instead, the purpose should be to better understand how human interaction with an operations system affects the performance of that system and to look for ways of improving the actions, behaviours and decisions of the human actors. Some work has been carried out in this area, for instance, Flitman and Hurron,⁵³ O'Keefe,⁵⁴ Williams,¹⁰⁹ Lyu and Gunasekaran,¹¹⁰ Moffat,⁵⁵ Baines and Kay,⁵⁶ Brailsford and Schmidt¹¹¹ and Robinson *et al.*⁵⁷ There is certainly room for more investigations of simulating humans in an operations system.

Altering the perception of simulation as a 'hard' OR tool, to one that can be useful in facilitating debate around a problem situation, could also help to broaden its use.¹¹² The requirement here is to be able to build and use (very approximate) simulations rapidly, possibly in a group decision-making environment. This returns to the question of the model development process and model use.

Integration with other simulation approaches

Another avenue that discrete-event simulation might follow is a closer integration with other simulation techniques and communities. System dynamics, agent-based simulation, simulation games and continuous simulation have all followed separate paths in both research and practice. The computer games industry is heavily reliant on simulation methods and has been extremely successful in developing

and transferring this technology to the market place. There are probably many lessons to learn and technologies that could be adopted from the computer games industry as well as from across each of the simulation communities. The author's own forays into the world of system dynamics have proved beneficial in forming ideas about how discrete-event simulation might be practiced.¹¹² Indeed, the system dynamics community has a well-developed philosophy of practice.^{113,114} This is something that does not exist among the advocates of discrete-event simulation.

As already stated, the younger generation are being brought up on the idea of simulation through their gaming environments. When they are older should it not be natural for them to want to simulate their working environment? Perhaps we need to learn now how this might be achieved in a way that will satisfy their expectations?

Conclusion: simulation and computing still conjoined?

Since the 1950s the development of simulation and computing have been very closely aligned. However, this path seems to have separated a little with the tardy uptake of the worldwide web by the simulation community. The question is, will simulation continue to follow trends in computing or will the separation become greater? And perhaps of even greater importance, should simulation continue to follow trends in computing? In concluding this review of simulation, these questions are considered with respect to two areas, the worldwide web and the continued increase in computing power.

The prevalence of the worldwide web is only likely to increase. Surely, simulation software providers will be forced to take greater advantage of the worldwide web under the weight of expectation from simulation users. Over the next few years, it would seem likely that software licensing arrangements will become more flexible making it possible to distribute, at least, a series of replications. Distributing the execution of a single model will probably remain an intractable problem, certainly, in terms of generality and ease-of-use, for some time to come. Software vendors are also in a position to encourage users to share model components by setting standards and putting in-place a financial arrangement that is beneficial to both the developer and user. The sharing of model components between different simulation software is less likely to succeed in the short-term as competitive pressures will tend to encourage non-compliance. In summary, it seems that simulation will follow trends in the worldwide web, but that it will continue to lag behind.

Computing power has always been a bottleneck to the development of simulation. Memory has limited the size of the models that could be developed and clock speed has restricted the experiments that could be run. Will this always be the case? It would seem that in recent years this

bottleneck has to an extent been removed. Certainly, it appears that simulation users complain less than they did about the restrictions on the size and to the speed of their models. Meanwhile, the power of the PC is set to continually increase, and should distributed execution become available, this will increase computing power even further.

With increasing computing power comes a problem for the simulation community. While more powerful computers extend the possibilities of simulation to larger and more complex problems, it also tends to lead users to design their simulations less carefully, simply because they know the computer can handle it. Poor design implies that models are probably more complex than they need to be. Against this idea is the constant advice of modelling experts that models should be kept as simple as possible.^{115,46} There are a number of benefits to be gained from simple models, among them are: faster development, greater flexibility, less data required and it is easier to understand and interpret the results.¹¹⁶⁻¹¹⁹ As the complexity increases, these benefits are lost.

The simulation community needs to address this issue by attending to the wider methodology of model development and use. This is an area in which there is only limited research taking place. In particular, there is a need to develop methods for aiding sensible model design (conceptual modelling) and training users accordingly. Without this, instead of computing being a bottleneck to the development of simulation, it might become a millstone.

So will simulation continue to follow trends in computing? The answer seems almost certainly to be 'yes', although for the time-being there will probably be a lag between the two, particularly in terms of utilizing the worldwide web. Should simulation continue to follow trends in computing? A cautious 'yes', but only if the wider methodological issues of simulation model development and use are properly addressed.

References

- 1 Jeffrey P and Seaton R (1995). The use of operational research tools: a survey of operational research practitioners in the UK. *J Opl Res Soc* **46**(7): 797-808.
- 2 Fildes R and Ranyard JC (1997). Success and survival of operational research groups — a review. *J O Res Soc* **48**(4): 336-360.
- 3 Clark DN (1999). Strategic level MS/OR tool usage in the United Kingdom and New Zealand: a comparative survey. *Asia-Pacific J Opl Res* **16**(1): 35-51.
- 4 Wild R (2002). *Operations Management*, 6th edn. Continuum: London.
- 5 Pidd M (1984). Computer simulation for operational research in 1984. In: Eglese R and Rand G (eds). *Developments in Operational Research*. Pergamon Press: Oxford, UK, pp 19-30.
- 6 Bell PC and O'Keefe RM (1987). Visual interactive simulation — history, recent developments, and major issues. *Simulation* **49**(3): 109-116.

- 7 Paul RJ (1991). Recent developments in simulation modelling. *J Opl Res Soc* **42**(3): 217–226.
- 8 Bell PC (1991). Visual interactive modelling: the past, the present, and the prospects. *Eur J Opl Res* **54**: 274–286.
- 9 Hollocks BW (2004). Still simulating after all these years — reflections on 40 years in Simulation. In: Brailsford SC, Oakshott L, Robinson S and Taylor SJE (eds). *Proceedings of the 2004 Operational Research Society Simulation Workshop (SW04)*. Operational Research Society, Birmingham, pp 209–222.
- 10 Araten M *et al* (1992). The winter simulation conference: perspectives of the founding fathers. In: Swain JJ, Goldsman D, Crain RC and Wilson JR (eds). *Proceedings of the 1992 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 37–63.
- 11 Banks J, Carson JS, Nelson BL and Nicol DM (2001). *Discrete-Event System Simulation*, 3rd edn. Prentice-Hall: Upper Saddle River, NJ.
- 12 Nance RE and Sargent RG (2002). Perspectives on the evolution of simulation. *Operations Research* **50**(1): 161–172.
- 13 Tocher KD (1963). *The Art of Simulation*. The English Universities Press: London.
- 14 Schriber T (1974). *Simulation Using GPSS*. Wiley: New York.
- 15 Markowitz HM, Hausner B and Karr HW (1962). *SIMSCRIPT: The Simulation Programming Language*. RAND Corporation, Cambridge, MA, RM-3310.
- 16 Dahl O and Nygaard K (1966). SIMULA: an algol-based simulation language. *Comm ACM* **9**(9): 671–678.
- 17 Amiry AP (1965). The simulation of information flow in a steelmaking plant. In: Hollingdale S (ed). *Digital Simulation in Operational Research*. English University Press, London, pp 347–356.
- 18 Pritsker AAB and Pegden CD (1979). *Introduction to Simulation and SLAM*. Wiley: New York.
- 19 Wolverine. www.wolverinesoftware.com/wolverin.htm, accessed March 2004.
- 20 Hurriion RD (1976). *The design, use and required facilities of an interactive visual computer simulation language to explore production planning problems*. PhD thesis, University of London.
- 21 Fiddy E, Bright JG and Hurriion RD (1981). SEE-WHY: interactive simulation on the screen. *Proceedings of the Institute of Mechanical Engineers C293/81, I. Mech. E.*, London, pp 167–172.
- 22 Gilman AR and Billingham C (1989). A tutorial on SEE WHY and WITNESS. In: MacNair EA, Musselman KJ and Heidelberger P (eds). *Proceedings of the 1989 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 192–200.
- 23 Szymankiewicz J, McDonald J and Turner K (1988). *Solving Business Problems by Simulation*. McGraw-Hill: Maidenhead, UK.
- 24 Concannon K and Becker P (1990). A tutorial on GENETIK simulation and scheduling. In: Balci O, Sadowski RP and Nance RE (eds). *Proceedings of the 1990 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 140–145.
- 25 Sturrock DT and Pegden CD (1989). Introduction to SIMAN. In: MacNair EA, Musselman KJ and Heidelberger P (eds). *Proceedings of the 1989 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 129–139.
- 26 Poorte JP and Davis DA (1989). Computer animation with CINEMA. In: MacNair EA, Musselman KJ and Heidelberger P (eds). *Proceedings of the 1989 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 147–154.
- 27 Harrell CR and Tumay K (1990). ProModel PC tutorial. In: Balci O, Sadowski RP and Nance RE (eds). *Proceedings of the 1990 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 128–131.
- 28 Hollocks B (1992). A well-kept secret? *Simulation in Manufacturing Industry Reviewed. OR Insight* **5**(4): 12–17.
- 29 Pegden CD and Davis DA (1992). Arena: a SIMAN/cinema-based hierarchical modeling system. In: Swain JJ, Goldsman D, Crain RC and Wilson JR (eds). *Proceedings of the 1992 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 390–399.
- 30 Hugan JC (1995). QUEST – queueing event simulation tool. In: Alexopoulos C, Kang K, Lilegdon WR and Goldsman D (eds). *Proceedings of the 1995 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 432–436.
- 31 Nordgren WB (1995). Taylor II manufacturing Simulation Software. In: Alexopoulos C, Kang K, Lilegdon WR and Goldsman D (eds). *Proceedings of the 1995 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 401–404.
- 32 Rohrer MW (1996). AutoMod Tutorial. In: Charnes JM, Morrice DM, Brunner DT and Swain JJ (eds). *Proceedings of the 1995 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 500–505.
- 33 Pritsker AAB and O'Reilly JJ (1996). AweSim: the integrated simulation system. In: Charnes JM, Morrice DM, Brunner DT and Swain JJ (eds). *Proceedings of the 1996 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 481–484.
- 34 Barnes CD and Laughery Jr KR (1997). Advanced uses for micro saint simulation software. In: Andradóttir S, Healy KJ, Withers DH and Nelson BL (eds). *Proceedings of the 1997 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 680–686.
- 35 Hullinger DR (1999). Taylor enterprise dynamics. In: Farrington PA, Nembhard HB, Sturrock DT and Evans GW (eds). *Proceedings of the 1999 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 227–229.
- 36 Nordgren WB (2002). Flexsim simulation environment. In: Yücesan E, Chen C-H, Snowden SL and Charnes JM (eds). *Proceedings of the 2002 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 250–252.
- 37 Pidd M (1998). *Computer Simulation in Management Science*, 4th edn. Wiley: Chichester, UK.
- 38 Simul8. www.simul8.com, accessed March 2004.
- 39 Kralh D (1996). Modeling with extend. In: Charnes JM, Morrice DM, Brunner DT and Swain JJ (eds). *Proceedings of the 1996 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 578–583.
- 40 ShowFlow. www.showflow.co.uk, accessed March 2004.
- 41 Law AM and McComas MG (2002). Simulation-based optimization. In: Yücesan E, Chen C-H, Snowden SL and Charnes JM (eds). *Proceedings of the 2002 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 41–44.
- 42 Law AM and Kelton WD (2000). *Simulation Modeling and Analysis*, 3rd edn. McGraw-Hill: New York.
- 43 Fu MC (2002). Optimization for simulation: theory vs practice. *INFORMS J Comput* **14**(3): 192–215.
- 44 Olafsson S and Kim J (2002). Simulation optimization. In: Yücesan E, Chen C-H, Snowden SL and Charnes JM (eds). *Proceedings of the 2002 Winter Simulation Conference*. IEEE, Piscataway, NJ, pp 79–84.
- 45 Hurriion RD (2000). A sequential method for the development of visual interactive meta-simulation models using neural networks. *J Opl Res Soc* **51**(6): 712–719.
- 46 Robinson S (2004). *Simulation: The Practice of Model Development and Use*. Wiley: Chichester, UK.
- 47 Lanner. www.lanner.co.uk, accessed March 2004.

- 48 Harrell CR and Price RN (2003). Simulation modeling using promodel technology. In: Chick S, Sanchez PJ, Ferrin D and Morrice DJ (eds). Proceedings of the 2003 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 175–181.
- 49 Rohrer MW and McGregor IW (2002). Simulating reality using AutoMod. In: Yücesan E, Chen C-H, Snowden SL and Charnes JM (eds). Proceedings of the 2002 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 173–181.
- 50 April J, Glover F, Kelly J and Laguna M (2001). Simulation/Optimization Using 'Real-World' Applications. In: Peters BA, Smith JS, Medeiros DJ and Rohrer MW (eds). Proceedings of the 2001 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 134–138.
- 51 Waller AP and Ladbroke J (2002). Experiencing virtual factories of the future. In: Yücesan E, Chen C-H, Snowden SL and Charnes JM (eds). Proceedings of the 2002 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 513–517.
- 52 Preddy SM and Nance RE (2002). Key requirements for CAVE simulations. In: Yücesan E, Chen C-H, Snowden SL and Charnes JM (eds). Proceedings of the 2002 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 127–135.
- 53 Flitman AM and Hurriion RD (1987). Linking discrete-event simulation models with expert systems. *J Opl Res Soc* **38**(8): 723–734.
- 54 O'Keefe RM (1989). The Role of artificial intelligence in discrete-event simulation. In: Widman LE, Loparo KA and Neilsen NR (eds). Artificial Intelligence, Simulation and Modeling. Wiley, New York, pp 359–379.
- 55 Moffat J (2000). Representing the command and control process in simulation models of conflict. *J Opl Res Soc* **51**(4): 431–439.
- 56 Baines TS and Kay JM (2002). Human performance modelling as an aid in the process of manufacturing system design: a pilot study. *Int J Prod Res* **40**(10): 2321–2334.
- 57 Robinson S *et al* (2003). Modelling and improving maintenance decisions: having foresight with simulation and artificial intelligence. *SAE 2002 Trans J Mater Manuf*, **VIII**(5), 256–264.
- 58 Gatersleben M and van der Weij SW (1999). Analysis and simulation of passenger flows in an airport terminal. In: Farrington PA, Nemhard HB, Sturrock DT and Evans GW (eds). Proceedings of the 1999 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 1226–1231.
- 59 Anton J (1999). *Call Center Performance Enhancement Using Simulation and Modeling*. Purdue University Press: West Lafayette, IN.
- 60 Melao N and Pidd M (2003). Use of business process simulation: a survey of practitioners. *J Opl Res Soc* **54**(1): 2–10.
- 61 Hueter J and Swart W (1998). An integrated labor-management system for Taco bell. *Interfaces* **28**(1): 75–91.
- 62 Jun JB, Jacobson SH and Swisher JR (1999). Application of discrete-event simulation in health care clinics: a survey. *J Opl Res Soc* **50**(2): 109–123.
- 63 Hlupic V and Robinson S (1998). Business process modelling and analysis using discrete-event simulation. In: Medeiros DJ, Watson EF, Manivannan M and Carson J (eds). Proceedings of the 1998 Winter Simulation Conference 1998 Piscataway, NJ, pp 1363–1369.
- 64 Fishwick PA (1996). Web-based simulation: some personal observations. In: Charnes JM, Morrice DM, Brunner DT and Swain JJ (eds). Proceedings of the 1996 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 772–779.
- 65 Buss AH and Stork KA (1996). Discrete event simulation on the world wide web using Java. In: Charnes JM, Morrice DM, Brunner DT and Swain JJ (eds). Proceedings of the 1996 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 780–785.
- 66 Nair RS, Miller JA and Zhang Z (1996). Java-based query driven simulation environment. In: Charnes JM, Morrice DM, Brunner DT and Swain JJ (eds). Proceedings of the 1996 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 786–793.
- 67 Koh K-H, de Souza R and Ho N-C (1996). Multi-processor distributed simulation for job-shop scheduling: boon or bane? *Int J Comput Integr Manuf* **9**(6): 434–442.
- 68 Rao DM and Wilsey PA (2002). An ultra-large-scale simulation framework. *Journal of Parallel Distrib Comput* **62**: 1670–1693.
- 69 Gabbar HA, Shinohara S, Shimada Y and Suzuki K (2003). Experiment on distributed dynamic simulation for safety design of chemical plants. *Simul Modell Pract Theory* **11**: 109–123.
- 70 Fujii S, Kaihara T and Morita H (2000). A distributed virtual factory in agile manufacturing Environment. *Int J Prod Res* **38**(17): 4113–4128.
- 71 Zülch G, Jonsson U and Fischer J (2002). Hierarchical simulation of complex production systems by coupling models. *Int J Prod Econ* **77**: 39–51.
- 72 Korn S, Burns GR and Harrison DK (1999). The application of multiparadigm simulation techniques to manufacturing processes. *Int J Adv Manuf Technol* **15**: 869–875.
- 73 Davis WJ (1998). On-line simulation: need and evolving research requirements. In: Banks J (ed). *Handbook of Simulation*. Wiley, New York, pp 465–516.
- 74 Anagnostopoulos D and Nikolaidou M (2003). Executing a minimum number of replications to support the reliability of FRTS predictions. In: Turner SJ and Taylor SJE (eds). Proceedings of the seventh IEEE International Symposium on Distributed Simulation and Real-Time Applications. IEEE Computer Society, Los Alamitos, CA, pp 138–146.
- 75 Biles WE and Kleijnen JPC (2003). Statistical methodology for WEB-based simulation. In: Turner SJ and Taylor SJE (eds). Proceedings of the seventh IEEE International Symposium on Distributed Simulation and Real-Time Applications IEEE Computer Society, Los Alamitos, CA, pp 147–149.
- 76 Yücesan E, Luo Y-C, Chen C-H and Lee I (2001). Distributed web-based simulation experiments for optimization. *Simul Pract Theory* **9**: 73–90.
- 77 Paris J-L and Pierrel H (2001). A distributed evolutionary simulation optimization approach for configuration of multiproduct Kanban systems. *Int J Comp Integr Manuf* **14**(5): 421–430.
- 78 Mizuta H and Yamagata Y (2002). Transaction cycle of agents and web-based gaming simulation for international emissions trading. In: Yücesan E, Chen C-H, Snowden SL and Charnes JM (eds). Proceedings of the 2002 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 801–806.
- 79 Jacobs PHM, Lang NA and Verbraeck A (2002). D-SOL; A distributed Java based discrete event simulation architecture. In: Yücesan E, Chen C-H, Snowden SL and Charnes JM (eds). Proceedings of the 2002 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 793–800.
- 80 Pooley R and Wilcox P (2000). Distributing decision making using Java simulation across the world wide web. *J Opl Res Soc* **51**(4): 395–404.
- 81 Wang Y-H and Liao Y-C (2003). Implementation of a collaborative web-based simulation modeling environment. In: Turner SJ and Taylor SJE (eds). Proceedings of the seventh IEEE International Symposium on Distributed Simulation and Real-Time Applications IEEE Computer Society, Los Alamitos, CA, pp 150–157.

- 82 Kilgore RA (2001). Open-source SML and silk for Java-based, object-oriented simulation. In: Peters BA, Smith JS, Medeiros DJ and Rohrer MW (eds). Proceedings of the 2001 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 262–268.
- 83 Taylor SJE, Robinson S and Ladbroke J (2003). Towards collaborative simulation modelling: improving human-to-human interaction through groupware. In: Al-Dabass D (ed). Proceedings of the 17th European Simulation Multiconference (ESM 2003). Society for Computer Simulation, Delft, pp 474–482.
- 84 Linebarger JM, Janneck CD and Kessler GD (2003). Shared simple virtual environment: an object-oriented framework for highly interactive group working. In: Turner SJ and Taylor SJE (eds). Proceedings of the seventh IEEE International Symposium on Distributed Simulation and Real-Time Applications IEEE Computer Society, Los Alamitos, CA, pp 170–180.
- 85 Paul RJ and Taylor SJE (2002). What use is model reuse: is there a crook at the end of the rainbow?. In: Yücesan E, Chen C-H, Snowden SL and Charnes JM (eds). Proceedings of the 2002 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 648–652.
- 86 Pidd M (2002). Simulation software and model reuse: a polemic. In: Yücesan E, Chen C-H, Snowden SL and Charnes JM (eds). Proceedings of the 2002 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 772–775.
- 87 International Review of Operational Research in the United Kingdom. Kensington Hilton, London, 5 March 2004. unpublished.
- 88 Edwards JS *et al* (2004). Using a simulation model for knowledge elicitation and knowledge management. *Simul Modell Pract Theory*, Forthcoming.
- 89 Brooks RJ and Tobias AM (1996). Choosing the best model: level of detail, complexity and model performance. *Math Comput Modell* **24**(4): 1–14.
- 90 Eldabi T, Robinson S, Taylor SJE and Wilcox PA (eds) (2002). Proceedings of the Operational Research Society Simulation Workshop. Operational Research Society: Birmingham, UK.
- 91 Brailsford SC, Oakshott L, Robinson S and Taylor SJE (eds) (2004). Proceedings of the 2004 Operational Research Society Simulation Workshop (SW04). Operational Research Society: Birmingham, UK.
- 92 Chick S, Sanchez PJ, Ferrin D and Morrice DJ (eds) (2003). Proceedings of the 2003 Winter Simulation Conference. IEEE: Piscataway, NJ.
- 93 Solomon (700 BC). Proverbs Chapter 29, Verse 18. In: *The Bible, Authorized King James Version*. Cambridge University Press, Cambridge, UK.
- 94 Amory A, Naicker K, Vincent J and Adams C (1999). The use of computer games as an educational tool: identification of appropriate game types and game elements. *Br J Educ Technol* **30**(4): 311–321.
- 95 Barton RR *et al* (2003). Panel: simulation — past, present and future. In: Chick S, Sanchez PJ, Ferrin D and Morrice DJ (eds). Proceedings of the 2003 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 2044–2050.
- 96 Pidd M (1992). Object orientation and three phase simulation. In: Swain JJ, Goldsman D, Crain RC and Wilson JR (eds). Proceedings of the 1992 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 689–693.
- 97 Law AM (1991). Simulation model's level of detail determines effectiveness. *Industrial Engineering* **23**(10): 16–18.
- 98 Sargent RG (1992). Validation and verification of simulation models. In: Swain JJ, Goldsman D, Crain RC and Wilson JR (eds). Proceedings of the 1992 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 104–114.
- 99 Balci O (1994). Validation, verification, and testing techniques throughout the life cycle of a simulation study. *Ann Opns Res* **53**: 121–173.
- 100 Balci O (1997). Principles of simulation model validation, verification, and testing. *Trans Soc Comput Simul* **14**(1): 3–12.
- 101 Robinson S (1999). Simulation verification, validation and confidence: a tutorial. *Trans Soc Comp Simul* **16**(2): 63–69.
- 102 Robinson S *et al* (2004). Simulation model reuse: definitions, benefits and obstacles. *Simul Modell Pract Theory*, Forthcoming.
- 103 Clementson AT (1986). Simulating with activities using CAPS/ECSL (the British approach to discrete-event simulation). In: Wilson J, Henriksen J and Roberts S (eds). Proceedings of the 1986 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 113–122.
- 104 Balmer D and Paul RJ (1986). CASM — the right environment for simulation. *Journal of the Operational Research Society* **37**(5): 443–452.
- 105 Arief LB and Speirs NA (2000). A UML tool for an automatic generation of simulation programs. In: Proceedings of the second International Workshop on Software and Performance (WOSP 2000) ACM Press: New York, pp 71–76.
- 106 Hollocks BW (2001). Discrete-event simulation: an inquiry into user practice. *Simulation Practice and Theory* **8**: 451–471.
- 107 Hlupic V (1999). Discrete-event simulation software: what the users want. *Simulation* **73**(6): 362–370.
- 108 Banks J *et al* (2003). The future of the simulation industry. In: Chick S, Sanchez PJ, Ferrin D and Morrice DJ (eds). Proceedings of the 2003 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 2033–2043.
- 109 Williams T (1996). Simulating the man-in-the-loop. *OR Insight* **9**(4): 17–21.
- 110 Lyu J and Gunasekaran A (1997). An intelligent simulation model to evaluate scheduling strategies in a steel company. *Int J Syst Sci* **28**(6): 611–616.
- 111 Brailsford S and Schmidt B (2003). Towards Incorporating human behaviour in models of health care systems: an approach using discrete event simulation. *Eur J Opl Res* **150**: 19–31.
- 112 Robinson S (2001). Soft with a hard centre: discrete-event simulation in facilitation. *J Opl Res Soc* **52**(8): 905–915.
- 113 Lane DC (1999). Social theory and system dynamics practice. *Eur J Opl Res* **113**: 501–527.
- 114 Morecroft J (2004). Mental models and learning in system dynamics practice. In: Pidd M (ed). *Systems Modelling: Theory and Practice*. Wiley, Chichester, UK, pp 101–126.
- 115 Pidd M (2003). *Tools for Thinking: Modelling in Management Science*, 2nd edn. Wiley: Chichester, UK.
- 116 Innis G and Rexstad E (1983). Simulation model simplification techniques. *Simulation* **41**(1): 7–15.
- 117 Ward SC (1989). Arguments for constructively simple models. *J Opl Res Soc* **40**(2): 141–153.
- 118 Salt J (1993). Simulation should be easy and fun. In: Evans GW, Mollaghasemi M, Russell EC and Biles WE (eds). Proceedings of the 1993 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 1–5.
- 119 Chwif L, Barretto MRP and Paul RJ (2000). On simulation model complexity. In: Joines JA, Barton RR, Kang K and Fishwick PA (eds). Proceedings of the 2000 Winter Simulation Conference. IEEE, Piscataway, NJ, pp 449–455.

Received March 2004;
accepted July 2004 after one revision