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

This item was submitted to Loughborough University's Institutional Repository by an author.


Additional Information:

- This article was published in the Journal of the Operational Research Society [Palgrave Macmillan © OR Society] and the definitive version is available at: http://dx.doi.org/10.1057/palgrave.jors.2601864

Metadata Record: https://dspace.lboro.ac.uk/2134/10207

Version: Accepted for publication

Publisher: Palgrave Macmillan © Operational Research Society

Please cite the published version.
This item was submitted to Loughborough’s Institutional Repository (https://dspace.lboro.ac.uk/) by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to: http://creativecommons.org/licenses/by-nc-nd/2.5/
Discrete-Event Simulation: From the Pioneers to the Present, What Next?

Stewart Robinson

Warwick Business School
University of Warwick
Coventry
CV4 7AL
United Kingdom
Tel: 02476-522132
Email: stewart.robinson@warwick.ac.uk
Discrete-Event Simulation: From the Pioneers to the Present, What Next?

Abstract

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 optimisation, virtual reality, integration with other software, simulation in the service sector, distributed simulation and the use of the world wide 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.

Key Words

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

Introduction

Discrete-event simulation is one of the most commonly used modelling techniques \(^1,^2,^3\). 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'\(^4\). Wild \(^4\) 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 organisations. 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 (1950s-1980s)

In describing the history of simulation four distinct periods can be identified, referred to here as the periods of 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, Bell and O'Keefe, Paul, Bell and Hollocks. For an American perspective see Araten et al, Banks et al, and Nance and Sargent.

A key feature of the development of simulation is that it has closely followed the development of computing. 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 (1950s to 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, Tocher\textsuperscript{13} published the three-phase simulation approach which is still used by a number of simulation packages today. The 1960s also saw the development of the first specialist simulation software, for example, GPSS\textsuperscript{14} around 1961 and SIMSCRIPT\textsuperscript{15}. SIMULA\textsuperscript{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\textsuperscript{17}.

\textit{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, SLAM\textsuperscript{18} and GPSS-H\textsuperscript{19}. The late 1970s saw the introduction of the first microcomputers. Further to this, Hurrion\textsuperscript{20} 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.

\textit{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 organisations. Two things were about to change all this. Firstly, reasonably powerful microcomputers started to become common place in organisations, not least because of IBM’s entry into the market. Secondly, the development of VIS software, starting with SEE-WHY\textsuperscript{21} 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\textsuperscript{22}, HOCUS\textsuperscript{23}, GENETIK\textsuperscript{24}, SIMAN/CINEMA\textsuperscript{25,26} and ProModel\textsuperscript{27}. Many organisations, 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\textsuperscript{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 world wide web. Simulation certainly took advantage of all these, with the possible exception of the world wide 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's\(^8\) 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 optimisation, 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 world wide 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, WITNESS\(^22\) and ProModel\(^27\) 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 Dynamics\(^35\) and Flexsim\(^36\). Pidd\(^37\) refers to these packages as visual interactive modelling systems (VIMS), recognising 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 organisation 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\(^{38}\), Extend\(^{39}\) and more recently ShowFlow\(^{40}\), which is
a re-launch of the Taylor II software. These are all sold for less than £1,000.

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 organisations that have started using 'low cost' VIMS and have later moved to a
more expensive package as their requirements change.

*Simulation Optimisation*

Law and McComas\(^{41}\) describe simulation optimisation 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 optimisation facility. The argument may
come over the extent to which the ready availability of optimisation capabilities has changed
simulation practice.
The problem is similar to that for any mathematical optimisation 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 optimisation is that no algorithms exist that guarantee an optimal solution will be found.

Simulation optimisation has long been of interest within the research community and various approaches have been investigated over the years\textsuperscript{42,43,44}. One such approach is based on response surface methodology or metamodelling, 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\textsuperscript{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 'optimiser' 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, Robinson\textsuperscript{46} argues that the term 'searchisation' rather than 'optimisation' more closely describes the function of these optimisers.

The majority of simulation software vendors seem to have adopted metaheuristic approaches, for example, WITNESS Optimizer\textsuperscript{47}, SimRunner for ProModel\textsuperscript{48} and AutoStat for
AutoMod\textsuperscript{49}. Also based on metaheuristics, the OptQuest software\textsuperscript{50} has been adopted as the optimisation engine for a number of packages including Arena, Enterprise Dynamics and Simul8. Although these optimisers 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 optimisation is something of an anathema to VIS, which emphasises the benefits of interaction with the model.

Fu\textsuperscript{43} describes a number of other methods of optimisation for simulation. In doing so he notes a divergence between research and practice in simulation optimisation. The software vendors, described above, have adopted general algorithms taken from the realm of deterministic optimisation. 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.

\textit{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\textsuperscript{49}, Flexsim\textsuperscript{36} and WITNESS VR\textsuperscript{51}. QUEST\textsuperscript{30} 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 3-dimensional animations that can be viewed from different perspectives.

There is a question over the benefit of such displays. Waller and Ladbrook\textsuperscript{51} 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 realised at a reasonable cost. Meanwhile, Preddy and Nance\textsuperscript{52} 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\textsuperscript{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\textsuperscript{55,56,57}.

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\textsuperscript{58}, call centres\textsuperscript{59}, business processes\textsuperscript{60}, fast food restaurants\textsuperscript{61} and health care clinics\textsuperscript{62}. The business process reengineering 'fad' surely had some effect on the demand for simulation as the need for simulation as part of the reengineering process was identified\textsuperscript{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 World Wide Web and Distributed Simulation

The greatest single change in computing over the past 10 to 15 years has obviously been the growth of the world wide 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 world wide web took place at the 1996 Winter Simulation Conference\textsuperscript{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\textsuperscript{65,66}.

Since that time, researchers have suggested many potential applications for simulation on the world wide 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 computers\textsuperscript{67,68,69} or separate models on several computers could be linked and run concurrently\textsuperscript{70,71}. Simulation models might be linked to remote third party software during execution, such as databases, artificial intelligence software or real-time systems\textsuperscript{72,73}. The speed and range of experimentation could be helped by distributing replications across a number of computers\textsuperscript{74,75} or by distributing different experimental scenarios\textsuperscript{76,77}. Distributed users could interact with a gaming simulation from remote locations\textsuperscript{78,79}. The process of performing a simulation study could be helped by sharing models and software between distributed users\textsuperscript{80,81,82}. Virtual meetings could improve communication during model development and use\textsuperscript{83,84}. Model developers could search for and download model components over the world wide web, reducing model development time\textsuperscript{85}.

Despite a whole range of opportunities and the prevalence of the world wide 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\textsuperscript{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\textsuperscript{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 up-take of the opportunities afforded by the world wide web. A key reason is that distributed simulation is difficult to implement. A major problem is synchronising 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, Pidd\textsuperscript{86} 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 world wide 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 Untied Kingdom\textsuperscript{87} 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 e.g. distributed simulation, model design methods
- Output analysis, experimental design and optimisation e.g. bootstrap methods, warm-up period determination
- Simulation practice e.g. surveys of simulation use
- Human performance modelling e.g. modelling worker performance and human decision-making
• Healthcare modelling e.g. 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\textsuperscript{88}, business processes\textsuperscript{60} and complexity science\textsuperscript{89}.

Various networks exist for sharing ideas across the research and practitioner community. GROUPSIM, centred at Brunel University, has been looking into strategies and infrastructure for collaborative simulation modelling. Cranfield University has set-up HPM-net which 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 one-day and half-day meetings. Its biennial conference\textsuperscript{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\textsuperscript{92}, which is the major international event in discrete-event simulation.

\textbf{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 world wide web, in other words to piece together someone else's 'code' \(^{85}\). What is more, they search in a fairly unstructured fashion 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’\textsuperscript{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\textsuperscript{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.\textsuperscript{97} Validation is better served by researchers and practitioners\textsuperscript{98,99,100,101}, 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 world wide web. At its lowest level, component reuse is already available in simulation packages as users select from a menu of basic components (e.g. machines or queues). Above this level, examples are harder to find.

Pidd\textsuperscript{86} and Paul and Taylor\textsuperscript{85} 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, Paul\textsuperscript{102} argues for an alternative model development process: grab-and-glue, run, reject, retry (G\textsuperscript{2}R\textsuperscript{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's CAPS/ECSL\textsuperscript{103} was motivated by this approach, as was Balmer and Paul’s investigations into a computer aided simulation environment (CASM)\textsuperscript{104}. More recently, researchers have turned their attention to the unified modelling language (UML) as a means for describing and then generating simulation code\textsuperscript{105}.  

17
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.

 Modelo Use

Hollocks\textsuperscript{106} identifies the need for additional help in the experimentation stage of a simulation study. Optimiser 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\textsuperscript{49}.

In a survey of simulation users, Hlupic\textsuperscript{107} 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\textsuperscript{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 e.g. 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 Hurrion, O'Keefe, Williams, Lyu and Gunasekaran, Moffat, Baines and Kay, Brailsford and Schmidt, and Robinson et al. 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. 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\textsuperscript{112}. Indeed, the system dynamics community has a well developed philosophy of practice\textsuperscript{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 world wide 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 world wide web and the continued increase in computing power.

The prevalence of the world wide web is only likely to increase. Surely simulation software providers will be forced to take greater advantage of the world wide 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 world wide 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\textsuperscript{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\textsuperscript{116,117,118,119}. 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 utilising the world wide 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


