Journal of the Operational Research Society (2008) 59, 278–290 0160-5682/08 $30.00

www.palgrave-journals.com/jors

Conceptual modelling for simulation Part I: definition and requirements

S Robinson*

University of Warwick, Coventry, UK

Conceptual modelling is probably the most important aspect of a simulation study. It is also the most difficult and least understood. Over 40 years of simulation research and practice have provided only limited information on how to go about designing a simulation conceptual model. This paper, the first of two, discusses the meaning of conceptual modelling and the requirements of a conceptual model. Founded on existing literature, a definition of a conceptual model is provided. Four requirements of a conceptual model are described: validity, credibility, utility and feasibility. The need to develop the simplest model possible is also discussed. Owing to a paucity of advice on how to design a conceptual model, the need for a conceptual modelling framework is proposed. Built on the foundations laid in this paper, a conceptual modelling framework is described in the paper that follows.

Journal of the Operational Research Society (2008) **59,** 278–290. doi:10.1057/palgrave.jors.2602368
Published online 24 January 2007

Keywords: discrete-event simulation; model development; conceptual model; validation

Introduction

Conceptual modelling is the process of abstracting a model from a real or proposed system. It is almost certainly the most important aspect of a simulation project. The design of the model impacts all aspects of the study, in particular the data requirements, the speed with which the model can be developed, the validity of the model, the speed of experimentation and the confidence that is placed in the model results. A well-designed model significantly enhances the possibility that a simulation study will be a success.

Although effective conceptual modelling is a vital aspect of a simulation study, it is probably the most difficult and least understood (Law, 1991). There is surprisingly little written on the subject. It is difficult to find a book that devotes more than a handful of pages to the design of the conceptual model. Neither are there a plethora of research papers, with only a handful of well-regarded papers over the last four decades. A search through the academic tracks at major simulation conferences on discrete-event simulation reveals a host of papers on other aspects of simulation modelling. There are, however, very few papers that give any space to the subject of conceptual modelling.

The main reason for this lack of attention is probably due to the fact that conceptual modelling is more of an 'art' than a 'science' and therefore it is difficult to define methods and procedures. Whatever the reason, the result is that the art of conceptual modelling is largely learnt by experience. This somewhat *ad hoc* approach does not seem satisfactory for such an important part of the simulation modelling process.

This paper is the first of two papers that attempt to bring more clarity to the area of conceptual modelling for simulation. The issue is addressed first by defining the meaning of conceptual modelling and establishing the requirements of a conceptual model. These are the subjects of this paper. Having provided a foundation for conceptual model development, the paper that follows describes a framework for developing a conceptual model (Robinson, 2007).

The domain of interest for this discussion is primarily in the use of discrete-event simulation for modelling *operations systems* or *operating systems*. 'An operating system is a configuration of resources combined for the provision of goods or services' (Wild, 2002). Wild identifies four specific functions of operations systems: manufacture, transport, supply and service. This is one of the prime domains for simulation in operational research. We might refer to it as 'business oriented' simulation while interpreting business in its widest sense to include, for instance, the public sector and health. Models in this domain tend to be of a relatively small scale, with a project life-cycle of normally less than six months (Cochran *et al*, 1995). The models are generally developed by a lone modeller acting as an external or internal consultant. Sometimes the models are developed on a 'do-it-yourself' basis with a domain expert carrying out the development. This is somewhat different to the nature of simulation modelling in the military domain, another major application of simulation in operational research, where models tend to be of a much larger scale and where they are developed by teams of people (Robinson, 2002). Although the focus is on discrete-event

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

simulation for modelling operations systems, this is not to say that the concepts do not have wider applicability.

In this paper, the meaning of the term conceptual model is discussed in relation to existing definitions. A refined definition of a conceptual model is then given and the scope of conceptual modelling is defined. There is a pause for thought concerning the purpose of a conceptual model before a discussion on the requirements of a conceptual model. The paper finishes with a brief review of the guidance that is available for conceptual modelling. The prime contributions of this paper are to provide a definition of a conceptual model and to identify the requirements for a conceptual model.

Throughout the paper, three roles in a simulation study are assumed:

- *The Clients*: The problem owners and recipients of the results.
- *The Modeller*: The developer of the model.
- *Domain Experts*: Experts in the domain being modelled who provide data and information for the project.

These roles do not necessarily imply individual or separate people. There are often many clients and domain experts involved in a simulation study. In some situations one of the clients or domain experts may also act as the modeller.

Before exploring the meaning of conceptual modelling, let us begin with an example that highlights how more than one (conceptual) model can be developed of the same system.

Example: Modelling the Ford Motor Company South Wales engine assembly plant

I had been called in to carry out some simulation modelling of the new engine assembly plant that Ford Motor Company (Ford) was planning to build in South Wales. Faced with a meeting room full of engineers I started, as normally I would, by asking what was the problem that they wished to address. There was a unanimous response: ‘Scheduling! We are not sure that there is enough space by the line to hold sufficient stocks of the key components. Obviously the schedules we run on the key component production lines and on the main engine assembly line will affect the inventory we need to hold’. After further questioning it was clear that they saw this as the key issue. In their view, there was no problem with achieving the required throughput, especially because they had designed a number of similar lines previously.

The engine assembly line was planned to consist of three main assembly lines (with well over 100 operations), a Hot Test facility and a Final Dress area. Figure 1 provides a schematic of the line. On the first line (Line A), engine blocks are loaded onto platens (metal pallets on which engines move around the conveyor system) and then pass through a series of operations. On the Head Line various components are assembled to the head before the complete sub-assembly is joined with the engine block on Line A. On leaving Line A, the engine is loaded to a Line B platen to continue the assembly process. The empty Line A platen is washed and returned so a new engine block can be loaded. At the end of Line B, completed engines are off-loaded and move to the Hot Test facility. In Hot Test, engines are rigged to test machines, run for a few minutes and monitored. Engines that pass Hot Test move to the Final Dress area for completion. Engines that fail Hot Test are rectified and then completed.

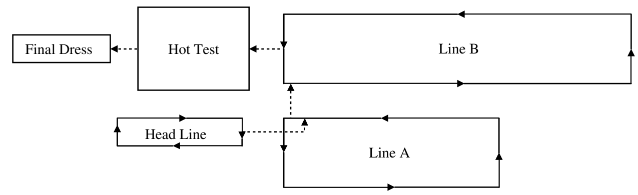

Figure 1 Schematic showing the layout of the South Wales engine assembly plant.

The majority of the operations on the three main assembly lines consist of a single automatic machine. Some operations require two parallel machines due to the length of the machine cycle, while a few other operations are performed manually. At various points along the line there are automatic test stations. When an engine fails the test, it is sent to an adjoining rework station, before returning to be tested again. All the operations are connected by a powered roller conveyor system.

The key components are the engine block, head, crankshaft, cam shaft and connecting rods. These are produced at nearby production facilities, delivered to the main assembly plant and stored line-side ready for assembly. Because various engine derivatives are made on the assembly line, a range of component derivatives needs to be produced and stored for assembly. The result was the concern over scheduling the production and the storage of these key components.

As with all such projects, time for developing and using the model was limited. It was important, therefore, to devise a model that could answer the questions about scheduling key components as quickly as possible while maintaining a satisfactory level of accuracy.

In considering the nature of the problem, it was clear that the key issue was not so much the rate at which engines progressed through the assembly line, but their sequence. The initial sequence of engines was determined by the production schedule, but this sequence was then disturbed by engines being taken out for rework and by the presence of parallel machines for some operations. Under normal operation the parallel machines would not cause a change in the sequence of engines on the line, but if one of the machines breaks down for a period, then the engines queuing for that machine would be delayed and their sequence altered.

It was recommended that the simulation model should represent in detail those elements that determined the sequence of engines on the main assembly line, that is, the schedule, the test and rework areas, and the parallel machines. All other operations could be simplified by grouping sections of the line that consisted of individual machines and representing them as a queue with a delay. The queue capacity needed to equate to the capacity of that section of the line. The delay needed to be equal to the time it took for an engine to pass through the section of the line, allowing for breakdowns. This would give a reasonable approximation to the rate at which engines would progress through the facility. Of course, the operations where the key components are assembled to the engine need to be modelled in detail, along with the line-side storage areas for those components.

Further to this, it was noted that detailed models of the key component production lines already existed. Alternative production schedules for each line could be modelled separately from the engine assembly line model and the output from these models stored. The outputs could then be read into the engine assembly line model as an input trace stating the component derivatives and their time of arrival at the assembly line. Some suitable delay needed to be added to allow for the transportation time between the key component lines and the main assembly line. It was also unnecessary to model the Hot Test and Final Dress, as all of the key components have been assembled prior to reaching these areas.

As a result of these simplifications, the model could be developed much more quickly and the final model ran much faster, enabling a greater amount of experimentation in the time available. The model fulfilled its objectives, sizing the line side storage areas and showing that shortages of key components were unlikely. What the model did suggest, however, was that there may be a problem with throughput.

Although the scheduling model indicated a potential problem with throughput, it did not contain enough detail to give accurate predictions of the throughput of the engine assembly line. As a result, a second model was developed with the objective of predicting and helping to improve the throughput of the facility. This model represented each operation in detail, but on this occasion did not represent the arrival and assembly of key components. It was assumed that the key components would always be available, as had been suggested by the scheduling model.

The second (throughput) model indeed confirmed that the throughput was likely to fall significantly short of that required by Ford and identified a number of issues that needed to be addressed. Over a period of time, by making changes to the facility and performing further simulation experiments, improvements were made such that the required throughput could be achieved.

This example demonstrates how two very different simulation models can be developed of the same system. But which model was the right one? The answer is both, since both answered the specific questions that were being asked of them. Underlying the differences between the models was the difference in the modelling objectives. Neither simulation model would have been useful for meeting the objectives of the other model. Of course, a single all encompassing model could have been developed, which could have answered both sets of questions. This, however, would have taken much longer to develop and it would certainly have run much slower, restricting the extent of the experimentation possible. Anyway, the need for the second model was only identified as a result of indications about throughput from the first model. Up to that point, a throughput model seemed unnecessary.

A more fundamental question that should be asked is if very different models can be developed of the same system, how can a modeller determine which model to use? Indeed, how can a modeller develop a model design, or a set of model designs from which to select? The only clue that comes from the example above is the importance of the modelling objectives in determining the nature of the model. Beyond this, modellers need some means for determining what to model. This process of taking a real-world situation and from it designing a model is often referred to as conceptual modelling.

What is conceptual modelling?

Conceptual modelling is about abstracting a model from a real or proposed system. All simulation models are simplifications of reality (Zeigler, 1976). The issue in conceptual modelling is to abstract an appropriate simplification of reality (Pidd, 2003). This provides some sense of what conceptual modelling is, but only in the most general of terms. How can the terms conceptual model and conceptual modelling be more precisely defined? Existing literature may shed some light on this topic.

In general, the notion of conceptual modelling, as expressed in the simulation and modelling literature, is vague and ill-defined, with varying interpretations as to its meaning. What seems to be agreed is that it refers to the early stages of a simulation study. This implies a sense of moving from the recognition of a problem situation to be addressed with a simulation model to a determination of what is going to be modelled and how. Balci (1994) breaks the early parts of a simulation study down into a number of processes: problem formulation, feasibility assessment of simulation, system and objectives definition, model formulation, model representation and programming. Which of these is specifically included in conceptual modelling is not identified. What is clear from Balci and other authors, for instance Willemain (1995), is that these early stages of a modelling study are not just visited once, but that they are continually returned to through a series of iterations in the life-cycle of a project. As such, conceptual modelling is not a one-off process, but one that is repeated and refined a number of times during a simulation study.

Zeigler (1976) sheds some light on the subject by identifying five elements in modelling and simulation from the 'real system' through to the 'computer' (the computer-based simulation model). In between is the 'experimental frame', 'base model' and 'lumped model'. The experimental frame is the limited set of circumstances under which the real system is observed, that is, specific input–output behaviours. The base model is a hypothetical complete explanation of the real system, which is capable of producing all possible input–output behaviours (experimental frames). The base model cannot be fully known since full knowledge of the real system cannot be attained. For instance, almost all systems involve some level of human interaction that will affect its performance. This interaction cannot be fully understood since it will vary from person-to-person and time-to-time.

In the lumped model, the components of a model are lumped together and simplified. The aim is to generate a model that is valid within the experimental frame, that is, reproduces the input–output behaviours with sufficient fidelity. The structure of the lumped model is fully known. Returning to the example of human interaction with a system, in a lumped model specific rules for interaction are devised for example a customer will not join a waiting line of more than ten people.

Nance (1994) separates the ideas of conceptual model and communicative model. The conceptual model exists in the mind of a modeller, the communicative model is an explicit representation of the conceptual model. He also specifies that the conceptual model is separate from model execution. In other words, the conceptual model is not concerned with how the computer-based model is coded. Fishwick (1995) takes a similar view, stating that a conceptual model is vague and ambiguous. It is then refined into a more concrete executable model. The process of model design is about developing and refining this vague and ambiguous model and creating the model code. In these terms, conceptual modelling is a subset of model design, which also includes the design of the model code.

The main debate about conceptual modelling and its definition has been held among military simulation modellers. Pace has lead the way in this debate and defines a conceptual model as 'a simulation developer's way of translating modelling requirements… into a detailed design framework…, from which the software that will make up the simulation can be built' (Pace, 1999, www.sisostds.org, accessed February 2006). In short, the conceptual model defines what is to be represented and how it is to be represented in the simulation. Pace sees conceptual modelling as being quite narrow in scope viewing objectives and requirements definition as precursors to the process of conceptual modelling. The conceptual model is largely independent of software design and implementation decisions. Pace (2000a, www.sisostds.org, accessed February 2006) identifies the information provided by a conceptual model as consisting of assumptions, algorithms, characteristics, relationships and data.

Lacy *et al* (2001, www.sisostds.org, accessed February 2006) further this discussion reporting on a meeting of the Defence Modelling and Simulation Office (DMSO) to try and reach a consensus on the definition of a conceptual model. The paper describes a plethora of views, but concludes by identifying two types of conceptual model. A *domain-oriented* model that provides a detailed representation of the problem domain and a *design-oriented* model that describes in detail the requirements of the model. The latter is used to design the model code. Meanwhile, Haddix (2001, www.sisostds.org, accessed February 2006) points out that there is some confusion over whether the conceptual model is an artefact of the user or the designer. This may, to some extent, be clarified by adopting the two definitions above.

The approach of military simulation modellers can be quite different to that of those working in business-oriented simulation (Robinson, 2002). Military simulations often entail large-scale models developed by teams of software developers. There is much interest in model reuse and distributed simulation, typified by the High Level Architecture (DMSO, 2005, http://www.dmso.mil/public/transition/hla/, accessed February 2006). Business-oriented simulations tend to be smaller in scale, involve lone modellers normally using a visual interactive modelling system (Pidd, 2004), and the models are often thrown-away on completion of a project. Interest in distributed simulation is moderate, mostly because the scale and life-time of the models does not warrant it (Robinson, 2005). As a result, although the definition and requirements for conceptual modelling may be similar in both these domains, some account must be made of the differences that exist.

In summary, the discussion above identifies some key facets of conceptual modelling and the definition of a conceptual model:

- Conceptual modelling is about moving from a problem situation, through model requirements to a definition of what is going to be modelled and how.
- Conceptual modelling is iterative and repetitive, with the model being continually revised throughout a modelling study.
- The conceptual model is a simplified representation of the real system.
- The conceptual model is independent of the model code or software (while model design includes both the conceptual model and the design of the code (Fishwick, 1995)).
- The perspective of the client and the modeller are both important in conceptual modelling.

It is clear, however, that complete agreement does not exist over these facets.

A definition of a conceptual model

Following the discussion above, Figure 2 defines a conceptual model as shown by the area within the dashed ellipse. It also places it within the wider context of a simulation study as defined in Robinson (2004). Figure 2 shows four key processes in the development and use of a simulation model: conceptual

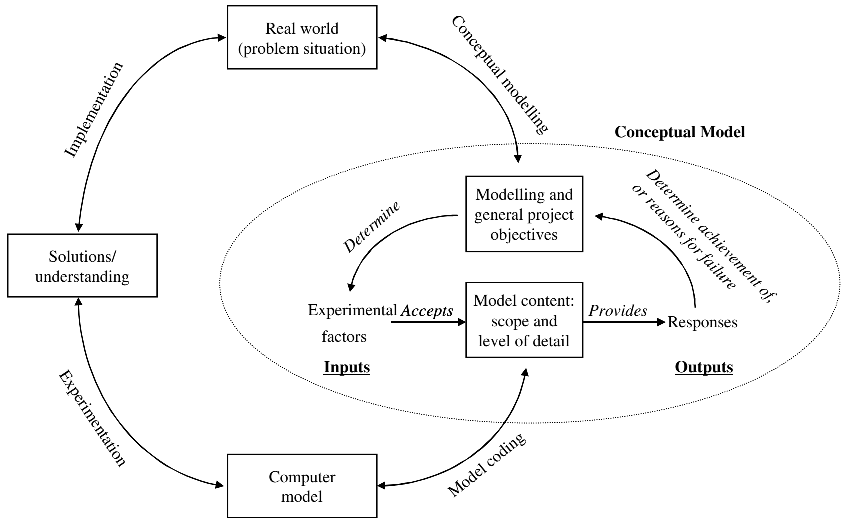

Figure 2 The conceptual model in the simulation project life-cycle (revised from Robinson, 2004).

modelling, model coding, experimentation and implementation. The outcome of each process is, respectively, a conceptual model, a computer model, solutions to the problem situation and/or a better understanding of the real world and improvements to the real world. The double arrows illustrate the iterative nature of the process and the circular diagram illustrates the potential to repeat the process of improvement through simulation a number of times. Missing from this diagram are the verification and validation activities involved in a simulation study. These are carried out in parallel with each of the four processes outlined in Figure 2. For a more detailed description of this life-cycle and model verification and validation, see Robinson (2004).

Based upon an understanding of the problem situation, which sits outside the conceptual model, the conceptual model is derived. This model is only a partial description of the real world, but it is sufficient to address the problem situation. The double arrow between the problem situation and objectives signifies the interplay between problem understanding and modelling. While the conceptual model reflects the understanding of the problem situation, the process of developing the conceptual model also changes the understanding of the problem situation. In particular, the nature of the questions that the modeller asks during conceptual modelling can lead to new insights on behalf of the clients and domain experts. At a greater extreme, ideas derived purely from conceptual modelling may be implemented in the real system, changing the actual nature of the problem situation.

The conceptual model itself consists of four main components: objectives, inputs (experimental factors), outputs (responses) and model content. Two types of *objective* inform a modelling project. First, there are the modelling objectives, which describe the purpose of the model and modelling project. Second, there are general project objectives which include the time-scales for the project and the nature of the model and its use (eg requirements for the flexibility of the model, run-speed, visual display, ease-of-use and model/component reuse). The definition of objectives is seen as intrinsic to decisions about the conceptual model. The Ford example above highlighted how different modelling objectives led to different models. Similarly, the general project objectives can affect the nature of the model. A shorter time-scale, for instance, may require a simpler conceptual model than would have been devised had more time been available. For this reason, the objectives are included in the definition of the conceptual model.

Including the modelling objectives as part of the definition of a conceptual model is at odds with Pace (1999). He sees the objectives and requirements definition as separate from the conceptual model. The author's view is that while understanding the problem situation and the aims of the organization lies within the domain of the real world (problem situation), the modelling objectives are specific to a particular model and modelling exercise. Different modelling objectives lead to different models within the same problem situation, as in the Ford example. As a result, the modelling objectives are intrinsic to the description of a conceptual model. Without the modelling objectives, the description of a conceptual model is incomplete.

The *inputs* (or experimental factors) are those elements of the model that can be altered to effect an improvement in, or better understanding of, the problem situation. They are determined by the objectives. Meanwhile, the *outputs* (or responses) report the results from a run of the simulation model. These have two purposes: first, to determine whether the modelling objectives have been achieved; second, to point to reasons why the objectives are not being achieved, if they are not.

Finally, the *model content* consists of the components that are represented in the model and their interconnections. The content can be split into two dimensions (Robinson, 1994):

- *The scope of the model*: The model boundary or the breadth of the real system that is to be included in the model.
- *The level of detail*: The detail to be included for each component in the model's scope.

The model content is determined, in part, by the inputs and outputs, in that the model must be able to accept and interpret the inputs and to provide the required outputs. The model content is also determined by the level of accuracy required. More accuracy generally requires a greater scope and level of detail.

While making decisions about the content of the model, various assumptions and simplifications are normally introduced. These are defined as follows:

- *Assumptions* are made either when there are uncertainties or beliefs about the real world being modelled.
- *Simplifications* are incorporated in the model to enable more rapid model development and use, and to improve transparency.

Assumptions and simplifications are identified as separate facets. Assumptions are ways of incorporating uncertainties and beliefs about the real world into the model. Simplifications are ways of reducing the complexity of the model. As such, assumptions are a facet of limited knowledge or presumptions, while simplifications are a facet of the desire to create simple models.

Based on these ideas a conceptual model is defined as follows:

> The conceptual model is a non-software specific description of the computer simulation model (that will be, is or has been developed), describing the objectives, inputs, outputs, content, assumptions and simplifications of the model.

This definition adds the point that the conceptual model is non-software specific in line with the views of the other authors described above. Considerations as to how the model code will be developed (whether it be a spreadsheet, specialist software or a programming language) should not dominate debate around the nature of the model that is required to address the problem situation. Conceptual modelling is about determining the right model, not how the software will be implemented.

In saying this, it must be recognized that many simulation modellers only have access to one or possibly two simulation tools. As a result, considerations of software implementation will naturally enter the debate about the nature of the conceptual model. This is recognized by the double arrow, signifying iteration, for the model coding process in Figure 2. What this definition for a conceptual model aims to highlight is the importance of separating as far as possible detailed model code considerations from decisions about the conceptual design.

The definition does not place the conceptual model at a specific point in time during a simulation study. This reflects the level of iteration that may exist in simulation work. A conceptual model may reflect a model that is to be developed, is being developed or has been developed in some software. The model is continually changing as the simulation study progresses. Whatever stage has been reached in a simulation study, the conceptual model is a non-software-specific description of the model as it is understood at that point in time. That said, the prime interest of this paper is in the role of the conceptual model during conceptual modelling, which implies it is describing a computer model that is yet to be developed, or at least the development is not yet complete.

Conceptual modelling defined

Put simply, conceptual modelling is the process of creating the conceptual model. Based on the definition given above this requires the following activities:

- understanding the problem situation (a precursor to conceptual modelling),
- determining the modelling and general project objectives,
- identifying the model outputs (responses),
- identify the model inputs (experimental factors),
- determining the model content (scope and level of detail), identifying any assumptions and simplifications.

These activities are explored in more detail in the paper that follows (Robinson, 2007). This suggests a general order in which the elements of a conceptual model might be determined. There is likely to be a lot of iteration forwards and backwards between these activities. Further to this, there is iteration between conceptual modelling and the rest of the process of model development and use (Robinson, 2004). Having said that the conceptual model is independent of the modelling software, it must be recognized that there is an interplay between the two. Since many modellers use the software that they are familiar with, it is possible (although not necessarily desirable) that methods of representation and limitations in the software will cause a revision to the conceptual model. Continued learning during model coding and experimentation may cause adjustments to the conceptual model as the understanding of the problem situation and modelling objectives change. Model validation activities may result in alterations to the conceptual model in order to improve the accuracy of the model. Availability, or otherwise, of data may require adjustments to the conceptual model. All this implies a great deal of iteration in the process of modelling and the requirement to continually revise the conceptual model. This iteration is illustrated by the double arrows between the stages in Figure 2.

The purpose of a conceptual model

In reflecting on the purpose of a conceptual model, one might question whether it is necessary to have one at all. Indeed, some might argue that the power of modern simulation software negates the need for conceptual modelling. Such software enables a modeller to move straight from developing an understanding of the problem situation to creating a computer model.

Albeit that this argument appears to have some credence, it ignores the fact that whatever practice a modeller might employ for developing the model code, decisions still have to be taken concerning the content and assumptions of the model. Modern simulation software does not reduce this level of decision making. What the software can provide is an environment for the more rapid development of the model code, enhancing the opportunities for iteration between conceptual modelling and model coding, and facilitating rapid prototyping. This does not negate the need for conceptual modelling, but simply aids the process of model design. It also highlights the point that conceptual modelling is not a one-off step, but part of a highly iterative process, particularly in relation to model coding.

Indeed, the power of modern software (and hardware) and the wider use of distributed processing may actually have increased the need for effective conceptual modelling. Salt (1993) and Chwif *et al* (2000) both identify the problem of the increasing complexity of simulation models; a result of the 'possibility' factor. People build more complex models because the hardware and software enable them to. While this may have extended the utility of simulation to problems that previously could not have been tackled, it also breads a tendency to develop overly complex models. There are various problems associated with such models including extended development times and onerous data requirements. This trend to develop ever more complex models has been particularly prevalent in the military domain (Lucas and McGunnigle, 2003). Indeed, it could be argued that there are some advantages in only having limited computing capacity; it forces the modeller to carefully design the model! As a result of the possibility factor it would seem that careful design of the conceptual model is more important than ever.

Beyond the general sense that careful model design is important, there are a number of reasons why a conceptual model is important to the development and use of simulation models. Pace (2003, www.sisostds.org, accessed February 2006) puts this succinctly by stating that the conceptual model provides a roadmap from the problem situation and objectives to model design and software implementation. He also recognizes that the conceptual model forms an important part of the documentation for a model. More specifically a well-documented conceptual model:

- Minimises the likelihood of incomplete, unclear, inconsistent and wrong requirements (Borah, 2002, www.sisostds.org, accessed February 2006; Pace, 2002).
- Helps build the credibility of the model.
- Guides the development of the computer model.
- Forms the basis for model verification and guides model validation.
- Guides experimentation by expressing the objectives, experimental factors and responses.
- Provides the basis of the model documentation.
- Can act as an aid to independent verification and validation when it is required.
- Helps determine the appropriateness of the model or its parts for model reuse and distributed simulation (Pace, 2000b).

Overall the conceptual model, if clearly expressed, provides a means of communication between all parties in a simulation study: the modeller, clients and domain experts (Pace, 2002). In so doing it helps to build a consensus, or least an accommodation, about the nature of the model and its use.

Requirements of a conceptual model

In designing a conceptual model it would be useful to have a set of requirements in mind. These could provide a basis against which to determine whether a conceptual model is appropriate. Indeed, Pritsker (1987) says that 'modelling is a difficult process because we do not have measurable criteria for evaluating the worth of a model'. In conceptual modelling, it may be difficult to identify a complete set of *measurable* criteria, since the model is purely descriptive at this stage. That said, a sense of requirements, even if they are more qualitative, would be helpful.

So what are the requirements for an effective conceptual model? This question is first answered by describing four main requirements after which the overarching need to keep the model as simple as possible is discussed.

Assessment criteria for models have been discussed by a number of authors, for instance, Gass and Joel (1981), Ören (1981, 1984), Robinson and Pidd (1998) and Balci (2001). The majority of this work is in the domain of large-scale military and public policy models; Robinson and Pidd is an exception. Furthermore, the criteria focus on assessing models that have been developed rather than on the assessment of conceptual models.

In terms of criteria for conceptual models in operational research there has been little reported. Willemain (1994), who investigates the preliminary stages of operational research interventions, briefly lists five qualities of an effective model: validity, usability, value to the clients, feasibility and aptness for the clients' problem. Meanwhile, Brooks and Tobias (1996a) identify eleven performance criteria for a good model. Requirements are also briefly discussed by Pritsker (1986), Henriksen (1988), Nance (1994), and van der Zee and van der Vorst (2005). Outside of operational research there are some discussions, for instance, Teeuw and van den Berg (1997, http://osm7.cs.byu.edu/ER97/workshop4/, accessed February

2006) who discuss the quality of conceptual models for business process reengineering.

Based on the discussions by simulation modellers and operational researchers, here it is proposed that there are four main requirements of a conceptual model: validity, credibility, utility and feasibility. Table 1 shows how the requirements discussed in the literature relate to these.

It is generally agreed that a valid model is one that is sufficiently accurate for the purpose at hand (Carson, 1986). However, since the notion of accuracy is of little meaning for a model that has no numeric output, conceptual model validity might be defined as:

> A perception, on behalf of the modeller, that the conceptual model can be developed into a computer model that is sufficiently accurate for the purpose at hand.

The phrase '... can be developed into a computer model...' is included in recognition that the conceptual model is a description of a model, not the computer model itself. Depending on the status of the simulation project, the conceptual model may be describing a computer model that will be developed, is being developed, or has been developed.

Underlying the notion of validity is the question of whether the model is 'right'. Note that this definition places conceptual model validity as a perception of the modeller. It also maintains the notion that a model is built for a specific purpose, which is common to most definitions of validity.

Credibility is similar to validity, but is taken from the perspective of the clients rather than the modeller. The credibility of the conceptual model is therefore defined as:

> A perception, on behalf of the clients, that the conceptual model can be developed into a computer model that is sufficiently accurate for the purpose at hand.

The clients must believe that the model is sufficiently accurate. Included in this concept is the need for the clients to be convinced that all the important components and relationships are in the model. Credibility also requires that the model and its results are understood by the clients. Would a model that could not be understood have credibility? An important factor in this respect is the transparency of the model which is discussed below.

Validity and credibility are seen as separate requirements because the modeller and clients may have very different perceptions of the same model. Although a modeller may be satisfied with a conceptual model, the clients may not be. It is not unusual for additional scope and detail to be added to a model, not because it improves its validity, but because it

Table 1 Requirements of a conceptual model related to those documented in the literature

	Documented requirements					
Proposed requirements	Pritsker (1986)	Henriksen (1988)	Nance (1994)	Willemain (1994)	Brooks and Tobias (1996a)	van der Zee and van der Vorst (2005)
Validity	Valid	Fidelity	Model correctness Testability	Validity Aptness for client's problem	Model describes behaviour of interest Accuracy of the model's results Probability of containing errors Validity Strength of theoretical basis of model	Completeness
Credibility	Understandable				Ease of understanding	Transparency
Utility	Extendible	Execution speed Ease of modification	Adaptability Reusability Maintainability	Value to client Usability	Portability and ease with which model can be combined with others	
Feasibility	Timely	Elegance		Feasibility	Time and cost to build model Time and cost to run model Time and cost to analyse results Hardware requirements	

improves its credibility. Not that adding scope and detail to gain credibility is necessarily a bad thing, but the modeller must ensure that this does not progress so far that the model becomes over complex. Simulation is particularly prone to such a drift through, for instance, the addition of non-vital graphics and the logic required to drive them.

The third concept, *utility*, is defined as:

> A perception, on behalf of the modeller and the clients, that the conceptual model can be developed into a computer model that is useful as an aid to decision-making within the specified context.

Utility is seen as a joint agreement between the modeller and the clients about the usefulness of the model. This notion moves beyond the question of whether the model is sufficiently accurate, to the question of whether the model is useful for the context of the simulation study. Utility includes issues such as ease-of-use, flexibility (ie ease with which model changes can be made), run-speed and visual display. Where the model, or a component of the model, might be used again on the same or another study, reusability would also be subsumed within the concept of utility. The requirements for utility are expressed through the general project objectives.

Within any context a range of conceptual models could be derived. The accuracy of these models would vary, but some or all might be seen as sufficiently accurate and, hence, under the definitions given above, they would be described as valid and credible. This does not necessarily mean that the models are useful. For instance, if a proposed model is large and cumbersome, it may have limited utility due to reduced ease-of-use and flexibility. Indeed, a less accurate (but still sufficiently accurate), more flexible model that runs faster may have greater utility by enabling a wider range of experimentation within a time-frame.

Hodges (1991) provides an interesting discussion around model utility and suggests that a ‘bad’ model (one that is not sufficiently accurate) can still be useful. He goes on to identify specific uses for such models. Bankes (1993) continues with this theme, discussing the idea of inaccurate models for exploratory use, while Robinson (2001) sees a role for such models in facilitating learning about a problem situation.

The final requirement, *feasibility*, is defined as follows:

> A perception, on behalf of the modeller and the clients, that the conceptual model can be developed into a computer model with the time, resource and data available.

A range of factors could make a model infeasible: it might not be possible to build the proposed model in the time available, the data requirements may be too onerous, there may be insufficient knowledge of the real system, and the modeller may have insufficient skill to code the model. Feasibility implies that the time, resource and data are available to enable development of the computer model.

The four requirements described above are not mutually exclusive. For instance, the modeller’s and clients’ perspectives on model accuracy are likely to be closely aligned, although not always. An infeasible model could not generally be described as a useful model, although a conceptual model that is infeasible could be useful for aiding problem understanding. Albeit that these concepts are related, it is still useful to identify them as four separate requirements so a modeller can be cognisant of them when designing the conceptual model.

The overarching requirement: keep the model simple

The overarching requirement is the need to avoid the development of an overly complex model. In general, the aim should be:

> to keep the model as simple as possible to meet the objectives of the simulation study (Robinson, 2004).

There are a number of advantages with simple models (Innis and Rexstad, 1983; Ward, 1989; Salt, 1993; Chwif *et al*, 2000; Lucas and McGunnigle, 2003; Thomas and Charpentier, 2005):

- simple models can be developed faster,
- simple models are more flexible,
- simple models require less data,
- simple models run faster,
- the results are easier to interpret since the structure of the model is better understood.

With more complex models these advantages are generally lost. Indeed, at the centre of good modelling practice is the idea of resorting to simplest explanation possible. Occam’s razor puts this succinctly, ‘*plurality should not be posited without necessity*’ (William of Occam) (quoted from Pidd, 2003), as does Antoine de Saint-Exupery who reputedly said that ‘*perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away*’.

The requirement for simple models does not negate the need to build complex models on some occasions. Indeed, complex models are sometimes required to achieve the modelling objectives. The requirement is to build the simplest model possible, not simple models *per se*. What should be avoided, however, is the tendency to try and model every aspect of a system when a far simpler more focused model would suffice.

The graph in Figure 3 illustrates the notional relationship between model accuracy and complexity (Robinson, 1994). Increasing levels of complexity (scope and level of detail) improve the accuracy of the model, but with diminishing returns. Beyond point x there is little to be gained by adding to the complexity of the model. A 100% accurate model will never be achieved because it is impossible to know everything about the real system. The graph illustrates a further point. Increasing the complexity of the model too far, may lead to a less accurate model. This is because the data and information

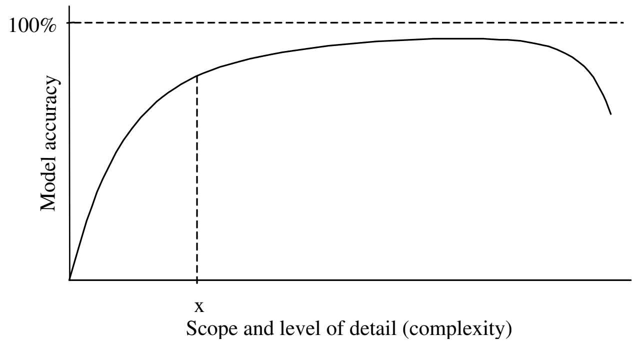

Figure 3 Simulation model complexity and accuracy (based on Robinson, 1994).

are not available to support such a detailed model. For instance, it is unlikely that we could accurately model the exact behaviour of individuals in a queue, and attempts to do so, beyond very simple rules, may lead to a less accurate result.

Ward (1989) provides a lucid account on the simplicity of models. In doing so, he makes a useful distinction between constructive simplicity and transparency. Transparency is an attribute of the client (how well he/she understands the model), while constructive simplicity is an attribute of the model itself (the simplicity of the model). Because transparency is an attribute of the client, it depends on his/her level of knowledge and skill. A model that is transparent to one client may not be transparent to another. In developing a conceptual model, the modeller must consider transparency as well as simplicity, designing the model with the particular needs of the client in mind. The need for transparency is, of course, confounded by the presence of multiple clients (as is the case in many simulation studies), all of whom must be satisfied with the model. These ideas closely link to the requirement for credibility, as discussed above, since a model that is not transparent is unlikely to have credibility.

Having emphasized the importance of simplicity, there are those that warn against taking this to an extreme. Pritsker (1986) reflects on his experience of developing models of differing complexity of the same system. He concludes that the simplest model is not always best because models need to be able to evolve as the requirements change. The simplest model is not always the easiest to embellish. Schruben and Yücesan (1993) make a similar point, stating that simpler models are not always as easy to understand, code and debug. Davies *et al* (2003) point out that simpler models require more extensive assumptions about how a system works and that there is a danger in setting the system boundary (scope) too narrow in case an important facet is missed.

Guidance on conceptual modelling

Exhortations to develop simple models highlight an important consideration in designing a conceptual model. Modelling requirements provide a guide as to whether a conceptual model is appropriate. Neither, however, describes how a modeller might go about determining what the conceptual model should be in a simulation study. So what help is offered in the simulation and modelling literature to guide modellers in designing the conceptual model?

First, it is worth recognizing that conceptual modelling requires creativity (Henriksen, 1989). Simulation modelling is both art and science (Shannon, 1975) with conceptual modelling lying more at the artistic end! As Schmeiser (2001) points out: 'While abstracting a model from the real world is very much an art, with many ways to err as well as to be correct, analysis of the model is more of a science, and therefore easier, both to teach and to do'. The need for creativity does not, however, excuse the need for guidelines on how to model (Evans, 1992). Ferguson *et al* (1997), writing about software development, point out that in 'most professions, competent work requires the disciplined use of established practices. It is not a matter of creativity versus discipline, but one of bringing discipline to the work so creativity can happen'.

In searching the modelling literature for advice from simulation modellers and operational researchers on how to develop models, three basic approaches can be found: principles of modelling, methods of simplification and modelling frameworks.

Principles of modelling

Providing a set of guiding principles for modelling is one approach to advising simulation modellers on how to develop (conceptual) models. For instance, Pidd (1999) describes six principles of modelling:

- model simple; think complicated,
- be parsimonious; start small and add,
- divide and conquer; avoid megamodels,
- use metaphors, analogies, and similarities,
- do not fall in love with data,
- modelling may feel like muddling through.

The central theme is one of aiming for simple models through evolutionary development. Others have produced similar sets of principles (or guidelines), for instance, Morris (1967), Musselman (1992), Powell (1995), Pritsker (1998) and Law and Kelton (2000). The specific idea of evolutionary model development is further explored by Nydick *et al* (2002).

These principles provide some useful guidance for those developing conceptual models. It is useful to encourage modellers to start with small models and to gradually add scope and detail. What such principles do not do, however, is to guide a modeller through the conceptual modelling process. When should more detail be added? When should elaboration stop? There is a difference between giving some general principles and guiding someone through a process.

Methods of simplification

Simplification entails removing scope and detail from a model or representing components more simply while maintaining a sufficient level of accuracy. In Zeigler's (1976) terms this could be described as further lumping of the lumped model. This is the opposite of the start small and add principle.

There are quite a number of discussions on simplification, both in the simulation and the wider modelling context. Morris (1967) identifies some methods for simplifying models: making variables into constants, eliminating variables, using linear relations, strengthening the assumptions and restrictions, and reducing randomness. Ward (1989) provides a similar list of ideas for simplification. Meanwhile, Courtois (1985) identifies criteria for the successful decomposition of models in engineering and science.

For simulation modelling, Zeigler (1976) suggests four methods of simplification: dropping unimportant components of the model, using random variables to depict parts of the model, coarsening the range of variables in the model, and grouping components of the model. There is an apparent contradiction between Morris' and Zeigler's advice in that the former suggests reducing randomness, while the latter suggests increasing it by representing sections of the model with random variables. This difference in opinion can be reconciled by recognizing that simplification methods are sensitive to the modelling approach that is being applied. Morris is concentrating more on mathematical algorithms where the inclusion of randomness is less convenient. Zeigler is writing about simulation specifically, where complex behaviours can sometimes be reduced to a single random variable.

Yin and Zhou (1989) build upon Zeigler's ideas, discussing six simplification techniques and presenting a case study. Sevinc (1990) provides a semiautomatic procedure based on Zeigler's ideas. Innis and Rexstad (1983) enter into a detailed discussion about how an existing model might be simplified. They provide a list of 17 such methods, although they do not claim that these are exhaustive. They conclude by suggesting that managers should be provided with both a full and a simplified simulation model. There is a sense in which the Ford example followed this approach, with one model being more detailed than the other, although neither could be described as a 'full' model. Robinson (1994) also lists some methods for simplifying simulation models. Finally, Webster *et al* (1984) describe how they selected an appropriate level of detail for generating samples in a timber harvesting simulation model.

Such ideas are useful for simplifying an existing (conceptual) model, but they do not guide the modeller over how to bring a model into existence. Model simplification acts primarily as a redesign tool and not a design tool.

Modelling frameworks

A modelling framework goes beyond the idea of guiding principles and methods of model simplification by providing a specific set of steps that guide a modeller through development of a conceptual model. There have been some attempts to provide such frameworks going back to Shannon (1975) who describes four steps: specification of the model's purpose; specification of the model's components; specification of the parameters and variables associated with the components; and specification of the relationships between the components, parameters and variables.

Both Nance and Pace have devised frameworks which relate primarily to the development of large-scale models in the military domain. Nance (1994) outlines the conical methodology. This is an object oriented, hierarchical specification language which develops the model definition (scope) top-down and the model specification (level of detail) bottom-up. A series of modelling steps are outlined. Balci and Nance (1985) focus specifically on a procedure for problem formulation. Meanwhile, Nance and Arthur (2006) identify the potential to adopt software requirements engineering (SRE) approaches for simulation model development. They also note that there is little evidence of SRE actually being adopted by simulation modellers.

Pace (1999, 2000a) explores a four-stage approach to conceptual model development, similar to that of Shannon: collect authoritative information on the problem domain; identify entities and processes that need to be represented; identify simulation elements; and identify relationships between the simulation elements. He also identifies six criteria for determining which elements to include in the conceptual model. These criteria focus on the correspondence between real world items and simulation objects (Pace, 2000a, p. 8).

Within our domain of interest, simulation for modelling operations systems, there is quite limited work on conceptual modelling frameworks. Brooks and Tobias (1996b) briefly propose a framework for conceptual modelling, but go no further in expanding upon the idea. Recent papers by Guru and Savory (2004) and van der Zee and van der Vorst (2005) propose conceptual modelling frameworks in some more detail. Guru and Savory propose a set of modelling templates (tables) useful for modelling physical security systems. Meanwhile, van der Zee and van der Vorst propose a framework for supply chain simulation. Both are aimed at an object-oriented implementation of the computer-based simulation model. Meanwhile, Kotiadis (2006) looks to the ideas of Soft Operational Research, and specifically soft systems methodology (SSM) (Checkland, 1981), for aiding the conceptual modelling process. She uses SSM to help understand a complex health care system and then derives the simulation conceptual model from the SSM 'purposeful activity model'.

Conclusion

There is, in large measure, a vacuum of research in the area of conceptual modelling for discrete-event simulation. Albeit that many simulation researchers consider effective conceptual modelling to be vital to the success of a simulation study, there have been few attempts to develop definitions and approaches that are helpful to the development of conceptual models. The discussion above attempts to redress this balance by offering a definition of a conceptual model and outlining the requirements for a conceptual model. The conceptual model definition is useful for providing a sense of direction to simulation modellers during a simulation study. If they do not know what they are heading for, how can they head for it? The requirements provide a means for determining the appropriateness of a conceptual model both during and after development. For researchers, the definition and requirements provide a common foundation for further research in conceptual modelling.

What the definition and requirements do not provide is a sense of how to develop a conceptual model. Three approaches have been used in this respect: principles of modelling, methods of simplification and modelling frameworks. The latter has potential to provide the most specific guidance on how to develop a conceptual model. It is also the area that has seen the least development of the three, particularly in simulation for operations systems. It is to a framework for conceptual modelling that our attention turns next. In the paper that follows (Robinson, 2007), a framework is described that is built upon the foundations laid out here. The framework is illustrated by applying it to the Ford engine assembly plant model.

Acknowledgements—Some sections of this paper are based on:

- Robinson S (2004). *Simulation*: *The Practice of Model Development and Use*. Wiley: Chichester, UK
- Robinson S (2004). Designing the conceptual model in simulation studies. In: Brailsford SC, Oakshott L, Robinson S and Taylor SJE (eds). *Proceedings of the 2004 Operational Research Society Simulation Workshop* (*SW*04). Operational Research Society: Birmingham, UK, pp 259–266.
- Robinson S (2006). Issues in conceptual modelling for simulation: Setting a research agenda. In: Garnett J, Brailsford S, Robinson S and Taylor S (eds). *Proceedings of the Third Operational Research Society Simulation Workshop (SW06)*. The Operational Research Society: Birmingham, UK, pp 165–174.
- Robinson S (2006). Conceptual modeling for simulation: Issues and research requirements. In: Perrone LF, Wieland FP, Liu J, Lawson BG, Nicol DM and Fujimoto RM (eds). *Proceedings of the 2006 Winter Simulation Conference*. IEEE: Piscataway, NJ, pp 792–800.

The Ford engine plant example is used with the permission of John Ladbrook, Ford Motor Company.

References

Balci O (1994). Validation, verification, and testing techniques throughout the life cycle of a simulation study. *Ann Opns Res* **53**: 121–173.

Balci O (2001). A methodology for certification of modeling and simulation applications. *ACM Trans Model Comput Simul* **11**: 352–377.

Balci O and Nance RE (1985). Formulated problem verification as an explicit requirement of model credibility. *Simulation* **45**(2): 76–86.

Bankes S (1993). Exploratory modeling for policy analysis. *Opns Res* **41**: 435–449.

Borah JJ (2002). Conceptual modeling—The missing link of simulation development. *Proceedings of the 2002 Spring Simulation Interoperability Workshop*. Available online.

Brooks RJ and Tobias AM (1996a). Choosing the best model: Level of detail, complexity and model performance. *Math Comput Model* **24**(4): 1–14.

Brooks RJ and Tobias AM (1996b). A framework for choosing the best model structure in mathematical and computer modeling. *Proceedings of the 6th Annual Conference AI, Simulation, and Planning in High Autonomy Systems*, University of Arizona, pp 53–60.

Carson JS (1986). Convincing users of model's validity is challenging aspect of modeler's job. *Ind Eng* **18**(6): 74–85.

Checkland PB (1981). *Systems Thinking, Systems Practice*. Wiley: Chichester, UK.

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.

Cochran JK, Mackulak GT and Savory PA (1995). Simulation project characteristics in industrial settings. *Interfaces* **25**: 104–113.

Courtois PJ (1985). On time and space decomposition of complex structures. *Commun ACM* **28**: 590–603.

Davies R, Roderick P and Raftery J (2003). The evaluation of disease prevention and treatment using simulation models. *Eur J Opl Res* **150**: 53–66.

DMSO (2005). Defence modeling and simulation office, HLA Page. Available online.

Evans JR (1992). Creativity in MS/OR: Improving problem solving through creative thinking. *Interfaces* **22**(2): 87–91.

Ferguson P *et al* (1997). Results of applying the personal software process. *Computer* **5**: 24–31.

Fishwick PA (1995). *Simulation Model Design and Execution: Building Digital Worlds*. Prentice-Hall: Upper Saddle River, NJ.

Gass SI and Joel LS (1981). Concepts of model confidence. *Comput Opns Res* **8**: 341–346.

Guru A and Savory P (2004). A template-based conceptual modeling infrastructure for simulation of physical security systems. In: Ingalls RG, Rossetti MD, Smith JS and Peters BA (eds). *Proceedings of the 2004 Winter Simulation Conference*. IEEE: Piscataway: NJ, pp 866–873.

Haddix F (2001). Conceptual modeling revisited: A developmental model approach for modeling and simulation. *Proceedings of the 2001 Fall Simulation Interoperability Workshop*. Available online.

Henriksen JO (1988). One system, several perspectives, many models. In: Abrams M, Haigh P and Comfort J (eds). *Proceedings of the 1988 Winter Simulation Conference*. IEEE: Piscataway: NJ, pp 352–356.

Henriksen JO (1989). Alternative modeling perspectives: Finding the creative spark. In: MacNair EA, Musselman KJ and Heidelberger P (eds). *Proceedings of the 1989 Winter Simulation Conference*. IEE: Piscataway: NJ, pp 648–652.

Hodges JS (1991). Six (or so) things you can do with a bad model. *Opns Res* **39**: 355–365.

Innis G and Rexstad E (1983). Simulation model simplification techniques. *Simulation* **41**(1): 7–15.

Kotiadis K (2006). Extracting a conceptual model for a complex integrated system in healthcare. In: Garnett J, Brailsford S, Robinson S and Taylor S (eds). *Proceedings of the Third Operational Research Society Simulation Workshop (SW06)*. The Operational Research Society: Birmingham: UK, pp 235–245.

Lacy LW *et al* (2001). Developing a consensus perspective on conceptual models for simulation systems. *Proceedings of the 2001 Spring Simulation Interoperability Workshop*. Available online.

Law AM (1991). Simulation model's level of detail determines effectiveness. *Ind Eng* **23**(10): 16–18.

Law AM and Kelton WD (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill: New York.

Lucas TW and McGunnigle JE (2003). When is model complexity too much? Illustrating the benefits of simple models with Hughes' salvo equations. *Naval Res Logist* **50**: 197–217.

Morris WT (1967). On the art of modeling. *Mngt Sci* **13**: B707–B717.

Musselman KJ (1992). Conducting a successful simulation project. In: Swain JJ, Goldsman D, Crain RC and Wilson JR (eds). *Proceedings of the 1992 Winter Simulation Conference*. IEEE: Piscataway: NJ, pp 115–121.

Nance RE (1994). The conical methodology and the evolution of simulation model development. *Ann Opns Res* **53**: 1–45.

Nance RE and Arthur JD (2006). Software requirements engineering: Exploring the role in simulation model development. In: Garnett J, Brailsford S, Robinson S and Taylor S (eds). *Proceedings of the Third Operational Research Society Simulation Workshop (SW06)*. The Operational Research Society: Birmingham: UK, pp 117–127.

Nydick RL, Liberatore MJ and Chung QB (2002). Modeling by elaboration: An application to visual process simulation. *INFOR* **40**: 347–361.

Ören TI (1981). Concepts and criteria to assess acceptability of simulation studies: A frame of reference. *Commun ACM* **28**: 190–201.

Ören TI (1984). Quality assurance in modeling and simulation: A taxonomy. In: Ören TI, Zeigler BP and Elzas MS (eds). *Simulation and Model-based Methodologies: An Integrative Approach*. Springer-Verlag: Heidelberg: Germany, pp 477–517.

Pace DK (1999). Development and documentation of a simulation conceptual model. *Proceedings of the 1999 Fall Simulation Interoperability Workshop*. Available online.

Pace DK (2000a). Simulation conceptual model development. *Proceedings of the 2000 Spring Simulation Interoperability Workshop*. Available online.

Pace DK (2000b). Ideas about simulation conceptual model development. *Johns Hopkins APL Tech Dig* **21**: 327–336.

Pace DK (2002). The value of a quality simulation conceptual model. *Model Simul Mag* **1**(1): 9–10.

Pace DK (2003). Thoughts about the simulation conceptual model. *Proceedings of the 2003 Spring Simulation Interoperability Workshop*. Available online.

Pidd M (1999). Just modeling through: A rough guide to modeling. *Interfaces* **29**: 118–132.

Pidd M (2003). *Tools for Thinking: Modelling in Management Science*, 2nd edition. Wiley: Chichester, UK.

Pidd M (2004). *Computer Simulation in Management Science*, 5th edition. Wiley: Chichester, UK.

Powell SG (1995). Six key modeling heuristics. *Interfaces* **25**: 114–125.

Pritsker AAB (1986). Model evolution: A rotary table case history. In: Wilson J, Henriksen J and Roberts S (eds). *Proceedings of the 1986 Winter Simulation Conference*. IEEE: Piscataway: NJ, pp 703–707.

Pritsker AAB (1987). Model evolution II: An FMS design problem. In: Thesen A, Grant H and Kelton WD (eds). *Proceedings of the 1987 Winter Simulation Conference*. IEEE: Piscataway: NJ, pp 567–574.

Pritsker AAB (1998). Principles of simulation modeling. In: Banks J (ed). *Handbook of Simulation*. Wiley: New York, pp 31–51.

Robinson S (1994). Simulation projects: Building the right conceptual model. *Ind Eng* **26**(9): 34–36.

Robinson S (2001). Soft with a hard centre: Discrete-event simulation in facilitation. *J Opl Res Soc* **52**: 905–915.

Robinson S (2002). Modes of simulation practice: Approaches to business and military simulation. *Simul Pract Theory* **10**: 513–523.

Robinson S (2004). *Simulation: The Practice of Model Development and Use*. Wiley: Chichester, UK.

Robinson S (2005). Distributed simulation and simulation practice. *Simulation: Trans Soc Model Comput Simul* **81**: 5–13.

Robinson S (2007). Conceptual modelling for simulation Part II: A framework for conceptual modelling. *J Opl Res Soc*, doi:10.1057/palgrave.jors.2602369.

Robinson S and Pidd M (1998). Provider and customer expectations of successful simulation projects. *J Opl Res Soc* **49**: 200–209.

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.

Schmeiser BW (2001). Some myths and common errors in simulation experiments. In: Peters BA, Smith JS, Medeiros DJ and Rohrer MW (eds). *Proceedings of the 2001 Winter Simulation Conference*. IEEE: Piscataway: NJ, pp 39–46.

Schruben L and Yücesan E (1993). Complexity of simulation models: A graph theoretic approach. In: Evans GW, Mollaghasemi M, Russell EC and Biles WE (eds). *Proceedings of the 1993 Winter Simulation Conference*. IEEE: Piscataway: NJ, pp 641–649.

Sevinc S (1990). Automation of simplification in discrete event modeling and simulation. *Int J Gen Syst* **18**: 125–142.

Shannon RE (1975). *Systems simulation: The Art and Science*. Prentice-Hall: Englewood Cliffs, NJ.

Teeuw WB and van den Berg H (1997). On the quality of conceptual models. In: Liddle SW (ed). *Proceedings of the ER'97 Workshop on Behavioral Models and Design Transformations*: *Issues and Opportunities in Conceptual Modeling*. Available online.

Thomas A and Charpentier P (2005). Reducing simulation models for scheduling manufacturing facilities. *Eur J Opl Res* **161**: 111–125.

van der Zee DJ and van der Vorst JGAJ (2005). A modeling framework for supply chain simulation: Opportunities for improved decision making. *Decis Sci* **36**: 65–95.

Ward SC (1989). Arguments for constructively simple models. *J Opl Res Soc* **40**: 141–153.

Webster DB, Padgett ML, Hines GS and Sirois DL (1984). Determining the level of detail in a simulation model—A case study. *Comput Ind Eng* **8**: 215–225.

Wild R (2002). *Operations Management*, 6th edition. Continuum: London.

Willemain TR (1994). Insights on modeling from a dozen experts. *Opns Res* **42**: 213–222.

Willemain TR (1995). Model formulation: What experts think about and when. *Opns Res* **43**: 916–932.

Yin HY and Zhou ZN (1989). Simplification techniques of simulation models. *Proceedings of Beijing International Conference on System Simulation and Scientific Computing*, IEEE: Piscataway: NJ, pp 782–786.

Zeigler BP (1976). *Theory of Modeling and Simulation*. Wiley: New York.

Received April 2006;
accepted November 2006 after one revision