Conceptual Modelling: Who Needs It?

Stewart Robinson
Warwick Business School, University of Warwick, Coventry, CV4 7AL, UNITED KINGDOM

Introduction

In this article we explore the meaning and need for conceptual modelling. As a modelling and simulation professional, or even as an interested outsider, you might wonder if this is something that is relevant to you at all. Having read this article, I hope that you will be convinced that conceptual modelling is something we all have an interest in. Indeed, it is something we are all already doing whether we know it or not.

Conceptual modelling is the abstraction of a simulation model from the part of the real world it is representing ('the real system'). The real system may, or may not, currently exist. Abstraction implies the need for simplification of the real system and for assumptions about what is not known about the real system. In other words, all simulation models are simplifications of the real world. The secret to good conceptual modelling is to get the level of simplification correct, that is, to abstract at the right level.

Before exploring the underlying ideas behind conceptual modelling further, let us start with a practical illustration.

Modelling an Outpatients Clinic

Our simulation and modelling team was recently asked to develop a simulation model for a hospital in Birmingham, UK. The hospital is investing in a new outpatients building, a multi-million dollar project, and their key question was how many consultation rooms are required? They had performed some calculations based on expected patient flows and on observations of the current outpatients system. However, there was obviously some concern with making major investment decisions based on these limited data.

We were quick to point out the problems of making calculations based on static data which do not take into account the effects of variability in patient flows and consultation times. This is something for which discrete-event simulation is very well suited.

When asked to build a model such as this, the typical approach would be to start collecting data and to develop a detailed model of the system. However, the more we investigated how an outpatients system works the more we realised just how complex the system is. There are many specialties using the facility, each with its own clinical team. Patients can progress through a series of tests and consultations. For some specialties, such as ophthalmology, specialist equipment and dedicated rooms are required. Scheduling patient appointments is a significant task and then there is the matter of late arrivals and non-attendances. Staff shifts, working practices and skills all impact upon the functioning of the system.

Given appropriate data, it would be quite possible to build a simulation model that took account of all these details. There were, however, two issues that made such a model infeasible:

- *Lack of data*: much of the necessary data had not previously been collected and even if we were to try, issues of patient confidentiality (e.g. you cannot sit in a consultation room timing consultation times) would make it impossible to collect all the data we needed.
- *Lack of time*: the hospital required an answer within a few weeks and we had

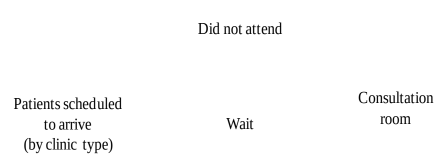
very limited time and resource to devote to the modelling work given the number of parallel activities in which we were engaged.

So what did we do? We focused on the critical issue of how many rooms were required and designed a simple model that would give at least an indication upon which the hospital managers could base a decision. Our world view was that the additional information a basic simulation could offer would be more beneficial than no simulation at all.

The simple model we constructed took a couple of days to build and experiment with. It provided a lower bound on the rooms required. In doing so it provided information that would give a greater level of confidence in making the decision that the hospital faced. This was all that was possible given the data and resource available, but we believed it was still valuable.

The model we designed is outlined in figure 1. Patient arrivals were based on the busiest period of the week – a Monday morning. All patients scheduled to arrive for each clinic, on a typical Monday, arrived into the model at the start of the simulation run, that is, 9.00am. For this model we were not concerned with waiting time, so it was not necessary to model when exactly a patient arrived, only the number that arrived.

Figure 1 *Simple Outpatients Building Model*



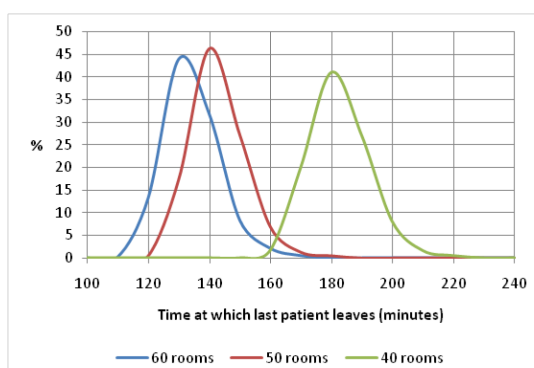
A proportion of patients do not attend their allotted clinic. Typical proportions of patients that do not attend were sampled at the start of the simulation run and these were removed before entering the waiting line.

Data on the time in a consultation room were limited, since they had not specifically been timed, but there were norms to which the clinical staff aimed to work. These data were available by clinic type and we used these as the mean of an Erlang-3 distribution to give an approximation for the variability in consultation time.

The input variable for the simulation experiments was the number of consultation rooms, which were varied from 20 to 60 in steps of 10. The main output variable was the time it took until the last patient left the system. A key simplification, which all involved recognised, was that there were no limitations on staff or equipment availability. Albeit extremely unlikely that this would be the case, the model was predicting a lower bound on the rooms required. In other words, shortages of staff and equipment would only increase the need for consultation rooms with patients waiting in the rooms while the resource became available.

For each room scenario the model was replicated 1000 times and a frequency chart was generated showing the probability that the system would be cleared in under 3 hours – the hospital's target. Figure 2 shows an example of these results.

Figure 2 *Example of Results from the Outpatients Building Model: Frequency Distributions for Time until Last Patient Leaves*



This example illustrates the very essence of conceptual modelling; abstracting a model from the real system. In this case, the real system was not in existence, but it was a proposed system. The model involved simplifications such as modelling only Monday morning's clinic and not modelling staff and equipment. It also involved assumptions about, among others, the consultation times. Because of the constraints on data and time, the conceptual model involved a great deal of simplification; as such, it might be described as a 'far abstraction.'

Whether we got the conceptual model right is in large measure a matter of opinion and one we will leave the reader to judge. I am sure that readers will form quite different judgements on the validity of our model.

What is Conceptual Modelling?

Conceptual modelling is about abstracting a model that is fit-for-purpose and by this we mean a model that is valid, credible, feasible and useful. So, in designing the outpatients building model we had to consider whether the model would:

- Produce sufficiently accurate results for the purpose: understanding the number of rooms required in the building (*validity*).
- Be believed by the clients (*credibility*).
- Be *feasible* to build within the constraints of the available data and time.
- Be *useful*, that is, sufficiently easy to use, flexible, visual and quick to run.

Because all models are simplifications of the real world, all modelling involves conceptual modelling. Even the most complex and detailed simulation still makes various assumptions about the real world and chooses to ignore certain details.

More formally we define a conceptual model as follows:

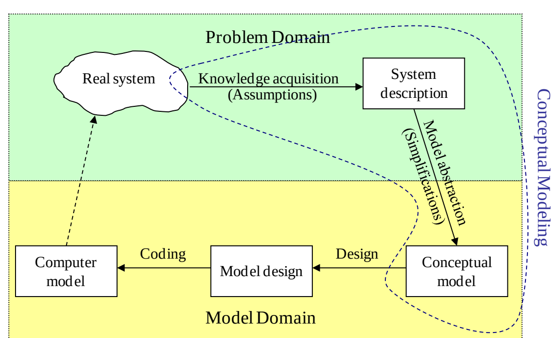
> '… a non-software specific description of the computer simulation model (that will be, is or has been developed), describing the objectives, inputs, outputs, content, assumptions and simplifications of the model.' (Robinson, 2008)

Let us explore this definition in some more detail.

First, this definition highlights the separation of the conceptual model from the computer model. The latter is software specific, that is, it represents the conceptual model in a specific computer code. The conceptual model is not specific to the software in which it is developed. It forms the foundation for developing the computer code.

Figure 3 aims to illustrate this separation further. The conceptual model is the bridge between the real system (problem domain) and the model, which is a simplified representation of the real system (model domain). Through knowledge acquisition we gain an understanding of the real system and through model abstraction we design a conceptual model. The system description includes all that we know about the real system, not that this knowledge can ever be complete. Based on our knowledge of the real system, we then abstract our simplified model.

Figure 3 *Artefacts of Conceptual Modelling (Robinson, 2010)*



Up to this point there has been no reference to the software in which the simulation will be developed. The model design, a software specific description of the model, is developed from the conceptual model and out of this the code for the computer model is created. (Note that I interpret computer coding very broadly from the use of a simple spreadsheet, through specialist simulation packages, to the use of general purpose programming languages.)

For those who are familiar with Zeigler's (1976) theory of modelling and simulation, the artefacts described in figure 3 bear close resemblance to his concepts of:

- *The real system*
- *The experimental frame*: the limited set of circumstances under which the real system is observed or experimented with (i.e. specific input-output behaviors).
- *The base model*: a model that is capable of accounting for all the input-output behavior of the real system (it cannot be fully known).
- *The lumped model*: a simplified model with components lumped together and interactions simplified (it can be fully known and is valid within a chosen experimental frame).
- *The computer*: the computational device for implementing the lumped model.

Returning to our definition of a conceptual model, it is stated that the description is of a computer simulation model that 'that will be, is or has been developed.' This serves to highlight the persistent nature of the conceptual model. It is not an artefact that gets created and is then dispensed with once the computer code has been written. It serves to document the basis of the computer model prior to development, during development and after development. Indeed, because the modelling process is iterative in nature (Robinson, 2004; Balci, 1985), the conceptual model is continually subject to change throughout the life-cycle of a simulation study.

Our definition is completed by a list of what a conceptual model describes. It is vital that the *objectives* of the model are known in forming the conceptual model. The model is designed for a specific purpose and without knowing this purpose it is impossible to create an appropriate simplification. Consider what would have happened if the purpose of the outpatients building model had not been properly understood. We would almost certainly have been driven to a more general purpose, and by nature much more complex, model. Poorly understood modelling objectives can lead to an overly complex model. Instead, because the purpose of the model was clear we were able to create a very simple model.

It is useful to know the model *inputs* and *outputs* prior to thinking about the content of the model. The *inputs* are the experimental factors that are altered in order to try and achieve the modelling objectives. In the example above, this was the number of consultation rooms in the outpatients building. The *outputs* are the statistics that inform us as to whether the modelling objectives are being achieved (e.g. the time to clear all patients from the outpatient system) and if not, why they are not being achieved.

Knowing the objectives, inputs and outputs of the model help inform the *content* of the model. In particular, the model must be able to receive the inputs (e.g. it must model the consultation rooms) and it must provide the outputs (e.g. it must model the flow of patients until all have exited the system).

The final two items that the conceptual model must describe are the *assumptions* and *simplifications* of the model. As illustrated in figure 3, these are quite distinct concepts (Robinson, 2008):

- *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.

So, assumptions are a facet of limited knowledge or presumptions, while simplifications are a facet of the desire to create simple models. These are generated through the process of knowledge acquisition and model abstraction respectively.

Is Conceptual Modelling Relevant?

I hope that the illustration and discussion above convinces the reader that conceptual modelling is relevant to all simulation modellers. After all, since all models are simplified versions of the real system they represent, then somewhere between understanding the real system and creating the computer code, someone must have done some conceptual modelling!

There are those, however, that raise objections to conceptual modelling, at least as a topic in its own right. Here I will discuss five common objections.

Objection 1: I can build the computer code without the need for a conceptual model.

This is an appeal that conceptual modelling is unnecessary. Indeed, it is true that the simulation model can be directly coded, especially when the model is relatively simple. This ability is enhanced by many modern simulation software tools giving the ability to code while thinking and to code through rapid prototyping.

What the objection misses is that even when the modeller apparently goes directly to computer coding, he/she is still making decisions about what to include and exclude from the model. As a result, conceptual modelling is still taking place, albeit not in a formal sense – hence, we discuss objection 2.

Objection 2: There is no need for a formal conceptual modelling process.

Agreed, and there is nothing in the discussion above which forces modellers to follow a formal conceptual modelling process. The conceptual model may remain conceptual, that is, a model in the head of the modeller. This can be entirely appropriate in some circumstances. However, it is often useful (and desirable) to communicate the conceptual model by documenting it and, hence, to be more formal; Nance (1994) describes this as the communicative model.

Objection 3: It is overly restrictive to require a full conceptual model before coding can begin.

Again agreed, because modelling is an iterative process. Whether conceptual modelling is being performed formally or informally, the conceptual is almost always refined through a series of iterative steps. Because the conceptual model is a persistent artefact of the simulation model, it continues to be refined before, during and after the model has been developed.

The nearest we might get to fully determining the conceptual model prior to any model coding, is where simulation is performed more as a software engineering project (Robinson, 2002). Here we see a team of modellers developing a simulation modelling product over an extended period of time. Such a project requires a more formal process including conceptual modelling. As a result of this, we see significant discussions around conceptual modelling in the military domain where a software engineering approach is most commonly employed due to the scale of the models involved (Pace, 2010).

Objection 4: How can you be sure you have the best conceptual model?

We cannot be sure of what constitutes the best conceptual model. This could only be discovered by building and fully validating all possible simulation models for a system and picking the best. It is neither feasible to build all models, since the set is likely to be extremely large, nor to fully validate any model (Robinson, 1999).

Only at an extreme can we say a model is completely wrong i.e. it models the wrong system. In general we can only describe models in relative terms as better or worse than one another. The aim of conceptual modelling is to identify the best possible model with the knowledge, time and resource available. This is not necessarily the best model, indeed, it is extremely unlikely that this will ever be the case.

Objection 5: Where is the science in conceptual modelling?

Simulation modelling has always been seen as involving art as well as science (Tocher, 1963; Shannon, 1975). Conceptual modelling is certainly more at the end of art. However, this is not an excuse to avoid the subject for more scientific endeavours. There is some agreement that conceptual modelling is the most important, and difficult, part of any simulation study (Law, 1991; Robinson, 2008). As a result, it must be the subject of attention for both simulation researchers and those in simulation practice.

It might be helpful to consider conceptual modelling to be a craft, rather than an art; but now we are in danger of arguing over semantics. 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.'

Some are making strides in bringing discipline to the craft of conceptual modelling, for instance: Kotiadis (2007) applies ideas from Soft Systems Methodology (Checkland, 1981); Arbez and Birta (2010) have developed the ABCmod Framework for conceptual modelling; van der Zee (2007) looks at principles for model decomposition. There is, however, a very long way to go.

In Summary

Conceptual modelling is important and it is something that all simulation modellers do, whether we follow a formal process or not in creating our simulation models. Conceptual models are persistent artefacts that maintain their life throughout the life-cycle of a simulation study. In theory, for any modelling project, a best conceptual model does exist, but we are extremely unlikely to find it. Our aim should be to identify the best possible conceptual model given the constraints of knowledge, time and resource. Conceptual modelling is not a science, but an art or craft. Despite some work to bring more discipline to this craft, there continues to be a need for a more concerted effort to develop the field of conceptual modelling in both research and practice.

We entitled this article with a question 'conceptual modelling: who needs it?' We close with the answer: 'every simulation modeller.'

Postscript: for those that would like to delve deeper into conceptual modelling for simulation and to understand the current state-of-the-art, then the book 'Conceptual Modeling for Discrete-Event Simulation' edited by Stewart Robinson, Roger J. Brooks, Kathy Kotiadis and Durk-Jouke van der Zee will be available through Routledge in the summer of 2010.

Acknowledgements

I acknowledge the help of Claire Worthington (Warwick Business School) in the modelling of the outpatients building. I am also grateful for the financial support of the Strategic Lean Implementation Methodology Project (SLIM) (www2.warwick.ac.uk/fac/soc/wbs/projects/slim) which is funded by the Warwick Innovative Manufacturing Research Centre.

References

Arbez, G. and Birta, L.G. (2010). The ABCmod Conceptual Modelling Framework *Conceptual Modeling for Discrete Event Simulation* (Robinson, S., Brooks, R.J., Kotiadis, K. and van der Zee, D.-J., eds.), Routledge, Florence, Kentucky Chapter 6. *Forthcoming.*

Balci, O. (1985). *Guidelines for Successful Simulation Studies.* Technical Report TR-85-2, Department of Computer Science, Virginia Tech, Blacksburg, VA.

Checkland, P.B. 1981. *Systems Thinking, Systems Practice.* Wiley, Chichester, UK.

Ferguson, P., Humphrey, W.S., Khajenoori, S., Macke, S. and Matuya, A. (1997). Results of Applying the Personal Software Process. Computer, 5: 24-31.

Kotiadis, K. (2007). Using Soft Systems Methodology to Determine the Simulation Study Objectives. *Journal of Simulation*, 1 (3): 215-222.

Law, A.M. (1991). Simulation Model's Level of Detail Determines Effectiveness. *Industrial Engineering*, 23 (10): 16-18.

Nance, R.E. (1994). The Conical Methodology and the Evolution of Simulation Model Development. *Annals of Operations Research*, 53: 1-45.

Pace, D.K. (2010). Conceptual Modeling Evolution within U.S. Defense Communities: The View from the Simulation Interoperability Workshop. *Conceptual Modeling for Discrete Event Simulation* (Robinson, S., Brooks, R.J., Kotiadis, K. and van der Zee, D.-J., eds.), Routledge, Florence, Kentucky, Chapter 16. *Forthcoming.*

Robinson, S. (1999). Simulation Verification, Validation and Confidence: A Tutorial. *Transactions of the Society for Computer Simulation International*, 16 (2): 63-69.

Robinson, S. (2002). Modes of Simulation Practice: Approaches to Business and Military Simulation. *Simulation Practice and Theory*, 10: 513-523.

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

Robinson, S. (2008). Conceptual Modelling for Simulation Part I: Definition and Requirements. *Journal of the Operational Research Society*, 59 (3): 278-290.

Robinson, S. (2010). Conceptual Modeling for Simulation. *Encyclopedia of Operations Research and Management Science* (Cochran, J.J., ed.). Wiley, New York. *Forthcoming.*

Shannon, R.E. (1975). *Systems Simulation: The Art and Science*. Prentice-Hall, Englewood Cliffs, NJ.

Tocher K.D. (1963). *The Art of Simulation.* The English Universities Press, London.

van der Zee, D.-J. (2007). Developing Participative Simulation Models – Framing Decomposition Principles for Joint Understanding. *Journal of Simulation*, 1: 187-202.

Zeigler, B.P. (1976). *Theory of Modeling and Simulation*. Wiley, New York.

Author Biography

Stewart Robinson is a Professor of Operational Research at Warwick Business School and Associate Dean for Specialist Masters Programmes. He holds a BSc and PhD in Management Science from Lancaster University. Previously employed in simulation consultancy, he supported the use of simulation in companies throughout Europe and the rest of the world. He is author/co-author of four books on simulation, including a shortly to be published edited volume on conceptual modelling for discrete-event simulation. His research focuses on the practice of simulation model development and use. Key areas of interest are conceptual modelling, model validation, output analysis, modelling human factors in simulation models and comparison of simulation methods. His email address is stewart.robinson@warwick.ac.uk.