Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds

CONCEPTUAL MODELING FOR SIMULATION

Stewart Robinson

School of Business and Economics
Loughborough University
Loughborough, LE11 3TU, UNITED KINGDOM

ABSTRACT

Conceptual modeling is the abstraction of a simulation model from the real world system that is being modeled; in other words, choosing what to model, and what not to model. This is generally agreed to be the most difficult, least understood and most important task to be carried out in a simulation study. In this tutorial the problem of conceptual modeling is first illustrated through an example of modeling a hospital clinic. We then define a set of terminology that helps us frame the conceptual modeling task, we discuss the role of conceptual modeling in the simulation project life-cycle, we identify the requirements for a good conceptual model and we discuss levels of abstraction. A framework that guides the activity of conceptual modeling is described. This framework may also be helpful for teaching effective conceptual modeling.

1 INTRODUCTION

One of the most difficult issues in simulation modeling is determining the content of the simulation model. The job of the modeler is to understand the real system that is the subject of the simulation study and to turn this into an appropriate simulation model. The chosen model could range from a very simple single server and queue, through to a model that tries to encapsulate every aspect of the system. In effect, there are an infinite number of models that could be selected within this range, each with a slightly different content. The question is: which model should we choose? We explore the answer to this question in this paper.

On the surface we might suggest the answer is to build the model that contains as much detail as possible. After all, this model will be the closest to the real system and so surely the most accurate. This might be true if we had complete knowledge of the real system and a very large amount of time available to develop and run the model. But what if we only have limited knowledge of the real system and limited time? Indeed, we rarely have the luxury of vast quantities of either knowledge or time, not least because the real system rarely exists at the time of modeling (it is a proposed world) and a decision needs to be made according to a tight time schedule.

So, if we need to develop a simpler model, we need to determine the level of abstraction at which to work. This process of abstracting a model from the real world is known as conceptual modeling. We shall define conceptual modeling and the process of doing it in more detail in a while, but first it is useful to illustrate the issues involved in conceptual modeling with a practical example.

2 CONCEPTUAL MODELING EXAMPLE: SIMULATING AN OUTPATIENTS CLINIC

Our simulation and modeling team was asked to develop a simulation model for a hospital in Birmingham, UK. The hospital was 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 calcula tions 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 realized 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 modeling 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 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.

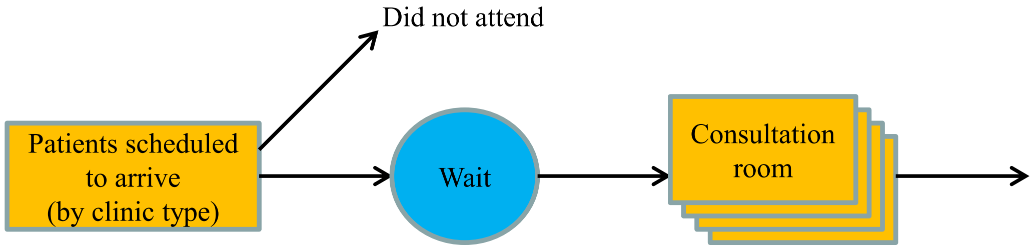

Figure 1: Simple Outpatients Building Model.

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 recognized, 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.

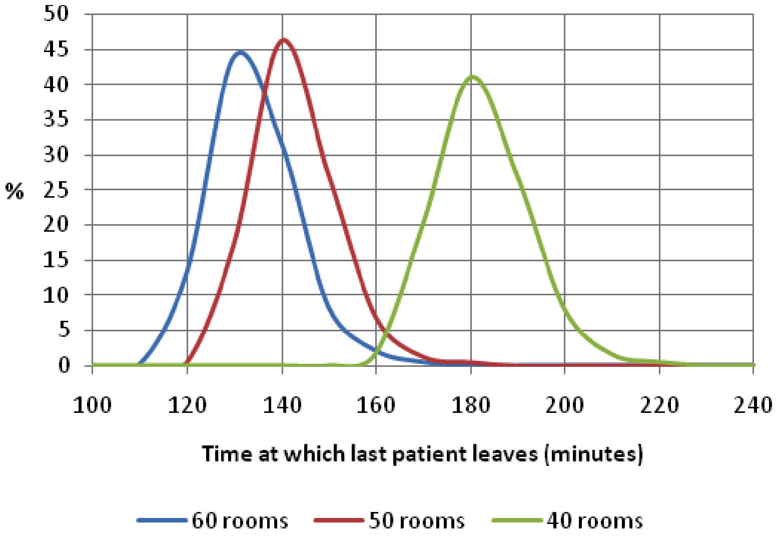

Figure 2: Example of Results from the Outpatients Building Model: Frequency Distributions for Time until Last Patient Leaves.

This example illustrates the very essence of conceptual modeling; 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 modeling only Monday morning's clinic and not modeling 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. It is certain that readers will form quite different judgments on the credibility of the model and so whether it was a good model or not.

3 WHAT IS CONCEPTUAL MODELING?

Conceptual modeling 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 a simplified representation of the real system in the simulation model. The secret to good conceptual modeling is to get the level of simplification correct, that is, to abstract at the right level.

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

3.1 Definition of a Conceptual Model

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 2008a)

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.

Second, 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, the conceptual model persists long beyond the end of the simulation study, since we cannot dispose of the model concept. Of course, because the modeling process is iterative in nature (Balci 1994; Willemain 1995; Robinson 2004), the conceptual model is continually subject to change throughout the life-cycle of a simulation study.

Finally, the 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 modeling 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 modeling 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 modeling 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 (e.g. the utilization of the consulting rooms).

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 model content can be thought of in terms of the model scope (what to model) and the level of detail (how to model it).

The final two items in the list of what a conceptual model describes are the *assumptions* and *simplifications* of the model. These are quite distinct concepts (Robinson 2008a):

- *Assumptions* are made either when there are uncertainties or beliefs about the real world being modeled.
- *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.

3.2 Artefacts of Conceptual Modeling

To understand conceptual modeling further it is useful to set it within the wider context of the modeling process for simulation. Figure 3 shows the key artefacts of conceptual modeling. The 'cloud' represents the real world (current or future) within which the problem situation resides; this is the problem that is the basis for the simulation study. The four rectangles represent specific artefacts of the (conceptual) modeling process. These are as follows:

- *System description*: a description of the problem situation and those elements of the real world that relate to the problem.
- *Conceptual model*: as defined in section 3.1
- *Model design*: the design of the constructs for the computer model (data, components, model execution, etc.) (Fishwick 1995).
- *Computer model*: a software specific representation of the conceptual model.

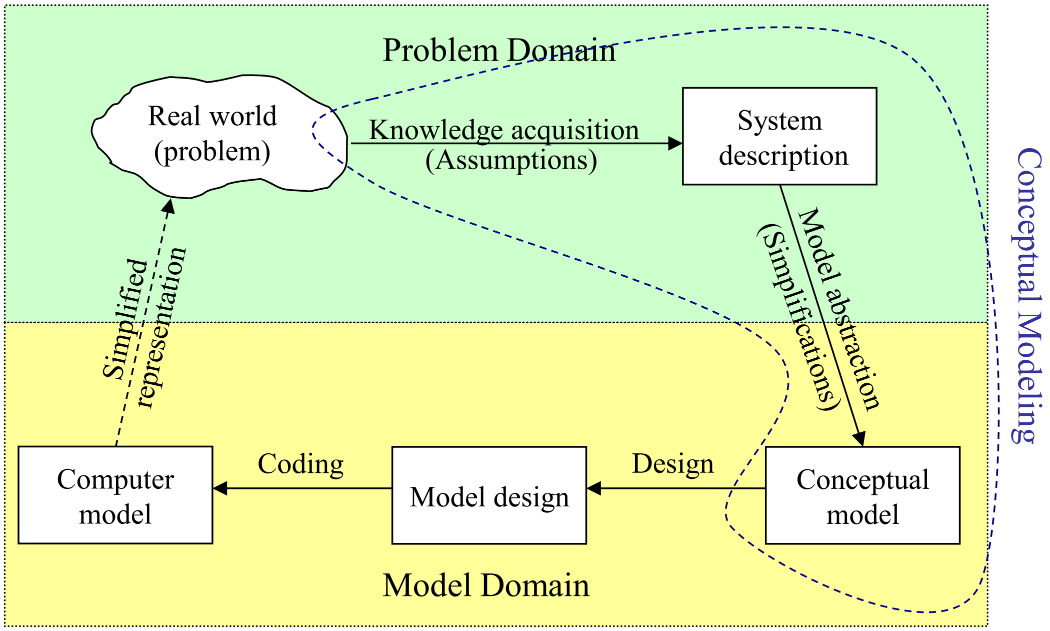

Figure 3: Artefacts of Conceptual Modeling (Robinson 2011).

These artefacts are quite separate. This is not to say that they are always explicitly expressed, with the exception of the computer model. For instance, the system description, conceptual model and model design may not be (fully) documented and can remain within the minds of the modeler and the problem owners. It is, of course, good modeling practice to document each of these artefacts and to use this as a means of communicating their content with the simulation project clients.

The model design and computer model are not strictly part of conceptual modeling, but they do embody the conceptual model within the design and code of the model. These artefacts are included in Figure 3 for completeness. Our main interest here is in the system description and conceptual model which make up the process of conceptual modeling; as represented by the shape with a dashed outline in Figure 3. Unlike the model design and computer model, these two artefacts are independent of the software that will ultimately be used for developing the simulation model.

It is important to recognize the distinction between the system description and the conceptual model. The system description relates to the problem domain, that is, it describes the problem and those elements of the real world that relate to the problem. The conceptual model belongs to the model domain in that it describes those parts of the system description that are included in the simulation model and at what level of detail. The author's experience is that these two artefacts are often confused and seen as indistinct. Indeed, a major failure in any simulation project is to try and model the system description (i.e. everything that is known about the real system) and to not attempt any form of model abstraction; this leads to overly complex models.

The arrows in Figure 3 represent the flow of information, for instance, information about the real world feeds into the system description. The processes that drive the flow of information are described as knowledge acquisition, model abstraction, design and coding. The arrows are not specifically representative of the ordering of the steps within the modeling process, which we know are highly iterative (Balci 1994; Willemain 1995; Robinson 2004). In other words, a modeler may return to any of the four processes at any point in a simulation study, although there is some sense of ordering in that information from one artefact is required to feed the next artefact.

The dashed arrow shows that there is a correspondence between the computer model and the real world. The degree of correspondence depends on the degree to which the model contains assumptions that are correct, the simplifications maintain the accuracy of the model, and the model design and computer code are free of errors. Because the model is developed for a specific purpose, the correspondence with the real world only relates to that specific purpose. In other words, the model is not a general model of the real world, but a simplified representation developed for a specific purpose. The issue of whether the level of correspondence between the model and the real world is sufficient is an issue of validation (Landry, Malouin, and Oral 1983; Balci 1994; Robinson 1999; Sargent 2008). Both conceptual modeling and validation are concerned with developing a simulation of sufficient accuracy for the purpose of the problem being addressed. As a result, there is a strong relationship between the two topics, conceptual modeling being concerned with developing an appropriate model and validation being concerned with whether the developed model is appropriate.

3.3 The Relationship with Conceptual Modeling in Information Systems and Software Engineering

Arthur and Nance (2007) discuss the role of software requirements engineering (SRE) in simulation conceptual modeling. They find very little evidence that formal SRE activities are being used in simulation. More recently Guizzardi and Wagner (2012) discuss the links between conceptual modeling in information systems and software engineering (IS/SE), and simulation. They propose the use of Onto-UML for simulation conceptual modeling. This goes someway to addressing Arthur and Nance's concerns that simulation conceptual modeling is quite separate to the work in SRE.

Guizzardi and Wagner identify an important discrepancy in the definition of a conceptual model in IS/SE and simulation. They state that in IS/SE conceptual (or domain) models are 'solution-independent descriptions of a problem domain.' In other words, the conceptual model belongs in the problem domain and is something akin to the system description in Figure 3. The definition of a conceptual model used in this paper places the conceptual model firmly in the model domain, and places its definition closer to a 'platform-independent design model' as described by Guizzardi and Wagner. Reconciling, or at least recognizing, this discrepancy is important if concepts and approaches from IS/SE are to be used in the simulation context.

We argue here that in simulation the conceptual model should describe the simulation model and not the real world. The conceptual model should describe how we conceive the model, in other words, how we have abstracted the model away from our understanding of the real world (system description). This distinction is important in simulation because of the emphasis on model abstraction. Consider the model described in section 2. Our conception of this model (conceptual model) is very distinct (and distant) from our description of the real world. Tolk et al. (2013), writing from a systems engineering perspective, make a similar point by distinguishing between the 'reference model', which describes the problem domain, and the 'conceptual model', which is the foundation for computer implementation.

So, if we are to adopt conceptual modeling approaches from IS/SE we must be mindful that these focus on describing the problem domain (knowledge elicitation) and not on model abstraction. To use these methods without cognizance of the need to also abstract away from our understanding of the real system could lead us to build overly complex simulation models; an issue to which our focus now turns.

4 REQUIREMENTS OF A CONCEPTUAL MODEL: CHOOSING WHAT TO MODEL

Before discussing how to perform conceptual modeling, let us consider what makes for a good conceptual model. The key requirements are that the model should be valid, credible, feasible and useful (Robinson 2008a). By these we mean the model should:

- 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.

Overarching all of this is the requirement to build the simplest model possible to meet the objectives of the simulation study. According to Innis and Rexstad (1983), Ward, (1989), Salt (1993), Chwif, Barreto, and Paul (2000), Lucas and McGunnigle (2003), and Thomas and Charpentier (2005), simpler models are preferred because:

- 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

As such, the need to abstract a conceptual model from the system description becomes even more pertinent. This does not, of course, mean that we should never develop more complex models, but that we should only develop them if they are required to meet the modeling objectives.

Figure 4 illustrates the relationship between model accuracy and model complexity (scope and level of detail). It shows that with increasing levels of complexity we obtaining diminishing returns in terms of accuracy, never reaching 100% accuracy. Eventually we may even find that the accuracy of the model reduces. This is because we do not have the knowledge or data to support the complexity that is being included in the model and we start to make assumptions that are incorrect.

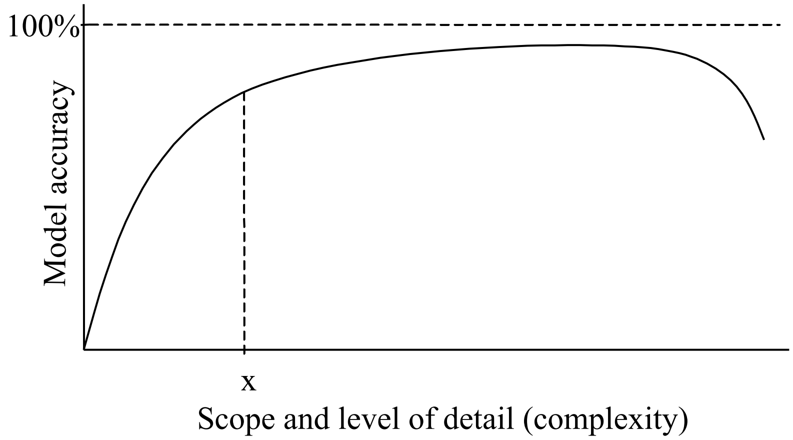

Figure 4: Simulation Model Complexity and Accuracy (Robinson 2008a).

So which conceptual model should we choose? We might argue that the model at point x in Figure 4 is the best. At this point we have gained a high level of accuracy for a low level of complexity. Moving beyond x will only marginally increase accuracy and adding further complexity generally requires ever increasing effort. Of course, if we have a specific need for an accuracy level greater than that provided by x, we will need to increase the complexity of the model.

The difficulty is in finding point x. Conceptual modeling frameworks, such as the ones listed below, aim to help us in that quest, but conceptual modeling is more of an art than a science (we might prefer to use the word 'craft'). As a result, we can only really hope to get close to x. In other words, there may be a 'best' model, but we are extremely unlikely to find it among an infinite set of models. What we should hope to do is identify the best model we can. As such, our quest is for better models, not necessarily the best.

5 FRAMEWORKS FOR CONCEPTUAL MODELING

A framework for conceptual modeling provides a set of steps and tools that guide a modeler through the development of a conceptual model. It is also useful for teaching conceptual modeling, especially to novice modelers. The simulation literature, however, provides very few such frameworks. Some examples, that the reader may wish to explore further are:

- Conceptual modeling framework for manufacturing (van der Zee 2007)
- The ABCmod conceptual modeling framework (Arbez and Birta 2011)
- Karagöz and Demirörs (2011) present a number of conceptual modeling frameworks: Conceptual Model Development Tool (KAMA), Federation Development and Execution Process (FEDEP), Conceptual Models of the Mission Space (CMMS), Defense Conceptual Modeling Framework (DCMF), and Base Object Model (BOM).
- The PartiSim framework (Tako et al. 2010)

For a more detailed discussion on conceptual modeling frameworks see Robinson et al. (2011). Here a very brief outline of the Robinson framework for conceptual modeling is given. For a more detailed account, and an illustration of the framework in use, see Robinson (2008b).

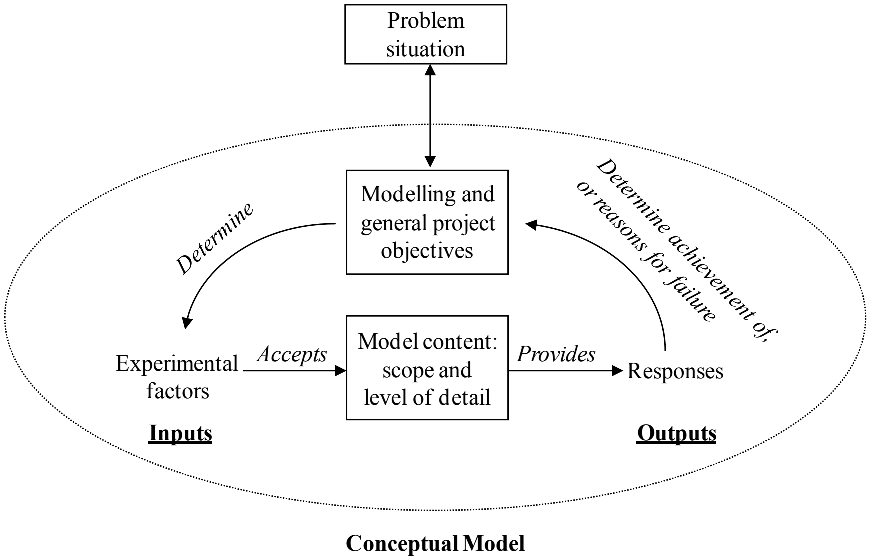
Problem situation
Modelling and general project objectives
Determine
Determine achievement of, or reasons for failure
Experimental factors
Accepts
Model content: scope and level of detail
Provides
Responses
Inputs
Outputs
Conceptual Model

Figure 5: A Framework for Conceptual Modeling (Robinson 2008b).

Figure 5 outlines Robinson's conceptual modeling framework. In this framework, conceptual modeling involves five activities that are performed roughly in this order:

- Understanding the problem situation
- Determining the modeling 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

Starting with an understanding of the problem situation, a set of modeling and general project objectives are determined. These objectives then drive the derivation of the conceptual model, first by defining the outputs (responses) of the model, then the inputs (experimental factors), and finally the model content in terms of its scope and level of detail. Assumptions and simplifications are identified throughout this process.

The ordering of the activities described above is not strict. Indeed, we would expect much iteration between these activities and with the other activities involved in a simulation study: data collection and analysis, model coding, verification and validation, experimentation and implementation.

The framework is supported by a conceptual model template which provides a set of tables that describe each element of the conceptual model. These tables describe:

- Modeling and general project objectives (organisational aim, modeling objectives, general project objectives)
- Model outputs/responses (outputs to determine achievement of objectives, outputs to determine reasons for failure to meet objectives)
- Experimental factors
- Model scope
- Model level of detail
- Modeling assumptions
- Model simplifications

Beyond completing these tables, it is also useful to provide a diagram of the model. For instance, process flow diagrams, similar to that presented in Figure 1, are useful for communicating the conceptual model.

The modeler works through these tables with the support of the stakeholders and domain experts, iteratively improving them to the point that the modeler and stakeholders are satisfied that the conceptual model meets the requirements for validity, credibility, feasibility and utility. This provides a structured framework for making the conceptual modeling decisions explicit (documentation) and for debating ways of improving the conceptual model. An illustration of the conceptual model template that accompanies this framework, using the example of a simple fast food restaurant problem, is provided at http://www-staff.lboro.ac.uk/~bsslr3/.

6 LEVELS OF ABSTRACTION

The conceptual modeling example in section 2 is described as a 'far abstraction.' By this we mean that the conceptual model involves many simplifications and so it is removed a long way from the system description. The implication of this is that the computer model is a highly simplified representation of the real world.

At the extreme, a far abstraction can lead to a (conceptual) model that bears little resemblance to the real world. As an example of this we briefly discuss Schelling's model of segregation (Schelling 1971). This is an early example of agent-based simulation in which the dynamics of a population, that is split into two groups who aim for a desired level of segregation, is investigated. Figure 6 shows an example of this model. The world is represented as a grid with green and red tokens. A token desires a certain number of its neighbors to be of a similar colour. If they are not, then the token moves to another space on the grid. This process continues until all tokens are satisfied with their neighborhood. The model demonstrates that a much higher level of segregation is achieved than desired by each individual.

Schelling used this simple model to understand and explain segregation in urban areas. It is obvious that the model is not a full explanation of individual behaviors that lead to segregation, in fact, it is not even based on empirical data. As such, it is clearly abstracted a long way from what we know about the real world.

From a modeling perspective we have to ask whether a complex model that tried to represent individual details, based on empirical data, would have had greater power for understanding segregation? Given the complexities involved this seems unlikely. We can even question whether an empirical model would have had any greater predictive power. What we do know is that the model would require orders of magnitude of additional effort to create and run.

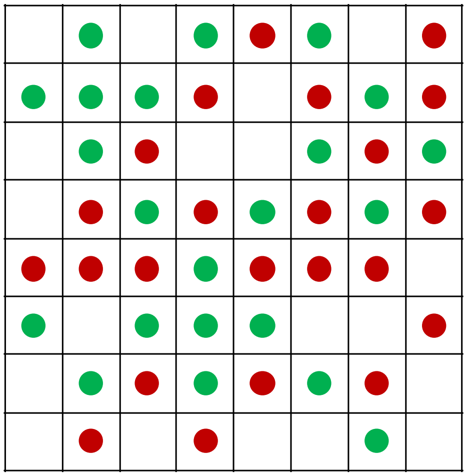

Figure 6: Schelling's Segregation Model.

This example, and the one in section 2, serve to illustrate the extent to which a model can be abstracted away from the real system to create a 'far abstraction.' However, we would not want to leave the impression that conceptual models have to be so far abstracted. Indeed it is not always desirable to abstract to this degree and for some simulation studies it is appropriate to model much of the scope and detail in the problem domain. We refer to this as 'near abstraction.' For an example, see the Ford engine plant model described in Robinson (2008a, 2008b). These papers describe a simulation that was designed to determine the throughput of a new engine assembly plant. The model contained much detail about the real system and took a considerable time to develop.

The level of abstraction should be determined by the requirement for the model to be valid, credible, feasible and useful. One danger with far abstraction is that whilst the model may be valid, it may lack credibility. Hence, we may need to reduce the level of abstraction, making the model nearer to the system description, to increase the credibility of the model.

7 CONCLUSION

Conceptual modeling is the abstraction of a simulation model from a real world system. It is probably the most important aspect of any simulation study. Get the conceptual model right and the rest of the simulation work will be more straightforward, providing the right information in the right time-scale.

This paper provides an illustration of how appropriate conceptual modeling, through far abstraction, made a simulation study feasible within the constraints of data and time available. The discussion that follows defines conceptual modeling, its artefacts and its requirements. From this base, some frameworks for conceptual modeling are listed and one framework is outlined in more detail. The framework aims to guide a modeler through the process of creating and documenting a conceptual model. We also discuss levels of abstraction, from far to near.

Conceptual modeling is not a science, but a craft or even an art. As with any craft, it can be learned and it can be improved upon with experience. Frameworks provide a good way of learning about conceptual modeling and for helping to do it better. At present, however, there are very few examples of conceptual modeling frameworks and this is an area where more research needs to be undertaken.

ACKNOWLEDGEMENTS

I acknowledge the help of Claire Worthington (University of Central Lancashire) in the modeling of the outpatients building. I am also grateful for the financial support of the Strategic Lean Implementation Methodology Project (which was funded by the Warwick Innovative Manufacturing Research Centre.

Sections of this paper are based on Robinson, S. 2010. "Conceptual Modelling: Who Needs It?" *SCS Modeling & Simulation Magazine* 1 (2): April. www.scs.org/magazines/2010-04/index_file/Files/Robinson.pdf; Robinson, S. 2011. "Conceptual Modeling for Simulation." In *Encyclopedia of Operations Research and Management Science*, Edited by J.J. Cochran, forthcoming. New York: Wiley; Robinson, S. 2011. "Designing Simulations that are Better than the Rest: Conceptual Modelling for Simulation". Keynote paper, Young OR Conference, Nottingham, 2011. Birmingham, UK: The Operational Research Society.

REFERENCES

Arbez, G. and Birta, L.G. 2011. "The ABCmod Conceptual Modeling Framework." In *Conceptual Modeling for Discrete-Event Simulation*, Edited by S. Robinson, S., R.J. Brooks, K. Kotiadis, and D-J. van der Zee, 133-178. Boca Raton, FL: Chapman and Hall/CRC.

Arthur, J.D. and R.E. Nance. 2007. "Investigating the Use of Software Requirements Engineering Techniques in Simulation Modelling." *Journal of Simulation* 1 (3): 159-174.

Balci, O. 1994. "Validation, Verification, and Testing Techniques Throughout the Life Cycle of a Simulation Study*." Annals of Operations Research* 53: 121-173.

Chwif, L., M.R.P. Barretto, and R.J. Paul. 2000. "On Simulation Model Complexity." In *Proceedings of the 2000 Winter Simulation Conference*, Edited by J.A. Joines, R.R. Barton, K. Kang, and P.A. Fishwick, 449-455. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Fishwick, P.A. 1995. *Simulation Model Design and Execution: Building Digital Worlds*. Upper Saddle River, New Jersey: Prentice-Hall, Inc.

Guizzardi, G. and G. Wagner. 2012. Tutorial: Conceptual Simulation Modeling with Onto-UML. In *Proceedings of the 2012 Winter Simulation Conference*, Edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Innis, G., and E. Rexstad. 1983. "Simulation Model Simplification Techniques." *Simulation* 41 (1): 7-15.

Karagöz, N.A. and Demirörs, O. 2011. "Conceptual Modeling Notations and Techniques." In *Conceptual Modeling for Discrete-Event Simulation*, Edited by S. Robinson, S., R.J. Brooks, K. Kotiadis, and D-J. van der Zee, 179-209. Boca Raton, FL: Chapman and Hall/CRC.

Landry, M., J.L. Malouin, and M. Oral. 1983. "Model Validation in Operations Research." *European Journal of Operational Research* 14 (3): 207-220.

Lucas, T.W., and J.E. McGunnigle. 2003. "When is Model Complexity too Much? Illustrating the Benefits of Simple Models with Hughes' Salvo Equations." *Naval Research Logistics* 50: 197-217.

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

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

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

Robinson, S. 2008b. "Conceptual Modelling for Simulation Part II: A Framework for Conceptual Modeling." *Journal of the Operational Research Society* 59 (3): 291-304.

Robinson, S. 2011. "Conceptual Modeling for Simulation." In *Encyclopedia of Operations Research and Management Science*, Edited by J.J. Cochran, forthcoming. New York: Wiley.

Robinson, S., R.J. Brooks, K. Kotiadis, and D.J. van der Zee. 2011. *Conceptual Modelling for Discrete-Event Simulation*. FL, USA: Taylor and Francis.

Salt, J. 1993. "Simulation Should be Easy and Fun." In *Proceedings of the 1993 Winter Simulation Conference*, Edited by G.W. Evans, M. Mollaghasemi, E.C. Russell, and W.E. Biles, 1-5. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Sargent, R.G. 2008. "Verification and Validation of Simulation Models." In *Proceedings of the 2008 Winter Simulation Conference*, Edited by S.J. Mason, R.R. Hill, L. Mönch, O. Rose, T. Jefferson, and J.W. Fowler, 157-169. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Schelling, T.C. 1971. "Dynamic Models of Segregation." *Journal of Mathematical Sociology* 1: 143-186.

Schruben, L. W. 1979. "Designing Correlation Induction Strategies for Simulation Experiments." In *Current Issues in Computer Simulation*, Edited by N. R. Adam and A. Dogramaci, 235–256. New York: Academic Press.

Tako, A.A., Kotiadis, K. and Vasilakis, C. 2010. "A Participative Modelling Framework for developing Conceptual Models in Healthcare Simulation Studies." In *Proceedings of the 2010 Winter Simulation Conference*, Edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 500-512. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Thomas, A., and P. Charpentier. 2005. "Reducing Simulation Models for Scheduling Manufacturing Facilities." *European Journal of Operational Research* 161 (1): 111-125.

Tolk, A., S.Y. Diallo, J.J. Padilla and H. Herencia-Zapana. 2013. "Reference Modelling in Support of M&S – Foundations and Applications." *Journal of Simulation* 7 (2): 69-82.

van der Zee, D.J. 2007. "Developing Participative Simulation Models: Framing Decomposition Principles for Joint Understanding." *Journal of Simulation* 1 (3): 187-202.

Ward, S.C. 1989. "Arguments for Constructively Simple Models." *Journal of the Operational Research Society* 40 (2): 141-153.

Willemain, T.R. 1995. "Model Formulation: What Experts Think About and When." *Operations Research* 43 (6): 916-932.

AUTHOR BIOGRAPHY

STEWART ROBINSON is Professor of Management Science and Associate Dean Research at Loughborough University, School of Business and Economics. 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 five books on simulation. His research focuses on the practice of simulation model development and use. Key areas of interest are conceptual modelling, model validation, output analysis and alternative simulation methods (discrete-event, system dynamics and agent based). Professor Robinson is co-founder of the Journal of Simulation and President of the Operational Research Society. Home page: www.stewartrobinson.co.uk. His email is s.l.robinson@lboro.ac.uk.