

## **A TUTORIAL ON SIMULATION CONCEPTUAL MODELING**

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 part of the real world it is representing; in other words, choosing what to model, and what not to model. This is generally agreed to be the most difficult, least understood, but probably the most important activity to be carried out in a simulation study. In this tutorial we explore the definition, requirements and approach to conceptual modeling. First we ask ‘where is the model?’ We go on to define the term ‘conceptual model’, to identify the artefacts of conceptual modeling, and to discuss the purpose and benefits of a conceptual model. In so doing we identify the role of conceptual modeling in the simulation project life-cycle. The discussion then focuses on the requirements of a conceptual model, the approaches for documenting a conceptual model, and frameworks for guiding the conceptual modeling activity. One specific framework is described and illustrated in more detail. The tutorial concludes with a discussion on the level of abstraction.

### **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 create an appropriate simulation model of that system. 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, or even very, different content. The question is: which model should we choose? We explore the answer to this question in this tutorial.

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. Even in an environment of plentiful knowledge and time, a simpler model is often sufficient to address the problem at hand and so there is no need for a more complex model.

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. In this tutorial we shall define conceptual modeling, its requirements and discuss how to approach conceptual modeling.

### **2 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.

## 2.1 Where is the Model?

Before providing a more formal definition of a conceptual model, first let us consider the question: where is the model? Figure 1 illustrates three possible locations for the model: in the mind of the modeler, in the documentation for the model, and on the computer. The modeler generates the concept for the model, which is either documented and transferred to a computer, or it is directly transferred to the computer. We argue here that the model exists within the mind of the modeler and that the documentation (should it exist) and the computer model are simply explicit representations of that model. Hence, modeling is a cognitive process; documenting and coding are ways of making that process explicit. In this respect, conceptual modeling is the cognitive process of conceiving the model.

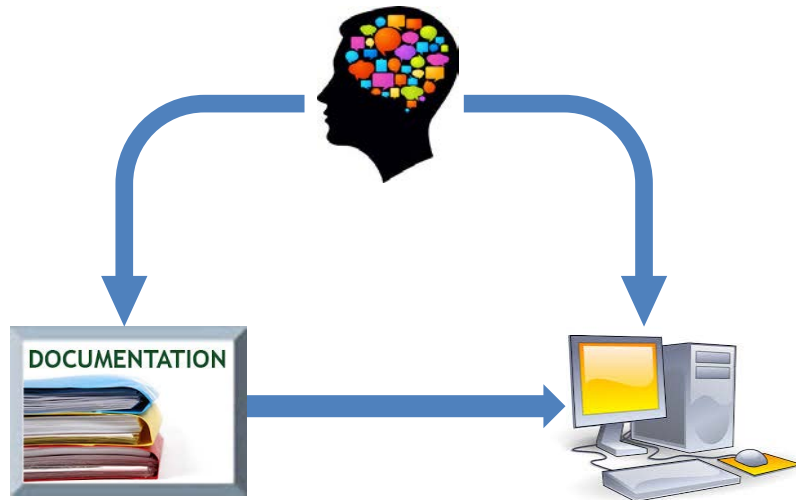


Figure 1: Where is the model? (Robinson et al. 2015)

## 2.2 Definition of a Conceptual Model

Conceptual modeling is the activity of deciding what to model and what not to model – ‘model abstraction’. A 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*’ (Robinson 2008a). This definition establishes four facets of a conceptual model, as follows.

*Facet 1: the conceptual model is non-software specific.* Considerations about what to model and what not to model should not be directed by the software that is used for developing the model code. Indeed, the conceptual model should direct the choice of software.

*Facet 2: the conceptual model describes the computer simulation model.* The conceptual model should describe how we conceive the model, and it should not describe the real system. In other words, the conceptual model describes how we have abstracted the model away from our understanding of the real world. When the conceptual model entails a significant level of abstraction (i.e., many simplifications) it can be very distinct (and ‘far’) from our description of the real world.

*Facet 3: the conceptual model is a persistent artifact.* The conceptual model describes the computer simulation model *that will be, is or has been developed*. This statement identifies the conceptual model as a ‘persistent artifact’. From the inception of the simulation study the modeler starts to form a conceptual model; identifying how the real system might be modeled. As the simulation study progresses and the modeler learns more about the real system and the requirements of the clients, the conceptual model is very likely to change. Eventually the conceptual model is embedded within the code of the computer

model which is subsequently validated and used. The conceptual model still exists, at least in the mind of the modeler, throughout this life-cycle and even beyond the use of the model. The modeler may even revise the conceptual model, that is, his/her understanding of the best way to model the problem, after the model is in use or has become obsolete.

This third facet highlights three important points. First, conceptual modeling is not a pre-coding phase, but a continuous activity that starts at the inception of a simulation study, and continues throughout and even beyond the study's formal completion. Second, the conceptual model exists within the mind of the modeler (it is the modeler's implicit concept of the model) and it may or may not be formally expressed (made explicit). Third, model documentation and the model code are means of formally expressing (making explicit) the conceptual model. Even if the conceptual model is never documented, or indeed, the model code written, the conceptual model still exists.

*Facet 4: the components of the conceptual model.* The conceptual model consists of a set of components: *the objectives, inputs, outputs, content, assumptions and simplifications of the model.* The modeling *objectives* describe the purpose of the model and simulation project. 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 in order to meet the modeling objectives. The *outputs* (or responses) report the results from a run of the simulation model. These have two purposes: first, to determine whether the modeling objectives have been achieved; second, to point to reasons why the objectives are not being achieved, if they are not. The *model content* consists of the components that are represented in the model and their interconnections. The model content can be thought of in terms of two dimensions:

- *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.

*Assumptions* and *simplifications* are conceptually different. *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 the transparency of the model. So, assumptions are a facet of limited knowledge or presumptions, while simplifications are a facet of the desire to create simple models.

### 2.3 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 2 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 activity. 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 2.2
- *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.

These artefacts are quite separate. However, with the exception of the computer model, they may not be explicitly expressed. 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.

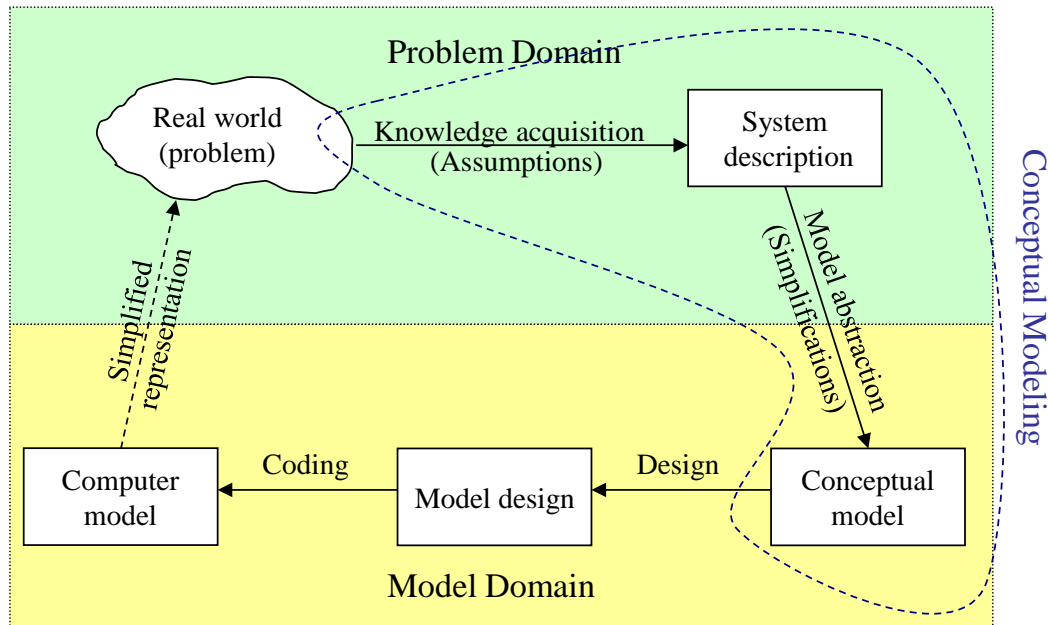


Figure 2: The artefacts of conceptual modeling (Robinson 2011).

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 2 for completeness. Our main interest here is in the system description and conceptual model which make up the activity of conceptual modeling, as represented by the shape with a dashed outline in Figure 2. 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; although the capabilities of the chosen software may in some ways constrain the (conceptual) 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 2 represent the flow of information, for instance, information about the real world feeds into the system description. The activities 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 2014). In other words, a modeler may return to any of the four activities 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 specific and different roles of assumptions and simplifications are highlighted in Figure 2. Assumptions relate to knowledge acquisition, that is, they fill in the gaps in the knowledge that can be acquired about the real world. Meanwhile, simplifications relate to model abstraction, since they are deliberate choices to model the world more simply.

The dashed arrow in Figure 2 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 is-

sue 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 2013; Law 2015). 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.

The artefacts described in this section are similar to Zeigler's concepts of the real system, the experimental frame, the base model, the lumped model, and the computer. The interested reader is referred to Zeigler and Kim (2000).

## 2.4 Purpose and Benefits of Conceptual Modeling

The question of the purpose of conceptual modeling is, in fact, superfluous in that a simulation model cannot exist without there being a conceptual model as defined above. No model can exist without there being a concept of that model. The valid question is to ask: what is the purpose of making the conceptual model explicit by documenting (and communicating) the model? The answer: it provides a means of communication between all parties in a simulation study, for example, the modeler, code developers, domain experts, end users and clients. In so doing it helps to build a consensus, or least an accommodation, about the nature of the model and its use.

The benefits of a documented conceptual model are as follows:

- It minimizes the likelihood of incomplete, unclear, inconsistent and wrong requirements
- It helps build the credibility of the model
- It guides the development of the computer model
- It forms the basis for model verification and guides model validation
- It guides experimentation by expressing the modeling objectives, and model inputs and outputs
- It provides the basis of the model documentation
- It can act as an aid to independent verification and validation when it is required
- It helps determine the appropriateness of the model or its parts for model reuse and distributed simulation

## 3 REQUIREMENTS OF A CONCEPTUAL 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 have utility (Robinson 2008a). By these we mean the model should:

- Produce sufficiently accurate results for the purpose at hand (*validity*);
- Be believed by the clients (*credibility*);
- Be *feasible* to build within the constraints of the available data and time;
- Have *utility*, 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 structure of the model is easier to understand;
- The results are easier to interpret.

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. For further discussion on the topic of model complexity, with respect to how a model is used, see Pidd (2010).

Figure 3 illustrates the relationship between model accuracy and model complexity (scope and level of detail). It shows that with increasing levels of complexity we obtain 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.

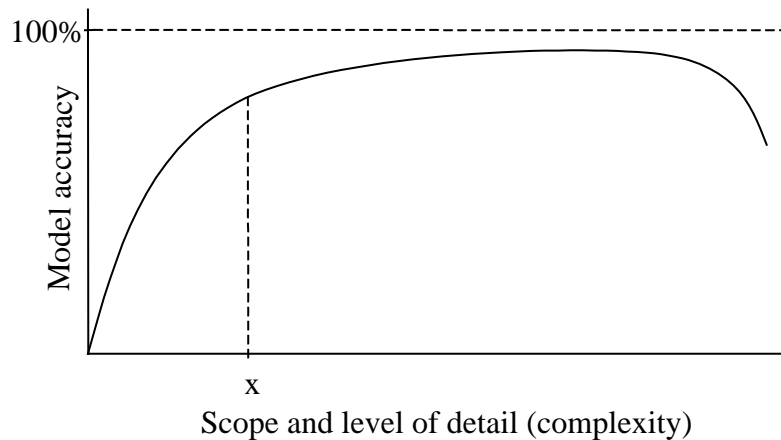


Figure 3: How simulation model accuracy changes with the complexity of the model (Robinson 2008a).

So which conceptual model should we choose? We might argue that the model at point  $x$  in Figure 3 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. Indeed, Sargent (2013) suggests that a model's 'acceptable range of accuracy' should be determined early in the modeling process so it can guide both model development and model validation. See Shannon (1998) for further discussion on this point in relation to Pareto's law.

The difficulty is in finding point  $x$ . Conceptual modeling frameworks, such as the ones listed in section 5, 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.

#### 4 DOCUMENTING THE CONCEPTUAL MODEL

As stated above, the conceptual model is not always explicitly expressed, but can remain within the mind of the modeler. That said, it is good practice to document the conceptual model and in so doing to provide a means of communication between all parties in a simulation study. There are no set standards for documenting discrete-event simulation conceptual models, but a range of approaches have been proposed, including (references to examples are provided in brackets):

- Component list (appendix)
- Process flow diagram (Robinson 2014)
- Activity cycle diagram (Robinson 2014)
- Logic flow diagram (Robinson 2014)
- List of assumptions and simplifications (appendix)
- Unified modeling language (UML) (Richter and März 2000)
- Petri nets (Torn 1981)
- Condition specification (Overstreet and Nance 1985)

The documentation for a conceptual model should be kept simple and focus on identifying *what* is to be modeled and *what* is not to be modeled. There is no need for elaborate documentation because detailed decisions about *how* to model something are not being made during conceptual modeling. These decisions are made while creating the model design from the conceptual model. For example, in conceptual modeling a decision is made to model a flight schedule at an airport; in model design the way in which that flight schedule is to be modeled is determined. Hence, the conceptual model documentation only needs to state that the flight schedule is to be modeled, while the model design documentation needs to provide the detail of how to model the flight schedule.

## 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, 2016)
- 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)
- Conceptual modeling with Onto-UML (Guizzardi and Wagner 2012)
- Conceptual modeling using the Structured Analysis and Design Technique (Ahmed, Robinson, and Tako 2014)
- The PartiSim framework (Tako and Kotiadis 2015)

For a more detailed discussion on conceptual modeling frameworks see Robinson et al. (2010). Here a very brief outline and illustration of Robinson's framework for conceptual modeling is given. For a more detailed account, and an illustration of the framework in use, see Robinson (2008b).

Figure 4 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)
- Identifying the model inputs (experimental factors)

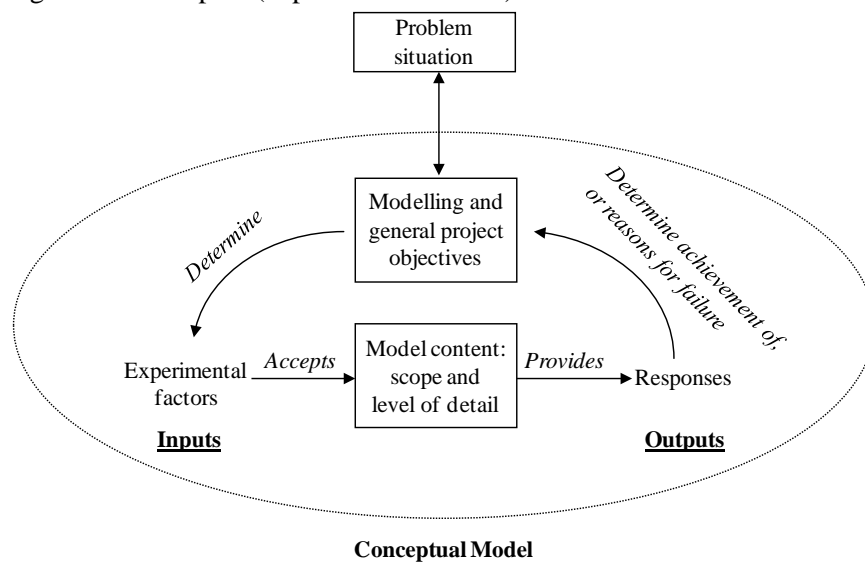


Figure 4: A framework for conceptual modeling (Robinson 2008b).

- 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)
- Model inputs/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 are useful for communicating the conceptual model (Robinson 2014).

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 in the appendix.

## **6 LEVELS OF ABSTRACTION**

We describe a conceptual model that involves many simplifications, and so is removed a long way from the system description, as a ‘far’ abstraction. 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. A good illustration of this is Schelling’s model of segregation, which contains no real world data and only the simplest possible representation of the phenomena under study (Schelling 1971). Although a far abstraction, Schelling’s model has certainly attracted a lot of attention.

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 have utility. 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 the part of the real world it is representing. 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 tutorial provides a definition of a conceptual model, its artefacts and its requirements; it also discusses the benefits and approaches to documenting the conceptual model. From this base, some frameworks for conceptual modeling are listed and Robinson's framework is outlined in more detail. The framework aims to guide a modeler through the activity of creating and documenting a conceptual model. We also discuss levels of abstraction, from near to far.

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

Sections of this paper are based on: Robinson, S. 2011 "Conceptual Modeling for Simulation." In *Encyclopedia of Operations Research and Management Science*, edited by J.J. Cochran. New York: Wiley; Robinson, S. 2011; Robinson, S., Arbez, G., Birta, L. Tolk, A. and Wagner, G. 2015 "Conceptual Modeling: Definition, Purpose and Benefits." In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 2812-2826. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

## A APPENDIX: EXAMPLE OF CONCEPTUAL MODEL TEMPLATE

This appendix provides an illustration of the conceptual model template that accompanies Robinson's conceptual modeling framework.

The problem: a fast food restaurant is experiencing problems with one of the branches in its network. Customers regularly complain about the length of time they have to queue at the service counters. It is apparent that this is not the result of shortages in food, but a shortage of service personnel.

### Project

Fast Food Restaurant
----------------------

### Modeling and General Project Objectives

#### *Organisational Aim*

To improve customer service levels
------------------------------------

#### *Modeling Objectives*

The number of service staff required during each period of the day to ensure that 95% of customers queue for less than 3 minutes for service. Due to space constraints, a maximum of 6 service staff can be employed at any one time.
---

*General Project Objectives*

Time-scale	5 working days
Flexibility	Limited (extensive changes unlikely)
Run-speed	Many experiments to be run
Visual display	Simple 2D
Ease-of-use	Use by modeler only

**Model Outputs/Responses**

*Outputs (to determine achievement of objectives)*

Percentage of customers queuing for less than 3 minutes
---

*Outputs (to determine reasons for failure to meet objectives)*

Bar chart of waiting time for each customer in the queues, mean, minimum and maximum Time-series of mean queue size by hour Staff utilisation (cumulative percentage)
---

**Experimental Factors**

Staff rosters (total number of staff in each hour of the day), varied over a range of 1 to 6
--

**Model Scope**

Component	Include/Exclude	Justification
<b>Entities:</b>		
Customers	Include	Flow through the service process
<b>Activities:</b>		
Service points	Include	Experimental factor, required for staff utilization response
Tables	Exclude	Not related to waiting for food
Cooking	Exclude	Assumption: material shortages are not a significant problem
Cleaning	Exclude	Not related to speed of service
<b>Queues:</b>		
Service point queues	Include	Required for waiting time and queue size response
Table queues	Exclude	Tables are not being modeled
Queues of food	Exclude	Assumption: material shortages are not a significant problem
<b>Resources:</b>		
Service staff	Exclude	Simplification: represented by service points
Kitchen staff	Exclude	Cooking not being modeled
Cleaning staff	Exclude	Cleaning not being modeled

**Model Level of Detail**

<b>Component</b>	<b>Detail</b>	<b>Include/ Exclude</b>	<b>Justification</b>
<b>Entities:</b> Customers	Quantity: 1 entity represents 1 customer group	Include	Simplification: removes need to model individual customers and then to group them Required to model customer demand
	Arrival pattern: inter-arrival time distribution varying over the day	Include	
	Attributes: size of customer order	Exclude	Simplification: represented in the service time
	Routing: to shortest queue	Include	Impacts on waiting time and queue size responses
<b>Activities:</b> Service points	Quantity: number available each hour of the day	Include	Experimental factor
	Nature (X in Y out)*	Exclude	Simple 1 in 1 out
	Cycle time: service time distribution (accounts for order size and variability in staff performance)	Include	Represents workload and so utilization of staff (response)
	Breakdowns/repairs	Exclude	Assumption: assume no breakdowns
	Set-up/changeover	Exclude	n/a
	Resources	Exclude	No resources modeled
	Shifts: number available each hour of the day	Include	Experimental factor
	Routing: to world	Include	Not modeling further activities in the restaurant e.g. tables
Other: Absenteeism	Exclude	Simplification: if required can be modeled by perturbations to the number of staff available	
<b>Queues:</b> Service point queues	Quantity: one for each service point	Include	Required for queuing responses
	Capacity: unlimited	Exclude	Simplification: no limit to number of people who can wait
	Dwell time	Exclude	n/a
	Queue discipline: FIFO	Include	Simplification: no pushing in to queues and no balking, jockeying or leaving from queues.
	Breakdown/repair	Exclude	n/a
Routing: to service point	Include	Flow of entities through the system	
<b>Resources:</b> n/a			

\* X refers to number of input items, Y to number of output items. For a simple assembly activity X=2 and Y=1.

## Modeling Assumptions

Material shortages are not a significant problem Assume no breakdowns of service points
--

## Model Simplifications

Service staff represented by the service points 1 entity represents 1 customer group Size of customer order represented in the service time distribution Absenteeism modeled through perturbations to the number of staff available No pushing in to queues, no balking, no jockeying or no leaving from queues No limit to number of people who can wait in a queue
---

## REFERENCES

- Ahmed, F., S. Robinson, and A. A. Tako. 2014. "Using the Structured Analysis and Design Technique (SADT) in Simulation Conceptual Modeling. In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S.D. Diallo, I.O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 1038-1049. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Arbez, G. and L.G. Birta 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.
- Arbez, G. and L.G. Birta. 2016. "A Tutorial on ABCmod: An Activity Based Discrete Event Conceptual Modelling Framework." In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 88-102. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- 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, 52-66. 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.
- Law, A.M. 2015. *Simulation Modeling and Analysis, 5<sup>th</sup> ed.* New York: McGraw-Hill.
- 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.
- Overstreet, M. C. and R. E. Nance. 1985. "A Specification Language to Assist in Analysis of Discrete Event Simulation Models." *Communications of the ACM* 28 (2): 190-201.
- Pidd, M. 2010. "Why Modelling and Model Use Matter." *Journal of the Operational Research Society* 61 (1): 14-24.

- Richter, H. and L. März. 2000. "Toward a Standard Process: The Use of UML for Designing Simulation Models." In *Proceedings of the 2000 Winter Simulation Conference*, edited by J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 394-398. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Robinson, S. 1999. "Simulation Verification, Validation and Confidence: A Tutorial." *Transactions of the Society for Computer Simulation International* 16 (2): 63-69.
- 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. 2014. *Simulation: The Practice of Model Development and Use*, 2<sup>nd</sup> ed. London, UK: Palgrave.
- Robinson, S., R. J. Brooks, K. Kotiadis, and D. J. van der Zee. 2010. *Conceptual Modelling for Discrete-Event Simulation*. FL, USA: Taylor and Francis.
- Robinson, S., G. Arbez, L. Birta, A. Tolk, and G. Wagner. 2015. "Conceptual Modeling: Definition, Purpose and Benefits." In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 2812-2826. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- 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. 2013. "Verification and Validation of Simulation Models." *Journal of Simulation* 7 (1): 12-24.
- Schelling, T. C. 1971. "Dynamic Models of Segregation." *Journal of Mathematical Sociology* 1: 143-186.
- Shannon, R.E. 1998. "Introduction to the Art and Science of Simulation." In *Proceedings of the Winter Simulation Conference 1998*, edited by D.J. Medeiros, E.F. Watson, M. Manivannan, J. Carson, 7-14. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Tako, A. A. and K. Kotiadis. 2015. "PartiSim: A Multi-Methodology Framework to Support Facilitated Simulation Modelling in Healthcare." *European Journal of Operational Research* 244(2): 555-564.
- Thomas, A. and P. Charpentier. 2005. "Reducing Simulation Models for Scheduling Manufacturing Facilities." *European Journal of Operational Research* 161 (1): 111-125.
- Torn, A. A. 1981. "Simulation Graphs: A General Tool for Modeling Simulation Designs." *Simulation* 37 (6): 187-194.
- 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.
- Zeigler, B. P. and T.G. Kim. 2000. *Theory of Modeling and Simulation*, 2<sup>nd</sup> ed. New York: Wiley.

## AUTHOR BIOGRAPHY

**STEWART ROBINSON** is Dean and Professor of Management Science 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 six 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 the UK Simulation Workshop conference series. He was President of the Op-

*Robinson*

erational Research Society (2014-2015). Home page: [www.stewartrobinson.co.uk](http://www.stewartrobinson.co.uk). His email is [s.l.robinson@lboro.ac.uk](mailto:s.l.robinson@lboro.ac.uk).