Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

CONCEPTUAL MODELING: DEFINITION, PURPOSE AND BENEFITS

Stewart Robinson

School of Business and Economics

Loughborough University
Loughborough, LE11 3TU UNITED KINGDOM

Gilbert Arbez
Louis G. Birta

School of Electrical Engineering and Computer Science
University of Ottawa
800 King Edward, P.O. Box 450, Stn A
Ottawa, Ontario, K1N 6N5 CANADA

Andreas Tolk

Computer Science Principal
The MITRE Corporation
J84B Simulation Engineering Department
Hampton, VA 23666 USA

Gerd Wagner

Brandenburg University of Technology
Institute of Informatics
P. O. Box 101344
03013 Cottbus GERMANY

ABSTRACT

Over the last decade there has been a growing interest in 'conceptual modeling' for simulation. This is signified by a greater intensity of research and volume of papers on the topic. What is becoming apparent, however, is that when it comes to conceptual modeling there are quite different views and opinions. These differences may be beneficial for creating a debate that takes the field forward, but they can also lead to confusion. The purpose of this panel is for leading researchers to identify and discuss their views on conceptual modeling. In particular we will debate the definition, purpose and benefits of conceptual modeling for the field of simulation. Through the discussion we hope to highlight common ground and key areas of difference.

1 INTRODUCTION

The growing interest in conceptual modeling for simulation is demonstrated by a more active research community in this domain. Over the last decade there has been an increase in the number of conference and journal papers on conceptual modeling, and an edited book on the topic (Robinson et al, 2010). Recent Winter Simulation Conferences have even included introductory and advanced tutorials on conceptual modeling.

A closer look at this work shows that there are quite different opinions about the nature, purpose and benefits of conceptual modeling in simulation. Although such differences are quite normal for an emerging, or even established, field, they can lead to confusion. So rather than continue a debate through the pages of individuals' publications, the purpose of this paper is to draw together in a single paper a range of perspectives on conceptual modeling. We do not aim to reconcile the differences in opinion, but simply to express our different views and so highlight where our perspectives differ. The ensuing panel discussion aims to help us understand how and why our views differ as the basis for moving towards an accommodation of views.

978-1-4673-9743-8/15/$31.00 ©2015 IEEE

In this paper five leading researchers in conceptual modeling for simulation identify their perspective on the topic: Gilbert Arbez and Lou Birta (University of Ottawa), Stewart Robinson (Loughborough University), Andreas Tolk (MITRE Corp.) and Gerd Wagner (Brandenburg University of Technology). Each was asked to answer the following three questions:

1. What is a conceptual model? (and how does the conceptual model relate to the real world (problem) and to the computer model?)
2. What is the purpose of conceptual modeling?
3. What are the benefits of a conceptual model?

The perspectives are now presented in alphabetic order of last name.

2 PERSPECTIVE OF ARBEZ AND BIRTA

We explore the notion of a conceptual model within the framework of a modeling and simulation *project*. This project framework embraces two key notions; first there is the notion of a "system context"; i.e., there is a system that has been identified for investigation, and second, there is a problem relating to the identified system that needs to be solved. Obtaining an acceptable solution to this problem is the purpose of the modeling and simulation project. We use the term "system" in its broadest possible sense; it could, for example, include the notions of a process or a phenomenon. Furthermore, physical existence of the system is not a prerequisite; the system in question may simply be a concept, idea or proposal. What is a prerequisite, however, is the requirement that the system in question exhibits "behavior over time"; in other words, that it be a *dynamic* system. We refer to such a system as the system under investigation (SUI).

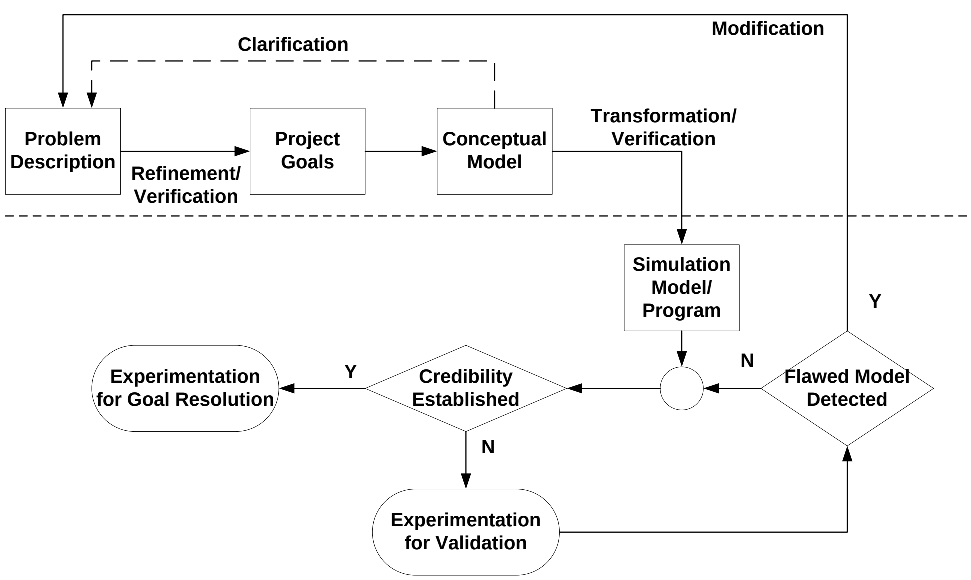

Figure 1: Stages of a modeling and simulation study.

2.1 The Conceptual Model – a Definition

An overview of the process for a modeling and simulation study is provided in Figure 1. It needs to be emphasized that a modeling and simulation project of even modest size is often carried out by a team of professionals where each member of the team typically contributes some special expertise. There is, therefore, a need for effective communication among team members. Some facets of the discussion have their basis in this communication requirement.

Four main artefacts (the rectangles shown in Figure 1) are produced during the modeling process. Experimentation makes use of the simulation program artefact to arrive at a goal resolution. Experimentation is also used to validate the model. We begin with, a brief overview of the two prerequisite artefacts for conceptual model development .

- Problem Description: The first important task of the modeling and simulation team is the development of a document called the problem description. This key document evolves from the elaboration of the rudimentary information received from the client in whatever ways are deemed essential by the project team. The intent of the elaboration process is to ensure that the problem description document can serve as a meaningful foundation for the initiation of a modeling and simulation study. However the document is not static but rather is subject to refinement as the model building effort progresses. Nevertheless its initial rendition must have sufficient content to enable the initiation of the model development phase.
- Project Goals: We regard the formulation of goals for the project to be the first step in the refinement process that transforms the problem description into a conceptual model. Our perspective is that the achievement of some set of clearly defined project goals will hopefully coincide with the problem solution. This may however be overly idealistic since their achievement may merely contribute to, rather than constitute, the problem solution. Goals are typically stated in terms of parameters and/or policy options (including details of the experimentation during which these are manipulated) together with output variables observed during experimentation. The project goals have a fundamental impact upon model development and therefore must be regarded as a prerequisite to that development process.

The information provided in the problem description is, for the most part, unstructured and relatively informal. A refinement phase that enhances precision and completeness is generally required. This typically involves introducing variables and parameters to facilitate the introduction of detail, and the use of formalisms wherever helpful. Notice that a corollary to this assertion is the fact that there may be facets of the SUI that are purposely ignored because they are judged to be devoid of relevance. The result of this refinement process is called the *Conceptual Model* for the modeling and simulation project.

The conceptual model is *a concise and precise consolidation of all goal-relevant structural and behavioral features of the SUI presented in a predefined format*. It provides foundation for the development of the simulation program.

As shown in Figure 1, the conceptual model is subsequently transformed into an alternate representation that properly respects the syntax and semantic constraints of some programming language or more generally, a simulation program development environment. The result of this transformation is the simulation program. The simulation program is used in the experimentation phases shown in Figure 1 for the purposes of model validation and goal resolution.

2.2 Purpose of a Conceptual Model

Apart from the fundamental requirement to capture the essential behavioral features of the SUI, there are two important qualities that must be addressed in the construction of any conceptual model, namely:

- The conceptual model must be sufficiently transparent so that all stakeholders in the project are comfortable in using it as a means for discussing those mechanisms within the SUI that have relevance to the characterization of its behavior (as interpreted from the perspective of project goals).

- The conceptual model must be sufficiently comprehensive so that it can serve as a specification for developing a computer program, namely the simulation program, that will provide the means for carrying out the simulation study. The simulation program is a software product and its development relies on considerable precision in the statement of requirements. One of the important purposes of the conceptual model is to provide the prerequisite guidance for the software development task.

Both of these requirements place constraints on the format which is used to formulate the conceptual model. Indeed these constraints are, to some extent, conflicting. The second constraint clearly implies a high level of clarity and precision. But such a representation may be regarded as tedious by a sizable community of the stakeholders in the project. Thus care is needed in developing a framework that balances, but nevertheless accommodates, these requirements.

An effective approach for dealing with this dual requirement is illustrated in the ABCmod framework (Activity-Based Conceptual modeling) (Birta and Arbez 2013) in which the conceptual model evolves in two stages. The purpose of the initial "high level" stage is to provide a preliminary perspective for the model that is unencumbered by excessive detail and is primarily intended for discussion amongst the project stakeholders. The subsequent "detailed level" stage elaborates upon structural and behavioral aspects with a view toward creating a representation that can serve as a specification for a simulation program.

2.3 Benefits of a Conceptual Model

The benefits of a conceptual model can best be examined by considering the alternative of not having a conceptual model. The implication here is that the code development for the simulation model within some particular program development environment begins immediately following the presentation of a problem statement.

Normally a modeling and simulation project involves a variety of individuals who have a keen interest in the project but possibly from different perspectives and with different domains of expertise (e.g. problem domain experts and M&S experts). This group (the stakeholders) typically have varying levels of understanding of the programming environment (possibly none at all). This implies that the expectations and/or insights of many of these individuals will be compromised since they are unable to make a meaningful assessment of the simulation model that is under development. In other words they are effectively "sidelined" and consequently the group cannot come to an understanding of the model to be developed because the general discussion framework that a conceptual model provides, is non-existent. What then are the consequences? Some are listed below:

- In effect the goals of the study are never clearly identified. Consequently insightful discussion and agreement by the stakeholders does not take place. How then is it possible to decide when the project has been successfully completed?
- The problem description rarely contains sufficient information to make key decisions about the level of granularity that is appropriate for specifying behavior characteristics. Such insights rest with the stakeholders but in the absence of a conceptual model these decisions, by and large, become relegated to the program developers.
- Because of their varying perspectives, the stakeholders likely have a range of views on what constitutes important/relevant output. Without the context of a conceptual model, decisions about output are superficially made by program developers and not necessarily aligned with the expectations of the stakeholders.
- The nature of the experiments to be carried out with the simulation model are often intertwined with ranges of parameter values that need to be explored. Identification of these ranges of significance can only be made by the stakeholders and without a conceptual model this critical

aspect of the study is ill-defined and improper decisions could undermine the hoped for outcome of the study.

- The validation of a simulation model is critical in terms of establishing the credibility of the results that flow from the M&S study. This is not an easy task and effective strategies are typically founded upon the expertise of the stakeholders, but in the absence of a conceptual modeling framework, there is no platform for the presentation of these insights.
- When model development rests with the programming team, there is considerable likelihood that SUI behavior features will be significantly biased to programming constructs rather than genuinely reflecting properties of the SUI.

3 PERSPECTIVE OF ROBINSON

In opening this discussion, first I would like to answer the question: where is the model? Figure 2 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. My perspective is 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.

3.1 What is 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, 2008). This definition establishes four facets of a conceptual model, as follows.

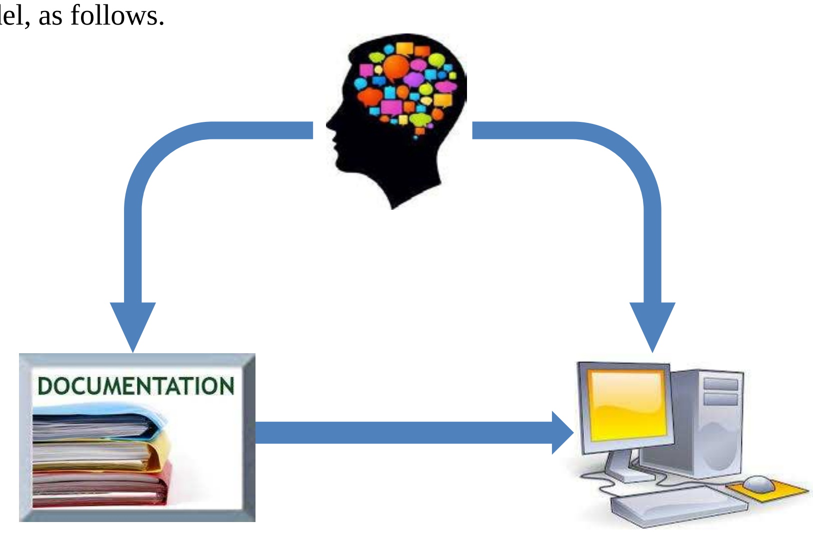

Figure 2: Where is the model?

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. 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.

Figure 3 describes how conceptual modeling fits within the wider context of the modeling process for simulation by showing the key artifacts 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 artifacts of the (conceptual) modeling process. These artifacts, which are quite separate, 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 above.
- 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.

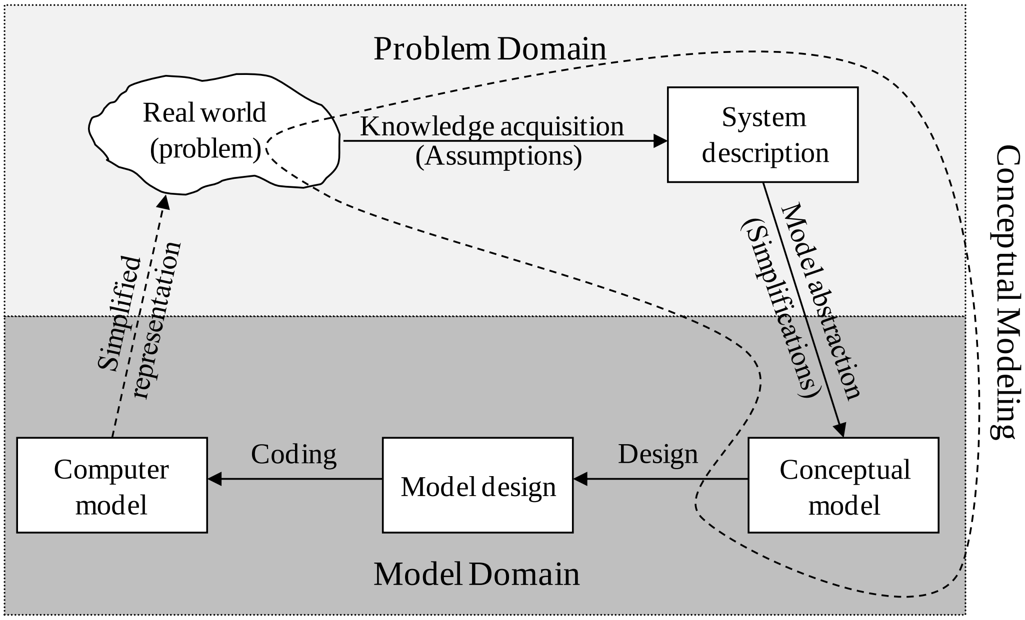

Figure 3: Artifacts of conceptual modeling (Robinson, 2010, 2013, 2014).

The arrows show how information flows between these four artifacts. The activities that drive the flows of information are described as knowledge acquisition, model abstraction, design and coding. Meanwhile, the specific and different roles of assumptions and simplifications are highlighted on Figure 3. 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 model design provides the technical details for how the conceptual model can be converted into the chosen language or software package for the computer model; it is software specific. The computer model is the executable code developed in the chosen language or software package. 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 artifacts are included on 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 artifacts are independent of the software that is used for developing the simulation model.

Finally, 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; it describes those parts of the system description that are included in the simulation model and at what level of detail. These two artifacts are often confused and seen as indistinct. Indeed, a major failure in any simulation project is to try and model the system description (everything that is known about the real system) and to not attempt any form of model abstraction; this leads to inappropriately complex models.

3.2 What is the Purpose 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.

3.3 What are the Benefits of a Conceptual Model?

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

To conclude, the conceptual model exists in the mind of the modeler. The documentation and model code are means for making that model explicit.

4 PERSPECTIVE OF TOLK: CONCEPTUAL MODELING AS A GENERAL METHOD

The purpose of this section is to broaden the view on the definition, purpose, and benefits of conceptual modeling beyond the tradition application domains and phases in the life cycle of a simulation system.

Modeling & Simulation (M&S) comprises two important activity categories of equal importance: modeling activities and simulation activities. Each activity category is made up of several phases, and where exactly conceptual modeling ends and other modeling activities begin, and how we define these tasks and deliverables is subject of many discussions. In this contribution, by looking at the various phases two objectives shall be accomplished that seem to be mutual exclusive at the first look: (a) focusing the definition of conceptual modeling, and (b) generalizing its application.

Our community seem to agree that the modeling activities address several levels of abstraction. Federici, Redaelli, and Vizzari (2006) describe four of these levels that will help to guide the discussion. They introduce the target system, the abstract model, the computational model, and the software model. The target system is driven by the task to be conducted and identifies the important subset of the known universe that needs to be addressed in order to address the task to be conducted. The abstract model is a collection of elements constitutive of the target system, comprising how the elements are causally related. The computational model introduces specific computation-specific interpretations and elements, including choosing the best applicable set of modeling paradigms. Finally, the software model implements these ideas, addressing language and platform. This idea is very close to the well-known framework introduced in Zeigler, Praehofer, and Kim (2000): source system, experimental frame, model, and simulator.

The common understanding seems to be that we start with a conceptual model, then derive a computational model, and finally implement the software model. However, conceptual modeling will be shown to be applicable for reliability, visualization, and data farming as well. By showing how the insights from conceptual modeling will be applied as own activities this section motivates the feasibility and necessity to generalize conceptual modeling, which is not limited to the initial phase of a simulation project but needs to be applied through all life-cycle phases.

4.1 Conceptual Modeling in Preparation for Simulation Program Development

Fishwick (2007) presents examples of the many modeling paradigms regarding modeling methodologies – such as discrete event systems, system dynamics, or agent based approaches – and model types – such as ordinary differential equations, process algebra, temporal logic, etc. All these activities have in common that modeling is the task–driven, purposeful simplification and abstraction of a perception of reality that is shaped by physical and cognitive constraints, leading to a conceptualization of the relevant subset of the problem domain (Tolk 2013). This definition of modeling deserves some attention to the details. First, the modeling team uses a perception of reality that is shaped by access constraints, but also by the cognitive capabilities. The access to the data the team would like to obtain describing the system, can be constrained by many circumstances, including ethical and legal barriers when data about humans is collected (Anderson 2007), or physical barriers that do not allow to observe certain internal behavior without destroying the system in the process, or the system is simply still not constructed and only exists as an idea itself. The other component is that the team will have a certain scientific theory in mind that represents the best theory currently available. However, science itself is a series of perpetually improving models of reality (Goldman 2006). In addition to this fundamental barrier – that we can only model something relative to what we know as scientists – different team members will have different degrees of knowledge of the domain. What often is referred to as reality is therefore not necessarily the same for all teams, and may not even be the same within the team. Second, the team uses this perception of reality that already is shaped by non-purposeful constraints to further simplify and abstract the model to support the given task. Simplification is the essence of reductionism: it is focusing on only the important elements and discard the rest. What is perceived to be important is defined by the task. The team purposefully focuses on the elements needed by cutting everything away that is not essential. In the same way, all things that influence the important elements are abstracted into something simpler that provides just the functionality needed by the important parts. How and if the chosen modeling paradigm will significantly shape the conceptual model is an open question of research, but at least it will shape the cognitive aspects. The idiom “if all you have is a hammer, everything looks like a nail” may be applicable here as well for simulationists who only are educated in one modeling paradigm and type.

In any case, modeling results in a conceptualization of the problem domain. A conceptual model therefore captures the results of the modeling process regarding the concepts and their relations that describe the problem, which have to include assumptions and constraints that shape the process and the results. In earlier work (Tolk et al. 2010) we already recommended that this should be done as a formal specification of the conceptualization, which is easily recognized as the definition for an ontology introduced by Gruber (1993). Computational and simulation modeling activities add specificity, but conceptually they only add detail, they should not change the conceptual ontological structure.

4.2 Conceptual Modeling supporting Reliability and Trust

Computational and simulation modeling are not straightforward processes and require many implementation choices, all of which result in compromises. Oberkampf et al. (2002) identify mathematical modeling, discretization and algorithm selection, computer programming, and numerical solutions as pivotal activities that all contribute to potential errors and uncertainties. They show how the choice of partial differential equations, auxiliary physical equations, boundary and initial conditions, and non-deterministic representations contribute to uncertainties and errors during mathematical modeling activity. They show that conceptual modeling is pivotal to provide the basis for reliability and trust, as it establishes the foundation for all future steps. A formal representation of the conceptualization is therefore necessary to provide this solid foundation for following implementation steps as well as for reuse or composition with other simulation-based solutions (Tolk et al. 2010).

4.3 Visualization and Data Farming

To make the case that the more specific definition of a conceptual model as a formal specification of the conceptualization of the problem domain can and should be generalized, let us focus on two important phases in the life cycle of a simulation: visualizing the results and using simulation for data farming. In both cases, we are applying a new layer of modeling to our simulation.

The importance of visualization has been emphasized for long, see, e.g., (Rohrer 1997). However, when visualizing a simulation, we select a subset of what we simulated, focusing on the important aspects and visualizing them in the context understood to be important for the task the customer is interested in. In other words, visualization is a task-driven, purposeful simplification and abstraction of the simulation data that is accessible. Visualization is therefore modeling, and as we are focusing on showing the important concepts interacting with each other, conceptual modeling results are of pivotal importance for visualization. Therefore, visualization should never be an afterthought in the process, but an integrated part already supporting the first phases of modeling. Many users look at the visualization as the representation of the simulation system. If something is important enough to be visualized, it is important enough to be conceptually modeled.

The same is true when simulations are used for data farming: only if we design our simulation from the conceptual modeling phase on to be able to produce the data we need to address in our data farming challenge, the project can be successful. If something is not considered during the conceptual modeling phase, it cannot be introduced in hindsight. This also implies that the conceptual model needs to be available – and machine readable – for such applications.

4.4 Definition, Purpose, and Benefits of Conceptual Modeling: A Broader View

Within this section, a conceptual model has been defined as the result of the processes leading from the task to the specification of the conceptualization of the ontological structure of the problem domain, comprising assumptions and constraints relevant to all relevant modeling decisions. The result shall be captured as a formal representation, which allows machine support and supports a better understanding of errors and uncertainty.

The purpose is to capture and communicate the conceptualization with intended as well as potentially unforeseen simulation users, which includes providers of visualization tools and data farmers. It is therefore important to realize that conceptual modeling is not a limited activity in the initialization phase of a simulation system, but a perpetually reoccurring process that drives the design of experimentation, the providing of necessary data, the evaluation and presentation of results, and many more activities conducted within the life cycle of the system.

The benefit is building trust by unambiguously documenting the model – which is the foundation of the resulting simulation – which is pivotal in case of reuse or composition, as discussed in more detail in Tolk (2013). Without the unambiguous documentation of conceptualizations for the various activities over the life cycle, it is highly likely that inconsistencies will compromise the interpretation of the simulation results, which leads to insufficient training, ambiguous research results, and suboptimal decisions.

5 PERSPECTIVE OF WAGNER

5.1 What is a Conceptual Model?

Historically, research in conceptual modeling has first been carried out in the computer science field of Database Systems. It started with two proposals for a conceptual data modeling language: the semantic model proposed by Abrial (1974) and the entity-relationship (ER) model proposed by Chen (1976), which triggered the series of ER conferences (ER 2012) starting in 1979. Later it was noticed that conceptual modeling, e.g., in the forms of enterprise modeling and business process modeling, plays an important role in software engineering, in general.

Today, in the field of Information Systems (IS) and Software Engineering (SE), there is a widely accepted distinction between three kinds of models as engineering artifacts resulting from corresponding activities in the analysis, design and implementation phases: solution-independent domain models, platform-independent design models, and platform-specific implementation models.

The term "conceptual model" (CM) has been used differently by researchers in Modeling and Simulation (M&S) and by researchers in IS/SE. While in IS/SE, there is an agreement that a conceptual model is the result of modeling a real world problem domain independently of the solution design, and is therefore also called a domain model, there is no such agreement in M&S, and several authors, e.g., Robinson (2013), suggest that a conceptual model is the result of making a solution design independently of a specific simulation technology platform.

In a previous WSC panel discussion on conceptual modeling, (Zee et al. 2010), three different definitions of what is a CM have been proposed:

1. A *document that states "what you will and will not include in the simulation and why"*, or "a repository of high-level conceptual constructs and knowledge specified in a variety of communicative forms (e.g., animation, audio, chart, diagram, drawing, equation, graph, image, text, and video)" intended to assist in the design of a simulation, as proposed by Balci et al. (2008);
2. *"A formal specification of a conceptualization"*, or "an ontological representation of the simulation that implements it" as proposed in Turnitsa et al. (2010);
3. *"*The specification of an executable simulation model*"*, or "a non-software specific description of the computer simulation model" as proposed by Robinson (2008).

The first definition, by Balci et al., identifies a CM with an informal requirements definition, while ignoring the important question of using a CM that is expressed in a (preferably diagrammatic) modeling language with a well-defined semantics. The second definition, by Turnitsa et al., roughly corresponds to what is called a domain model in IS/SE. Finally, the third definition, by Robinson, corresponds to what is called a design model in IS/SE.

I think that the question of what is a CM subsumes three related issues or sub-questions:

1. Which types of models should we use in the development of a computer simulation?
2. How should we call these types of models?
3. Which modeling languages can be used for making these types of models?

The really important questions are 1 and 3, while 2 is a purely terminological issue. Considering question 1, it seems pretty clear to me that since simulation engineering is just a special case of SE, we have to deal with the same types of models as in SE, namely with models they call domain, design and implementation models. I don't see any benefit in using different names than they do. On the contrary, this would just create confusion. Since in IS/SE, they often call domain models conceptual models, because they describe the concepts used by subject matter experts, this is the terminology I would prefer to be also used in M&S.

Adopting this approach implies that a conceptual simulation model is a solution-independent description of a real world problem domain, from which a platform-independent simulation design model can be derived for a given set of research questions, such that this design model is, in turn, the basis for coding executable simulation models, which are computer programs tailored towards specific simulation platforms.

Table 1: A summary of the perspectives on conceptual modeling.

Conceptual model	Arbez and Birta	Robinson	Tolk	Wagner
Definition	The conceptual model is a concise and precise consolidation of all goal-relevant structural and behavioral features of the SUI presented in a predefined format.	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.	Result of the processes leading from the task to the specification of the conceptualization of the ontological structure of the problem domain, comprising assumptions and constraints on all relevant modeling decisions.	A solution-independent description of a real-world problem domain, from which a platform-independent simulation design model can be derived for a given set of research questions.
Purpose	The conceptual model should enable all stake holders to discuss the SUI's behavior and it must be sufficiently comprehensive to serve as a specification for a computer program.	A simulation model cannot exist without a conceptual model. A documented conceptual model is for communication.	Capturing and communicating the conceptualization with intended as well as potentially unforeseen simulation users.	To capture a sufficiently large, and sufficiently complete, part of the real world problem domain, for which a simulation study is to be performed, in such a way that all kinds of research questions concerning this domain can be investigated.
Benefits	The conceptual model ensures that key SUI features (e.g. behavior, granularity) evolve from discussion with all stakeholders rather than from a programming bias.	A documented conceptual model is the basis for guiding all activities in simulation model development and use.	Building trust by unambiguously documenting the model – which is the foundation of the resulting simulation – which is pivotal in case of reuse or composition.	The CM can help to clarify questions about the scope and purpose of a simulation project, and it is an asset that can be re-used for making different solution designs for different research questions.

This leaves us with question 3: which modeling languages can or should we use for making conceptual models? While the answer is pretty clear for conceptual information models, it is less clear for conceptual process models.

The most advanced and most widely used language for information modeling is the language of UML Class Diagrams. There is no reason, why UML Class Diagrams should not be used for conceptual information modeling in simulation engineering. As shown in Wagner (2014) , they allow to model both object types and event types, and also allow to model randomness.

Concerning conceptual process modeling, we have the problem that the most widely used language for modeling processes based on discrete events, the Business Process Modeling Notation (BPMN), has an official semantics that is not adequate for discrete event simulation, as explained in Wagner (2014). However, this does not imply that BPMN could not be used for conceptual process modeling in simulation engineering. Rather, it implies a need for developing an alternative semantics.

5.2 What is the Purpose of Conceptual Modeling?

The purpose of conceptual modeling is to capture a sufficiently large, and in some sense sufficiently complete, part of the real world problem domain, for which a simulation study is to be performed, in such a way that all kinds of research questions concerning this domain can be investigated by first deriving suitable simulation design models from the conceptual simulation model, and then implementing these design models with the help of one or more simulation platforms.

5.3 What are the Benefits of a Conceptual Model?

The main benefits of first making a CM as a solution-independent description of a real world problem domain with the help of one or more CM languages, instead of directly making a simulation design model for a given research question, or instead of directly coding the mental simulation model that the simulation developer has in her mind for a specific simulation platform, are:

- The CM can help to clarify questions about the scope and purpose of a simulation project (e.g., in discussions with the customer or with subject matter experts).
- The CM is an asset that can be re-used for making different solution designs for different research questions, either in the same or in different simulation projects.

6 CONCLUSION

Table 1 summarizes the four perspectives on conceptual modeling discussed in this paper with respect to the purpose, definition and benefits of a conceptual model. As is obvious from the discussion above, the contributors express quite a range of views. By drawing these differing views together we aim to help conceptual modeling researchers better understand the similarities and differences in opinion. Over time, the community needs to move towards a more unified, or at least accommodated, view of to the definition, purpose and benefits of conceptual modeling.

REFERENCES

Abrial, J-R. 1974. “Data Semantics.” In: Klimbie and Koffeman (eds.), Data Management Systems, North-Holland.

Anderson, J. G. 2007. “Social, Ethical and Legal Barriers to e–Health.” International Journal of Medical Informatics 76(5): 480–483.

Balci, O., J.D. Arthur, and R.E. Nance. 2008. “Accomplishing Reuse with a Simulation Conceptual Model.” In Proceedings of the 2008 Winter Simulation Conference, edited by S. J. Mason, R. Hill, L. Moench, and O. Rose, 959-965. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Birta, L. G. and G. Arbez. 2013. Modelling and Simulation: Exploring Dynamic System Behaviour. 2nd ed. London: Springer-Verlag.

Chen, P. 1976. “The Entity-Relationship Model: Towards a Unified View of Data.” ACM Transactions on Database Systems 1(1).

ER (International Conference on Conceptual Modeling). 2012. http://www.conceptualmodeling.org/. Accessed May 23, 2012.

Federici, M.L., S. Redaelli, and G. Vizzari. 2006. "Models, Abstractions and Phases in Multi-Agent Based Simulation." Proceedings CEUR Workshop 204: 144-150.

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

Fishwick, P. A. 2007. Handbook of Dynamic System Modeling. Boca Raton, Florida: Chapman & Hall/CRC Press, Taylor and Francis Group.

Goldman, S.L. 2006. Science Wars: What Scientists Know and How They Know It. Lehigh University, Chantilly, Virginia: The Teaching Company.

Gruber, T. R. 1993. "A Translation Approach to Portable Ontology Specifications." Knowledge Acquisition 5(2): 199-220.

Oberkampf, W. L., S. M. DeLand, B. M. Rutherford, K. V. Diegert, and K. F. Alvin. 2002. "Error and Uncertainty in Modeling and Simulation." Reliability Engineering & System Safety 75(3): 333-357.

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

Robinson, S. 2010. "Conceptual Modeling for Simulation." In Encyclopedia of Operations Research and Management Science, edited by J.J. Cochran. New York Wiley.

Robinson, S. 2013. "Conceptual Modeling for Simulation." In Proceedings of the 2013 Winter Simulation Conference, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 377-388. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Robinson, S. 2014. Simulation: The Practice of Model Development and Use. 2nd ed. London: Palgrave Macmillan.

Robinson, S., R.J. Brooks, K. Kotiadis, and D.-J. van der Zee. 2010. Conceptual Modeling for Discrete-Event Simulation. Boca Raton, FL, USA: Chapman and Hall/CRC.

Rohrer, M. 1997. "Seeing is Believing: The Importance of Visualization in Manufacturing Simulation." IIE Solutions 29(5): 24-28.

Tolk, A., S. Y. Diallo, R. D. King, C. D. Turnitsa, and J. J. Padilla. 2010. "Conceptual Modeling for Composition of Model-Based Complex Systems." In Conceptual Modeling for Discrete-Event Simulation, 355-381, Boca Raton, Florida: CRC Press, Taylor and Francis Group.

Tolk, A. 2013. "Interoperability, Composability, and their Implications for Distributed Simulation: Towards Mathematical Foundations of Simulation Interoperability." In Proceedings of the 2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications, IEEE Computer Society, 3–9.

Turnitsa, C., J.J.. Padilla, and A. Tolk. 2010. "Ontology for Modeling and Simulation." In Proceedings of 2010 Winter Simulation Conference, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan and E. Yücesan, 643−651. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Wagner. G. 2014. "Tutorial: Information and Process Modeling for Simulation." In Proceedings of the 2014 Winter Simulation Conference, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley and J. A. Miller, 103–117. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zee, D.-J. van der, K. Kotiadis, A.A. Tako, M. Pidd, O. Balci, A. Tolk, and M. Elder. 2010. "Panel Discussion: Education on Conceptual Modeling for Simulation – Challenging the Art." In Proceedings of 2010 Winter Simulation Conference, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan and E. Yücesan, 290−304. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zeigler, B. P., H. Praehofer, and T. G. Kim. 2000. Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems. San Diego, California: Academic Press.

AUTHOR BIOGRAPHIES

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 six books on simulation. His research focuses on the practice of simulation model development and use. Key areas of interest are conceptual modeling, 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.

GILBERT ARBEZ is an Assistant Professor at the School of Electrical Engineering and Computer Science (EECS). He has been teaching university courses since 1994 and became a full time teacher at EECS in 2003. Part of his teaching load includes a course on computer modeling and simulation fundamentals. He has co-authored a book on Modeling and Simulation fundamentals which features a conceptual modeling framework; he has subsequently contributed to a chapter in a conceptual modeling book and contributed a number of papers on the topic. Current interests include modeling and simulation of discrete event dynamic systems and continuous systems as well as teaching in engineering. He is starting an exploration of problem based learning and its application to engineering curriculum and courses. When he joined EECS, he brought with him 20 years of information technology experience gained from industry and government projects in various areas. Home page: http://engineering.uottawa.ca/eecs/people/arbez-gilbert. His e-mail is garbez@uottawa.ca.

LOUIS BIRTA is an Emeritus Professor in the School of Electrical Engineering and Computer Science at the University of Ottawa. His research career has focused on a broad range of topics within the modeling and simulation domain. Included here are parallel methods for continuous system simulation, optimization, validation of simulation models and conceptual modeling. He has, as well, contributed to the formulation of a "body of knowledge" for the discipline and also to the development of a code of ethics. He was, for several years, a member of the Board of the Society for Modeling and Simulation (SCS), has served as editor for the SCS journal SIMULATION, and was founding editor of the SCS M&S Magazine. He is a co-author (with Gilbert Arbez) of an introductory book (second edition) on modeling and simulation and is currently Senior Editor of the recently initiated Springer book series called Simulation Foundations, Methods and Applications. His email address is: LBirta@uottawa.ca.

ANDREAS TOLK is Computer Science Principal in the Simulation Engineering Department of MITRE Corp., Hampton VA USA. He received a PhD and a Master degree in Computer Science, both from the University of the Federal Armed Forces in Munich, Germany. He received the Outstanding Professional Contribution and the Distinguished Professional Achievements Award from Society for Modeling and Simulation (SCS) for his contributions to interoperability and composability of model-based systems. His email is atolk@mitre.org.

GERD WAGNER is Professor of Internet Technology at Brandenburg University of Technology, Germany. His research interests include (agent-based) modeling and simulation, foundational ontologies, knowledge representation and web engineering. His email address is G.Wagner@b-tu.de.