Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds

CONCEPTUAL MODELING WITH PROCESSES

Andreas Tolk

Old Dominion University
Engineering Management & Systems Engineering
Norfolk, VA 23549, USA

Charles Turnitsa

Columbus State University
TSYS School for Computer Science
Columbus, GA 31907, USA

ABSTRACT

Western philosophy of science has been heavily influenced by the idea that substantials are the main carriers of knowledge. Objects and their attributes and their relations to other objects dominate the world of knowledge representation. Processes play a subordinated role as they are merely seen as the things that create, change, or destroy objects. A recent study has shown that this view is dominant in modeling and simulation as well. The paper presents the (semi-) formal method developed in the doctoral study and its application to conceptual modeling techniques as they are taught in M&S education. The result shows that objects and relations are well captured, but that processes can be used as an alternative viewpoint as well. Using a process driven viewpoint opens new conceptual insights. We show that using the formal method allows to extend legacy conceptual methods to address these new aspects as well.

1 INTRODUCTION

Computer simulations are programs and as such they are influenced by computer engineering methods and tools. Due to their discrete nature, digital computers are closely connected with the idea of finite state machine, which are abstracts machines that can be in one of several finite states. In order to change the state, an event occurs that changes the states. Several useful modeling techniques have been developed in the recent years, and we will have a look at a selected subset in this paper, but we will also show that the entity or the state machine or the substantial is in the canter of all these methods while the events or processes always play a secondary role: the state is, the process or event only exist to change the state.

This idea is deeply rooted in Western philosophy, going back Plato and to Aristotle (350BC). In their philosophy, they postulated that substantials are the essence of what is real. What is can be described by properties and relations. As properties and relations can change, processes can be associated with such changes, but a process that doesn't change anything is meaningless. A process can also not create something out of nothing. Processes are agents of change, and the substantials are what are changed, and ultimately substantials are describing what is real. Reality is captured in matter and as such timeless. Processes change matter, but not reality. Aristotle used these basic ideas to constitute the beginnings of classification schemes to categories things and applying scientific methods to gain new knowledge. Ontology describes things and properties and how they are related. Substance has primacy over processes.

Only recently, the possibility to look at substantials and processes as different sides of the same coin has been discussed. Process philosophy states that processes have primacy over substantials, as a current stat is just a configuration of the underlying process, and while the processes endure, the substantials are changed by them (Rescher 1996).

The perspective of processes having primacy over substance allows for many interesting new perspectives, often reducing the complexity in methods used to describe a problem. A rainstorm with lightening, rain, and hail as a process is easier to describe as using the substantials to describe the storm. Styhre (2002) presents some interesting examples for strategic management challenges.

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

A recent PhD thesis completed at Old Dominion University therefore evaluated if current modeling methods support process philosophical views sufficiently, what elements are potentially missing, and what can be gained if we complement our state-machine centric methods with such alternative approaches. This paper describes the modeling techniques evaluated, analyzes the functional components of these methods, and presents the Object–Process–Relationship (OPR) method and its applicability.

2 MODELING AND MODELING TECHNIQUES

From a number of different sources, a distillation of the term modeling emerges in (Turnitsa 2012) that may be useful to consider here: *Modeling is the purposeful process of abstracting and theorizing about a system, and capturing the resulting concepts and relations in a conceptual model.*

From that definition, there are a number of implications that bear pointing out for this paper. First, in order for a captured abstraction that describes a system to be shareable with other parties, it is helpful to rely on methods (modeling techniques) that are understood within the community of practice the modeler operates in. Second, if such a technique is relied on to capture a model, it should have some capability of representing information about the system the model is an abstraction of. The combination of these two implications – (1) that modeling is done best with an established modeling technique, and (2) that a modeling technique is designed to capture knowledge about the system it represents – are key to the topic of this paper. The reasons for modeling are many, but in the case of conceptual modeling (the type of modeling being discussed in this paper, in spite of using the more general term of simply ‘modeling’), a very good case for when it is appropriate – especially when used in concert with a simulation – can be found in (Law and Kelton 2000). It illustrates that constructing a conceptual model is done when there are questions that need to be answered about the system to be modeled, but there is not a convenient way to use either the system itself, or some other representation.

2.1 Models as Representation of a System

In the literature on modeling, there are several views of what use a model should be described for. These are explained in detail in (Turnitsa 2012), however in brief they are as follows. The first, very commonly held, view is that a model exists in order to be the inspiration and design guidance for a simulation. This is seen, as an example, in (Robinson 2008). The second, almost as commonly held, view is that the model exists not only to serve as the inspiration for a simulation, but also for other purposes – for instance, subjecting the resulting simulation to verification and validation tests, or to stand in place of the system for analysis and optimization. Two sources for this view, as examples, are (Sargent 2005) and (Balci 1998). Finally, there is also a school of thought that expects a model to be the source of a number of different developments (not only a single simulation), and to be put to a variety of uses, not foreseen when the model is created. For this reason, the model should be as robust as possible in capturing knowledge about the system it describes. An example of this viewpoint is in (West 1996).

In all these cases, and others, the model itself stands in as a representation of the system. The model is intended to capture information about the system. Summarizing from (Robinson 2008) it captures information about the system’s “objectives, inputs, outputs, content, assumptions and simplifications”. From (King 2009) it has been shown that the assumptions and simplifications of a model are knowledge that is separate from what the model itself explicitly shows. However, the other elements (especially, inputs, outputs and content – which is how the inputs become transformed, over time, into the outputs) are of interest, and are what the model must capture, in order to be an abstraction of the system.

2.2 From Model to Simulation

Although a model, as representation of a system, should describe what that system is, and what that system does (Robinson 2008), the definition given above for model describes that the abstraction is with a purpose. That purpose is perspective that the modeler has of the system, and what questions are going to be asked of it. It is likely that each modeling perspective of a system will vary from others. In considering the motivation for modeling that (West 1996) gives, that perspective should be as wide as possible, but it is always an abstraction of the original system (in some way), and so it is likely that even wide-perspective models will vary from each other. Because of this variation in perspective, each model is likely to be different. Although, all models of a system are held to be accurate, if they can serve to answer the questions that determine their perspective.

One way to view this question of perspective, and how it shows that models lead to simulations, is to consider a contributing work from the field of semiotics. In (Ogden and Richards 1923) the idea of the semiotic triangle is presented. It shows how a semiotic representation of a referent is formed, by the person expressing the semiotic (symbol) first forms a conceptualization of the referent, and then this conceptualization is encapsulated in a symbol that is to stand in for the referent in some exchange of information. In the same way, a model is the embodiment of the system it is representing, and then the model gets encapsulated in a simulation that is to stand in for the system to present answers to questions.

As with semiotics, as many conceptualizations can serve a single referent, so many different models can serve to describe a single system. As mentioned above, if each of these models (from a different modeling perspective) is successful at representing the system such that questions inspiring the perspective are answerable, then it is a successful model. However, from any one of those models, it is possible that a number of different simulations could be developed. Consider, for example, in the case of computer simulations – from a single model a number of different simulations could be written using different computer programming languages (each of which would introduce differences in how the model is interpreted into a machine understandable program).

What this all exhibits is that while a model stands for a system, and a simulation interprets a model, for a single system there can be any number of models, and from a model, any number of simulations. To further add to the variety of different possibilities, the different models of a system could be done using the same modeling technique, but resulting in different models, or they could be done using different modeling techniques. Given such a case, it can be seen why an engineer, or some other interested party, would have a way to compare what it is that a model is trying to represent between different types of models.

2.3 Models as Representation of a Conceptualization

For a model to be useful in answering questions about the system that it represents, it must necessarily represent enough conceptual information about that system, to provide the basis for generating answers. Within computer science, for example, this has been seen since the early efforts at capturing information systems as diagrammatic models. The earlier examples of this were expressed as finite state machines. Two types of machines are commonly regarded; the first being the Moore machine (Moore 1956) which represents the states and transitions between them of a state machine, and indicates that system output is produced when each new state is attained. The second is the Mealy machine (Mealy 1955) which represents, likewise, states and transitions, but indicates system output with each transition. In both regards, the states of the machine are the main focus, with the transitions there to show the necessary, but incidental, arrangement of those states.

In addition to finite state machines, other early attempts to depict the operations of systems conceptually included flow charts, which conceptualize the algorithmic operations of an information system, without as much focus on interspacing states. The strength of the flow chart over the state machine is the possible depiction of branching flows of activity, parallel flows of activity, and so on – all concentrated on the sequencing of algorithmic steps. This type of conceptualization concentrates on the processes of the system, and their sequences, with only minimal representation of the objects and states.

In (Sowa 2000) it was shown that although both of these views represent valid conceptualizations of a system, it is necessary to consider both views (the state, or object view; and also the activity, or process view) in order to have a valid and accurate depiction of a system that can answer systems, and represent dynamic activity. For an example of how this can be accomplished, Sowa presents the Petri net (Petri 1966) as a possible method for showing such a system might be considered. That author (Sowa 2000) then goes on to show how not only are details and parameters of objects necessary to a conceptualization, but also the details and parameters of processes.

3 FUNCTIONAL COMPONENTS OF MODELING TECHNIQUES

That there are different techniques from different domains – all useful – but all with different perspectives is accepted as a given condition of the current state of the art in conceptual modeling. But the commonality of types of components that can be relied on to describe all of those techniques is shown, and this is then followed up on to suggest a (semi-)formal method that can be relied on to both describe and compare those techniques, and also to describe and allow combination of specific models.

3.1 Modeling Techniques each Support a separate Perspective

There are any numbers of different modeling techniques, from different domains, each existing in order to support a different type of modeling. Each of these different needs for modeling (from different domains, and serving different representational needs within those domains) is a different perspective. To support those needs, different techniques necessarily are oriented (in terms of what, about their referent system, they allow to represented with detailed knowledge) to different uses. However, in spite of these different domains, and different perspectives, it is the premise of the work presented here, which follows (Turnitsa 2012), that there are common components, assembled in an open, but orderly, fashion that makes a common formal method to describe all such techniques possible.

3.2 Common Components

The premise of the method presented in this paper is that all conceptual modeling techniques can be described with a minimal set of different types of components that are commonly found regardless of the domain that the modeling technique comes from. This can be seen from the literature for several domains. From modeling and simulation, it can be seen in conceptual models. From systems engineering, it can be seen in techniques such as SysML and Object Process Methodology (OPM). From software engineering, it can be seen from UML. And from knowledge engineering, it can be seen from ontological modeling. These are all presented in the following section, from the attendant literature.

From a perspective intended to show the functional decomposition of a system and represent it in a model, as shown in (Dori 2002), it is stated that systems can be modeled by objects and processes. Objects are the elements of the system that will remain the same unless operated on by some other part of the system, by his definitions. Processes are the elements of the system that perform those operations. Dori indicates that his method expands on most system modeling techniques, which rely on the capture of objects and the traditional system science technique of representing processes as state changes between states of objects, to show that processes are more complex and deserve additional consideration. This has additionally been shown to be the case within (Tolk et al. 2009), where alignment and the ability to compose models is shown to require consideration of objects, processes, and constraints. From (Shlezinger et al. 2006) it is seen that both elements are required – objects grant the model understanding of the structure of the system, and processes grant the model understanding of the dynamic behavior of the system. From (Turnitsa 2012) it is shown that both are equally necessary in order to exhibit the meaning of what a dynamic system is (and can do). This is summarized there, by the following: *The two equally important class types in OPM, objects and processes, differ in the values of their perseverance attribute: the perseverance value of object classes is static, while the perseverance value of process classes is dynamic. OPM's combination of objects and processes in the same single diagram type is intended to clarify the two most important aspects that any system features: structure and behavior (Shlezinger et al. 2006)*

Bringing this into alignment with (Robinson 2008), the behavior of the system is described by the model as its contents, and it relates how the model transforms input into output. If the model is considered from the resolution of considering that behavior as a single transformation (input into output) the model is a function, accepting input to the model as its domain, and producing output from the model as its range.

Internal to the system, unless it is trivial in composition, there will be any number of different sub-stages of this transformation, where parts of the model are being transformed to be used by other parts of the model, as per (Dori 2002). The Unified Modeling Language, or UML (OMG 2002) assembles a number of different modeling techniques to describe a system. These techniques are split into two categories, agreeing with (Shlezinger et al. 2006), in that they are described as the structural diagrams (giving the structure of the system) and behavioral diagrams (giving information about the functioning of the internal processes that make up the overall system transformation). Considering the Systems Modeling Language, or SysML, it is seen (OMG 2010) that these two categories are also present, but SysML also includes a new modeling technique, or diagram as they are called in SysML and UML, called the "requirements diagram", which is additional to, and separate from, the UML categories of behavioral and structural diagrams. It may be of interest that an internal study furthermore ensured that the concepts utilized within the Business Process Model and Notation (BPMN) as documented in (OMG 2011), which is commonly used in industry, are a subset of the concepts evaluated in the research underlying this paper. More in-depth evaluations have been presented by Shuman (2011).

In understanding what software engineering (UML) and systems engineering (SysML, OPM) declare that a model should consist of, we have seen so far the specific mention of objects and processes, a view from the literature that is corroborated by (Tolk et al. 2009). The model will serve as some knowledge representation of the system it represents; this is also in alignment with the idea of using modeling and simulation to represent the semiotic triangle (Ogden and Richards 1923). There are other modeling communities, however, that are also concerned with knowledge representation. One of those worth considering is the knowledge engineering community. This community, which is concerned with knowledge representation and knowledge modeling, is defined (in terms of modeling) as, "Knowledge engineering is the application of logic and ontology to the task of building computable models of some domain for some purpose" (Sowa 2000).

As the foundational method for describing knowledge, the philosophical pursuit of ontological representation will be considered to see if knowledge representation models contain the same contents as the other modeling domains considered (software engineering, system engineering and modeling and simulation). From the ontology literature it is clear that the presence of objects and processes, and their relationship, is such a clear part of the fabric of understanding, that it engenders two different viewpoints on how a system is understood. The first is the object-centric view, referred to in (Grenon and Smith 2004) as the 3-D, or three dimensional, viewpoint. It is so named because it assumes that objects are (in a Newtonian sense) in existence and inviolate in their definition, unless acted upon by some outside force – a process. A modern description of this metaphysical commitment is presented in (Lowe 1998). The second view on reality is the process-centric view, referred to in Grenon and Smith as a fourth dimensional view. This second metaphysical view is much more in alignment with some of the ideas of modern physics, in that what we perceive of as objects are just signifiers of ongoing processes – everything is always in some state of flux. This second view is presented in (Sider 2001). A source that unifies these two viewpoints is (Grenon and Smith 2004), and a conclusion made in that source is that regardless of the perspective (either object-centric or process-centric) the reality is defined by objects and processes and their relations with each other.

Not mentioned explicitly so far, but always there in all of the literature sources mentioned so far, is the implication that these content defining elements – the objects and processes – can be associated with each other. This is present, certainly, in data modeling with its view (Chen 1976) of the relational model. It is also there, in most graphical modeling techniques as the way of connecting the elements of the model – some boxes are related to others. Considering the diagrams from either UML or SysML this can readily be seen, as well as the conceptual models described by (Robinson 2008) where there is definite and committed association between objects such as input and output and internal model components, and the transformational processes that affect them. Without the ability to indicate association of the object and process elements by these relations, the conceptualization of the system would not be possible.

Taken together, then, the elements that Robinson (2008) claims make up a conceptual model are what (Dori 2002) claims are part of a systems model, and what (Grenon and Smith 2004) claim are part of an ontological model – objects, processes and the relations between them.

3.3 Objects, Process and Relations

From these common elements to all modeling can be developed the components are that make up a model. From the definition for modeling presented earlier from (Turnitsa 2012), it has been seen that "a model is a purposeful process of abstracting and theorizing about a system, and capturing the resulting concepts and relations in a conceptual model". As such, the model is a sign standing in place of the system it represents. Viewed this way, models are tokens for knowledge – about the system they are representing. Each component that is identified will either have to be useful for the model to either represent some aspect of the system being represent, or necessary in order to internally support the whole representation of that system. Each component, then, can be considered in isolation from all the rest of the model, yet it represents knowledge about some part of the system – as the model itself is representing knowledge about the whole system. In this relation (part to whole) although each component can be considered individually, they are all intended to contribute (and be part of) the whole model. This leads us to the following definition for a component of a model: *A model component is an identifiable part of the model that represents some part of the knowledge that makes up the whole model, but which can be considered individually as well.*

Mentioned earlier, the model must be representative of the system, and encapsulate enough of its complexity in order to be expressive (to answer whatever question is being asked of the model), yet simple enough to warrant the extra effort of using the model to represent the system, rather than work with the system itself. What is said about the components of a knowledge representation, and then more generally, any model, can be seen from existing literature, and was discussed at length in (Turnitsa 2012). The literature of conceptual models identifies many elements that are part of models. In (Robinson 2008) these are enumerated as objectives, inputs, outputs, content, assumptions, and simplifications.

Examining that list, there are two categories of informational elements about models. First are the elements that make up the model – inputs, outputs and content. Second are the elements that describe the character of the model – objectives, assumptions and simplifications. For the purpose of exploring the nature of the components of a model, the first set is of interest for consideration here. The effects of the elements that describe the character of the model are derived may influence the elements that make up the model, but those effects will be present in the definitions and parameters of the elements that make up the model.

As discussed earlier, the common components for the modeling techniques of a wide variety of different domains are all objects, processes or relations. The literature on knowledge representation, and the ontological representation of knowledge, is relied on to show that a system can be described with these three components.

From the literature of information theory, we can see in (Shannon 1948) that although the transferring of information includes a speaker, a listener, and a medium of transfer, it is implied that information will exist in "symbols". This is also held up by the earlier (Ogden and Richards 1923), which holds that a representation of a conceptualization is a symbol. This idea is also held up in the literature on ontology (Smith and Grenon 2004), which identifies that the symbols for describing a world consist of symbols for enduring entities and symbols for occurring entities. These are described as objects and processes.

Taking these two views (whether a system should be viewed as a number of objects, and they change because of processes; or where everything is in a constant state of change) together, under the frame of considering a model as a representation of knowledge, we see that a model (Robinson 2008) should accept input, produce output, and have some content that describes how the output is produced from the input. As the entities of a world are (at least) objects or processes (Smith and Grenon 2004) then the input, output and content should address these (as will be demonstrated later, a third possibility exists, and that is the association of those objects and processes – which will be referred to as relations). As the model includes as separate things the content, input and output, then it follows that all three of these are distinguishable from each other. The introduction of input into the content is identifiable as being separate from the content, as is the output. If the content contains a description of how the input is changed to become the output, then regardless of what the input consists of, something must occur that changes the identity of that input. That change is represented by one or more processes within the model content. It is in this way, at a very coarse level of consideration, that the modeled operation (the content) is a function. It accepts some input, and produces some output.

4 REPRESENTATION OF MODELING TECHNIQUES USING OPR

Identifying that the three components of a modeling technique in the preceding session, how these can be used to identify what each of the elements of a modeling technique are (defined in terms of those three components), and how they work together is what the OPR approach is designed to do. This section will do that by showing how the parts of a technique can be described using the three components; showing the formal rules for how those three components can work together; and some interesting implications, or corollaries, that come out of the interaction of the defined components.

4.1 Model reduced to Components

There are parts of a system that may be identified within a model that maintain identity over time, and these are objects (Corcho and Gomez-Perez 2000). The term object here is not to be confused with the computer science term object. In having an understanding of a system and its persistent elements, we can assume that they will remain stable, unless operated on by something that is within the system (a modal change) or something from without the system (extra-modal change). It is reasonable to expect this persistence (Chandrasekaran et al. 1999). The persistence of these elements is what allows for the meaningful capture of them into artifacts (Niles and Pease 2001). A method to distinguish objects from similar objects is the identification of the various qualitative and quantitative attributions that identifies the object (Guarino and Welty 2000).

The model describes that meaning of what the system it describes is, and what it does. The model uses symbols to express that meaning, and those symbols are in at least two different categories – objects and processes (Whitehead 1978). It is clear that ontologically objects are things that *are*, that is – they exist. Processes, ontologically, are things that *occur* – either at a single point, or over some time. While it may be enough for now to state that objects are the portions of a model that persist in their existence – the parts, components, etc., what the meaning a process conveys is not yet clear. A definition of process, and what a Process component can do, is required.

If there is change – that is, objects changing because of something represented within the model – then that change is a dynamic element of the model. In order to derive a definition for what that dynamic element is when it is considered as part of a model, it is required to look to the literature for definition. To simplify things, although the word has not been defined yet in the dissertation, the word process will be used for that dynamic element. In order to see what this is as the literature is guiding us to a formal description of what the model consists of, defining the word process is necessary. So far, whatever it is that changes within a model – and it has just been shown that a dynamic model is one that is responsible for some change of the input to the model into the output from the model – has had the method of enacting that change called either the dynamic aspect, or dynamic element of the model – or alternatively it has been referred to "what it is about the model that changes". Within (Turnitsa 2012) additional considerations and definitions from the various literature on what it means to have dynamic capacity within a model, and therefore what it means to have 'processes' within a model.

From these definitions, then, the thing that remains through them all is the defining identity that "a process is responsible for some transformation or change". The characteristics of the process will be described further on in this chapter. When considering the system as a whole, then, each process is a marker of the system being in one state, and then the transformation of the process occurs, and then the system is in another state. If the process exists over time, then the system is in a dynamic state during that time – neither the pre-transformation state, nor the post-transformation state. Here, when the state of the system is referred to, the same sense is implied as for the definition for "system state" from earlier.

With regard to a model, then it is seen that an object is something that retains a state of identity within that model. A process is something that is a marker between two different states of the model. In the case of the object, the word state is the identity of the object. In the second case of the process, the word state is the identity of the model. Since we have seen that the model consists of objects and processes, then a process is the change to one or more components that make up the content of the model. The component(s) that the process affects must be identified by the model, to express information about the system. As the content is some collection of objects and processes, and processes occur that mark a change to the state of one or more of these, then there must be a way of associating a process with what it is changing.

That association indicates a third entity that is part of the model's content. It is referred to in the computer science literature (Codd 1970, 1974) as a relation. This view was refined, and given further consideration (as the basis for relational database management systems) in Chen (1976). The reliance on a relation is identified as a key to representing information in all of the fourteen (or more, considering SysML) diagramming techniques presented in both UML and SysML (OMG 2002, 2010). Finally, it is represented within both the knowledge representation literature (Sowa 2000) as well as the systems modeling literature (Dori 2002) as the connection between objects and processes. This new component (the relation) is defined as "the means of associating together other parts of the model". As with processes, and objects, additional considerations are presented in (Turnitsa 2012) concerning relations, however it is also pointed out there that room is quite welcome for an in depth study of association between concepts within a model, and what this means for modeling and system representation. The three identified definitions of object, process and relation components are these: object component, process component, and relation component. Based on the definitions given earlier for a component and for a model, each of these components is expected to be an identifiable portion of the model that fulfills some role in expressing the information about the system that the model is describing.

An object component is a model component that has continued existence. The object component is a part of the model that is representing some part of the system that will retain its identity (have continued existence), until that identity is altered by some other component.

A process component is a model component that is responsible for describing change or transformation. As has been shown in the preceding section defining process, there are dynamic aspects to some systems, and therefore models of those systems must represent that change. A process component is the component that captures that information in a model.

A relation component is a model component that is responsible for associating other components. The most immediately apparent reason for associating together components comes from the definitions alone – object components do not change identity until operated on by some other component. Process components are responsible for some change. In order to represent within the model which process is affecting an object, association of the two is required, and the component that describes it is the relation. However, other associations are also possible, such as what has been shown already in defining what a relation is.

4.2 Defining Qualities of the Components

The components listed are fine for capturing the main elements making up the structure of the model, each conveying some information about the system being described. However, it is likely (necessary in some cases) for each of those components to have some sort of parameterization, giving it qualitative and quantitative distinction from other similar components, and also granting further definition to the description of the referent system. In the OPR system being presented here, these parameters are referred to as defining qualities. Although the defining qualities are described here, with emphasis on their structural role in the OPR system, their functional role deserves more exposition.

Objects are, as described herein, loci of identity. This matches with the earlier definition given of Objects as a component of the model that has continued existence. They are the things that can be observed independently within the Model, with identity separate from all other components. Giving them definition, however, are their Attributes. These give the Objects their qualitative and quantitative distinction from other objects. *Attributes are defining qualities for Objects, and grant them qualitative and quantitative distinction from other objects.* This is directly corroborated by the literature on ontological views of domains (Chandrasekaran et al. 1999), and the language of that community in defining what an object is (Guarino and Welty 2000).

Processes are, as described herein, the descriptions of changes within the model. What is being changed by the process is not limited, nor are there other means within the model to address change. The defining qualities of processes in the formal method described here are referred to as characteristics. These are necessary to identify the behavioral nature of the process, and to distinguish one process from another (Sowa 2000). These define for the process what it changes, and the existential definition of when the change occurs, and how it takes place. That these distinctions between processes exist, and can be captured is shown within (Whitehead 1978). *Characteristics are defining qualities for processes, and they describe the behavior of the process as well as providing qualitative and quantitative distinction from other processes.*

Relations are the means for associating together components. Although association between components may be in effect, there are likely to be conditions to that association, and that is what the defining qualities of relations define. This is addressed in the literature on the philosophy of information representation (Smith and Grenon 2004) and also in the literature of computer science, in how entities are related to each other (Codd 1970, 1974), and refined in (Chen 1976). These qualities are specifically granted identity in the formal method presented in the dissertation underlying this paper and will be called Rules. The literature is clear that processes are affecting other components (Whitehead 1978); to indicate this, the relation between the two is necessarily identifiable (Sowa 2000). *Rules are the defining qualities for relations; they serve to identify the nature of association the relation is making, and to provide qualitative and quantitative identity to the relation.*

It is interesting that while all of the components share the stated feature of objects that they are expected to retain identity (absent some change brought about by a process component), that the other two components (processes and relations) are reliant for their distinguishing definition on the presence of other components. Frege's study on the relations (Frege 1892) between a changing quality and the identity it affects illustrates the need for this association, and the requirement for a relation component. This is made clear in modern literature discussing the representation of a process based system (Haller et al. 2006). A relation that does not have defining characteristics identifying its nature within a temporal framework is meaningless in a system that exhibits change, as a dynamic system must.

4.3 Corollary Observations

Consideration of this brief semi-formal method will show that the three types of components can be combined in order to show a broad variety of systems and their activities. It covers the input, output and content discussed earlier, but also makes provisions for describing space, time, multiple references for either, entities and their behavior/activities within the chosen framework of space and time. It does not address the elements of the model captured in (Robinson 2008) defined earlier as describing the character of the model – its assumptions, motivation, and operational constraints. These have been addressed in (King and Turnitsa 2008), and proven in (King 2009) and have been shown to be outside of the components of a model.

Corollary 1 *The simplest model is one consisting of a single object.*

A minimal model – one that is only a depiction of a non-changing object – is one that has contents consisting of exactly one component – a singleton object.

This model does not exhibit any change, either in definition or in time. It is simply a description of an object, without reference to or consideration of time and space (each of which require other components to be in the model). From the definitional statements earlier, the object would require at least one defining quality, which is the attribute known as the identity quality.

Corollary 2 *The simplest dynamic Model is one consisting of three components – an object, a Process, and a Relation.*

Time in a model is represented by a set of three related components – an object, a process, and a relation associating the two of them. Together the three of these components can make up a simple dynamic model called "time". For convenience, we can refer to the object as the "time object", and the process as the "time process". The relationship can be called the "temporal relation". Together these three components represent a common example of the simplest dynamic model that can exist – one with one of each of the components.

When the three components of a simple dynamic model are those that are suggested above (the "time object", the "time process", and the "temporal relation") then it can be called time model.

Corollary 3 *A sub-model does not have any definitional relations for any of its components to other components outside of the sub-model.*

If a model has some conception of time as part of it, then it incorporates a time model as a sub-model. A sub-model is portion of a model that can be conceived of in isolation from the rest of the model. The sub-model cannot have any of its defining qualities have definitional relations to the main model. An example of a sub model from the literature would be an atomic DEVS component that is part of a coupled DEVS specification (Wainer 2009). A definitional relation is one where the value of the defining quality derives from some other defining quality, through a relation.

Other components outside the sub-model may be reliant on the sub-model for definition (for example – a model that has processes that occur at a certain time, may have a relationship between those processes and the time Sub-model), but not the other way.

Corollary 4 *The existence of types and instances of components are so indicated by a type-instance relation between the type component, and any instance components.*

Although each component is a non-empty set of one or more defining qualities, and each defining quality may have an associated value, the value does not have to be a singleton value. It can be a range, or a set or some other numeric construct. This allows for components that are types for other components to exist. A component that is an instance of another component (which itself is a type component) has all defining qualities that the type component has. It may have additional defining qualities, or it may have separate values for its defining qualities, but in some way it differs from the type component. Correspondingly, sibling components, that is, components that are all instances of the same type component, will all have some difference from each other, whether it is additional defining qualities, or simply different values for the same set of defining qualities, or some combination of these two. The relation component can be used to associate components that share a type-instance relation. This would be a member-of relation.

There are additional corollaries presented in (Turnitsa 2012), and those as well as these presented here, are described in more formal manner there, with specific representation of the various axioms that define the components and their defining qualities. These are presented here, as an indication of the sorts of implications that arise from the simple axioms that are seen to be part of conceptual modeling techniques. Furthermore, a strict formalization of the OPR concepts is part of Turnitsa 2012) as well, although it needs to be pointed out that the symbolic manipulation for inferring theories has not been part of the studies and remains an important topic of future evaluations.

5 APPLICABILITY OF OPR

The fact that there are identifiable components common to modeling techniques from different domains, and that some formal rules concerning the composition and interplay of those components can be identified is an interesting research topic. Further, that those components and the axioms governing their coexistence can be put to use to describe conceptual modeling techniques is yet more interesting. All of this has been presented, and is distilled in the preceding sections, from (Turnitsa 2012). However, in order to see if this method of description and definition is useful, several cases are described here where it may be put to good use by the modeling and simulation professional.

5.1 Comparison of Models - Dynamic Characteristics

The separate treatment of model processes by the OPR approach, allows for the specific recognition of the methods within modeling techniques that are there to represent the dynamic behavior within a system. That this is necessary, and possible, is shown within (Turnitsa 2012). There, by relying on the basic definition of processes and their characteristics (the defining qualities of the process component), as well as the specific characteristics discovered by both survey and analysis of the literature, it is shown that in cases where there is a sufficiently dynamic component to a system being modeled, it is likely that by only treating the model as a series of finite states of object representations, without describing the dynamic components that introduce and define the changes between states, that it is quite possible to actually lack (or mis-represent) information concerning they system within such a model. Therefore, in order to avoid that lack of information (or mis-representation) it is necessary to have a description of the dynamic element. When possible, this should be done with the native elements of the modeling technique being employed, however in other cases, or if the technique proves insufficient, then adaptation of new technique or extension of existing technique can be accomplished by referring to the OPR definition of processes and their characteristics.

5.2 Comparison of Modeling Techniques – Capabilities

By relying on, and using, a common meta-language to describe modeling techniques (and even specific models) it is possible to rely on such a language to compare specific capabilities of techniques one to another, and specific composition of models, one to another. Because the OPR method relies on a meta language, and formal descriptions of its own symbols (components and defining qualities), as well as operations for combining those symbols into statements (relations) and to alter the defining composition of one or more components into something new, all the basic requirements for a model to be treated as the subject of a formal method are in place.

5.3 Comparisons of Models from Different Techniques – Multi-Model Representation

In addition to having a neutral way to describe the components of a modeling technique, or the function of elements in a model, it is also useful to have a meta language (the OPR method) to describe the interaction between different models (or sub models) in a multi-model solution. In such a case, it is possible that the different models (or sub models) would be of different modeling technique types, and so to understand the interaction between them, a meta language is required that can take into account the composition and behavior of each of the existing modeling technique types.

6 CONCLUSION

The application of process philosophical ideas to conceptual modeling is not only possible, it can be beneficial as well, as a process oriented viewpoint allows to express different things than a state-oriented viewpoint. The formal method that has been derived by Turnitsa (2012) suggest what is possible in representation within a modeling technique, so that once the method is used to describe a technique, it should be able to suggest in specific ways how that technique could be modified, or extended, in order to make it capable of representing more about a system.

The development of the OPR formal method into its own modeling technique is a possibility as well. Without developing OPR into a formal method, however, future work will most likely be accomplished by using the method to evaluate, and express in the neutral terms it can deliver, explorations of any number of modeling techniques, and presenting these in the literature. In addition, with the application of specificity to the defining qualities of the components of the formal method, it should also be possible to use it to evaluate not only conceptual modeling techniques, but also specific conceptual models themselves. The implications and potential uses of such an application await future exploration.

The objective of this work was not to attack or criticize existing methods but to show how new findings can be used to enrich and complete them to allow for alternative approaches that may be of advantage in several domain that struggle with traditional approaches. As exemplified by Styhre (2002) for strategic management challenges, some problems that are hard to express otherwise can be communicated using a process-driven view. Process philosophical methodologies should therefore be included into the research agenda of M&S methods and be considered as a valuable viewpoint for conceptual modeling.

REFERENCES

Aristotle. 350 BC. *Physics*. Retrieved from http://classics.mit.edu/Aristotle/physics.html

Balci, O. 1998. Verification, Validation and Testing. In *Handbook of Simulation*. Edited by J. Banks, John Wiley & Sons, 335-393.

Chandrasekaran, B., J. Josephson, and V. Benjamins. 1999. What are ontologies, and why do we need them? *Intelligent Systems and their Applications* 14(1):20-26.

Chen, P. 1976. The Entity-Relationship Model – Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1(1):9-36.

Codd, E. 1970. A relational model of data for large shared data banks. *Communications of the ACM* 13(6):377-387.

Codd, E. 1974. Recent investigations in relational data base systems. *Proceedings IFIP Congress*, Amsterdam, Netherlands, 1017-1021.

Corcho, O., and A. Gomez-Perez. 2000. A roadmap to ontology specification languages. *Knowledge Engineering and Knowledge Management. Methods, Models and Tools*. Edited by R. Dieng and O. Corby. Berlin, Germany: Springer, 80-96.

Dori, D. 2002. *Object-Process Methodology*. Berlin, Germany: Springer-Verlag.

Frege, G. 1892. Über Begriff und Gegenstand. *Vierteljahresschrift für wissenschaftliche Philosophie*, 16:192-205.

Grenon, P., and B. Smith. 2004. SNAP and SPAN: Towards Dynamic Spatial Ontology. *Spatial Cognition & Computation: An Interdisciplinary Journal*, 4(1):69-104.

Guarino, N., and C. Welty. 2000. A formal ontology of properties. *EKAW-2000: The 12th International Conference on Knowledge Engineering and Knowledge Management.* Edited by R. Dieng and O. Corby, Berlin, Germany: Springer, 97-112.

King, R. 2009. *On the role of assertions for conceptual modeling as enablers of composable simulation solutions.* Ph.D. Dissertation, Batten College of Engineering, Old Dominion University, Norfolk, VA.

Law, A., and W. Kelton. 2000. *Simulation modeling and analysis (3rd ed.)*. New York, NY: McGraw Hill

Lowe, E. J. 1998. *The Possibility of Metaphysics: Substance, Identity, and Time.* Oxford University Press.

Mealy, G. 1955. A method for synthesizing sequential circuits. *Bell Systems Technical Journal* 34:1045–1079.

Moore, E. 1956. Gedankenexperiments on sequential machines. *Automata Studies, Annals of Mathematical Studies* 34:129–153.

National Institute of Standards and Technology (NIST). 1993. *IDEF0: Integration Definition for Function Modeling. Federal Information Processing Standard (FIPS).* Publication 183. Washington, DC.

Niles, I., and A. Pease. 2001. Towards a Standard Upper Ontology. *2001 Proceedings of the International Conference on Formal Ontology in Information Systems*, Ogunquit, ME, 2-9.

Ogden, C., and I. Richards. 1923. *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*. London, UK: Routledge & Kegan Paul.

Object Management Group (OMG). 2002. *Unified Modeling Language Specification, Version 2.2.* Retrieved from http://www.omg.org/ Last visited July 2012.

Object Management Group (OMG). 2010. *System Modeling Language Specification, Version 1.2.* Retrieved from http://www.omg.org/spec/SysML/1.2/ Last visited July 2012.

Object Management Group (OMG). 2011. *Business Process Model and Notation, Version 2.0.* Retrieved from http://www.omg.org/spec/BPMN/2.0/ Last visited July 2012.

Petri, C. 1966. *Communication with automata.* DTIC Research Report AD0630125, Princeton, NJ.

Rescher, N. 1996. *Process Metaphysics: An Introduction to Process Philosophy*. State University of New York Press

Robinson S. 2008. Conceptual modeling for simulation part I: Definition and requirements. *Journal of the Operations Research Society*. 59(3):278-290

Sargent, R. 2005. Verification and validation of simulation models. *Proceedings of the 37th Winter Simulation Conference*, Orlando, FL, USA, December 4-7. 130-143

Shlezinger, G., I. Reinhartz-Berger, and D. Dori. 2006. Analyzing Object-Oriented Design Patterns from an Object-Process Viewpoint. *Proceedings of the 2006 Next Generation Information Technologies and Systems*. Edited by O. Etzion, T. Kuflik, and A. Motro. LNCS 4032, Springer. 186-197.

Shuman, E. A. 2011. *Understanding the Elements of Executable Architectures through a Multi-Dimensional Analysis Framework*. Ph.D. Dissertation, Batten College of Engineering, Old Dominion University, Norfolk, VA.

Sider, T. 2001. *Four-Dimensionalism: An Ontology of Persistence and Time*. Clarendon Press.

Smith, B., and P. Grenon. 2004. The Cornucopia of Ontological Relations. *Dialectica* 58(3):279-296.

Sowa, J. 2000. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Pacific Grove, CA: Brooks Cole Publishing.

Styhre, A. 2002. How Process Philosophy Can Contribute to Strategic Management. *Systems Research and Behavioral Science* 19:577-587

Tolk, A., S. Y. Diallo, R. D. King, and C. D. Turnitsa. 2009. A Layered Approach to Composition and Interoperation in Complex Systems. *Complex Systems in Knowledge based Environments: Theory, Models and Applications*. Edited by A. Tolk and L. C. Jain. Studies in Computational Intelligence, SCI 168, Springer, 41-74.

Tolk, A., S. Y. Diallo, C. D. Turnitsa, And J. J. Padilla. 2011. How M&S Interoperability is different from other Interoperability Domains. *Spring Simulation Interoperability Workshop*, Boston, MA, April 2011, 12-20

Turnitsa, C. 2012. *Exploring the Components of Dynamic Modeling Techniques.* Ph.D. Dissertation, Batten College of Engineering, Old Dominion University, Norfolk, VA.

Wainer, G. A. 2009. *Discrete-Event Modeling and Simulation: A Practitioner's Approach.* CRC Taylor & Francis, Boca Raton, FL.

West, M. 1996. *Developing high quality data models.* Newcastle, UK: University of Newcastle

Whitehead, A. 1978. *Process and reality: An essay in cosmology*. Reprint of the 1929 original. New York, NY: Free Press.

AUTHOR BIOGRAPHIES

ANDREAS TOLK is Professor of Engineering Management and Systems Engineering with a joint appointment to Modeling, Simulation, and Visualization Engineering at Old Dominion University. He is also affiliated with the Virginia Modeling Analysis and Simulation Center. He holds a M.S. and Ph.D. in Computer Science from the University of the Federal Armed Forces in Munich, Germany. His email address is atolk@odu.edu.

CHARLES TURNITSA is currently a member of the faculty, Assistant Professor, in the TSYS School of Computer Science, at Columbus State University. He received his Doctor of Philosophy from Old Dominion University, in Modeling and Simulation. From 2004 through 2011, he served as a member of the research staff at the Virginia Modeling Analysis and Simulation Center. His email address is cturnitsa@gmail.com.