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

REPRESENTING THE CHARACTERISTICS OF MODELED PROCESSES

Charles Turnitsa

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

ABSTRACT

In modeling a system which exhibits some dynamic behavior, representation of the processes that originate the dynamic changes in the system is elementary to understanding how the system works, and also to having an accurate and meaningful model of that system. In order to model such processes, the defining characteristics of the processes prove useful in their composition and presentation. A minimal subset of those characteristics are presented here, with consideration for potential variations among them, and also consideration of possible implications that such modeling may lead to - understanding of system behavior that can be represented with such modeling, that may not be possible without these characteristics being exhibited.

1 INTRODUCTION

Dynamic systems are systems in which something changes. It can be whatever the system is intended to perform "work" on, or it can be elements of the system itself, or perhaps just the passage of time – but if a system is dynamic, then something changes. The modeling of systems, including dynamic systems, has been shown (Tolk and Turnitsa 2012) to be most effective when it is based on an understanding of what the system is composed of, and a representation of that composition. With a dynamic system, this has been shown to include a representation of what is changing, and also the means for bringing about that change. The established view for doing such a model, especially as seen in computer science, is to represent that change as some point in time where one or more objects change state. If considered from the point of view of the object, this implies that there is a point in time where the object in one configuration of existence, and then the next instance (after the state change occurs) it is in another configuration of existence. Something about the object has changed. This is a common, and extremely useful, way of considering systems, as this view leads to models that will be implemented as simulators to be run on digital computer systems. However, as we can use discrete event simulators to implement simulations that approximate continuous systems, and because there are such systems in the world – where the process may not be a simple point change of state, but may take place over time, it becomes useful to consider how to model such processes themselves. And once processes are viewed as something that may take place over time, then it follows that they themselves may change over time (different rates of change, different effects, and other changes to their behavior – over the time that they run). In order to treat non-trivial systems with increased chances of successful representation, it appears then that the modeling of processes equally important as the modeling of the objects and states of systems.

This view – resulting from considering the existential composition of models and the act of modeling – led to the research question of a recent doctoral dissertation at Old Dominion University (Turnitsa 2012). That dissertation resulted in a method for describing all of the identifiable components of a model or a modeling technique, and how they interrelate to each other. One type of component that was considered in that work, was the component of a model responsible for change to other components of the model – the process component. In performing and presenting the research of that work, a collection of observations and inferences arose concerning the characteristics of modeled processes. They are presented here, in order to contribute to the broader study of the theory of modeling, as considerations on the topic. This work follows on the work from WSC 2012 (Tolk and Turnitsa 2012) that established the need for such a system of representation by component description, and this represents the more in depth presentation of the work on the process component from the Object-Process-Relation (OPR) method. The method itself is not presented here in detail, but a brief overview of the three components (objects, processes, and relations) are described below (2.2).

2 CHARACTERISTICS OF PROCESSES

As presented earlier (Tolk and Turnitsa 2012) (Turnitsa 2012) it is beneficial for a model to have not only components representing the various functional elements of the system it is representing, but also to have defining qualities to explain the differences among, and parameterization of those components. As shown in the earlier works, this is in alignment with (Sowa 2000), showing that the activities in a system model are worth identity and representation on an equal basis with the objects and states in a model. The "activities" are referred to here as processes, and are given definition by having their parameters made explicit. In the case of the process component in a model, these parameters have previously been referred to by the name of *characteristics*. This paper will continue in that mode.

2.1 Four Characteristics of Processes

In considering what the characteristics of a process might be, other than some sort of identity for the process itself (so that it might be referenced within the model – either externally by the modeler or some other user, or internally by other components of the model), several characteristics have arisen just from considering the functional possibility of a locus of change existing within a system. That change, in order to be successfully represented within a simulator, must have some conditions under which it occurs (or begins to occur), so the first characteristic is necessarily the definition of when the process begins. Consideration of a process as either a single point in time where a state change occurs, or of some continuous process where the change takes place over time, suggests that a characteristic is the temporal behavior of the process. If the process represents change, then the question of 'what is changing' should be asked of the process. This leads to the requirement for a characteristic describing the effect of the process. Finally, if a process is to begin, and it may take place over some time, then it follows that the question 'when does the process halt' should be asked.

Together with the identity characteristic, these four characteristics – initialization, behavior, effect, and halting – are not the only characteristics that could be asked of a process, however they do seem to be the functional minimum. And even then, as will be seen in the following sections, there are times when certain types of processes make the defining of some of these characteristics to be redundant with each other. Then in the spirit of contributing to the conversation concerning the theory of modeling, but not presuming to end that conversation, these are presented as suggested characteristics, but not suggested as the only such list that could exist.

2.2 Other Components Assumed

From the method that this examination of processes is taken from (Turnitsa 2012), there are two other types of components assumed that together with processes can make up the means for describing all of the parts of a model, or a modeling technique. Those other two components are the **object** and the **relation**. Both of these, as part of OPR method, were introduced to WSC in 2012 (Tolk and Turnitsa 2012).

The OPR method is a framework for representing all of the elements of a model as one of three different types of components – objects, processes and relations. Each of these components is further defined by a number of defining qualities. To distinguish among them, objects have attributes, processes have characteristics, and relations have rules. Each individual component additionally has a defining quality of "identity" which names and individuates that component from all others.

Objects, as used here, are the (typically) most common of components, and represent all of the elements of a model (or modeling technique) that are not either processes or relations. The single identifying definition of an object in a model is that it retains identity and definition (and can be considered singly, in isolation from the rest of the model), until acted upon by some other component. Giving them definition, however, are their *attributes* (filling the same role as Characteristics do for processes). These give the objects their qualitative and quantitative distinction from other objects.

Processes, the second component in the method, is the subject of this paper, and its predecessor (Tolk and Turnitsa 2012). As the paper itself is dedicated to the definition of processes for modeling, the only succinct definition given here is that process components, while also retaining identity and definition (as with objects), in addition are each responsible for some change to one or more components in the model. So while no component (including the third component, relations, as seen in the next paragraph) will have its identity or definition changed unless it is done by some component in the model, the components that are responsible for such changes are the process components. That is their role.

Relations, as the third component in the method, are the means whereby other components are associated together in order to provide meaningful description of the system being represented by the model. Although association between two or more components may be in effect (and described by a relation), there are likely to be conditions to that association, and that is what the defining qualities of relations define. Because of this, the defining qualities of relations address just these sorts of limitations and effects, as well as other parameters defining the relation. These qualities are called *rules* in the case of relations (for the OPR method).

In order to make the following sections (on the four identified characteristics of processes) easier to follow, they may refer to either objects or relations, but for the purposes of this paper, these definitions of those two components, in addition to processes, given above should suffice to make the meaning clear.

2.3 Changes Within a Model

Following this definition of the three types of components in a model, if there is any change that occurs within or caused by the system represented by the model, it should be explained. As there can be changes to the model itself (or to the simulator that instantiates the model) that are out of control of the model, those changes are considered to be extra-modal to the model. All changes that take place, due to the intentional design of the model, are within the mode of the model, so they are modal changes. Every modal change should be represented by a process within the model definition. The risk of not describing some process within the model, that is still an accepted part of the mode of operation of the system the model is representing, is to lead to a model that does not adequately describe the behavior of the system being modeled.

3 PROCESS INITIALIZATION

All activity that takes place within a system, or that is represented by a model for a simulator of that system, takes place within some time line, and therefore should be ordered within that timeline. A process, representing some change that takes place in the timeline of the simulated model of the system, is no different – it takes place at some ordered point in time. This is so, whether the process is a single point in time (representing an instantaneous state change) or if it takes place over some duration or span of time (representing some continuous or time-spanning process). In both cases There is an initialization point. The initialization of a process can be simply defined as “ some defined condition which must be true for the process to begin”.

3.1 Process Initialization Definition

Consider that all of the components (objects, processes and relations) that comprise a model are a *set* of components. Also, each of these components can have defining qualities (objects have attributes, processes have characteristics, and relations have rules – these are the terms used for the defining qualities in this system). Each of those defining qualities may or may not have a parameterizing value associated with it. For a process to be initialized, it must be defined as some specific subset of all the components, defining qualities, and values that are in a certain configuration in order for the process to begin. In addition, the process should also define whether the process begins the first time, every time, or some specific number of occurrences of the initialization enabling configuration. This approach is similar to situation theory (Barwise and Cooper 1991) and shares the assumption of a small world view of all modeling, in that to describe a logical activity in the model, only the explicit elements that have bearing on that activity (process) are mentioned. Anything not mentioned is assumed to be non-consequential to the process's initialization.

3.2 Process Initialization Representation

The representation of the initialization can be an enumeration of the elements (components, defining qualities, values) that enable the initialization. More commonly, in the case of many processes, this will instead refer to "time" or some timeline being at a certain point. Within the OPR method, the time stream, and its passage, are considered to be a part of the model. The timeline is an object with an attribute representing the current_time, and the act of time passing (or progressing), is a process that increases the current_time attribute of the timeline. This allows, for the purposes of the method, the definition of process initialization to be enabled when the configuration of the value of current_time reaches a particular value (for instance, "12:00pm", or "8 minutes after initializing the simulator").

Following (McCarthy and Hayes 1969) such a description is a partial *situation*. A full situation would be a complete enumeration of all the elements at one particular instance, with all of their values As the configuration that could be reached if the partial situation becomes actualized, this is a *fluent*, and in fact, a propositional fluent.

3.3 Process Initialization Consideration

It has been mentioned that the existence of a process within the timeline of a simulation of the system must be ordered. It is not intended to imply that the ordering is always known before the running of the simulation, in an objective manner. It is possible that the initialization conditions of a process are not a particular point in the timeline of the simulation, but based on the existence or occurrence of some set of conditions – meaning that the defining qualities of one or more components (either an object, process or relation) should be some value indicated in the definition of the process as being necessary for the process to initialize.

It is possible, to consider that a process (a continuous process) will be in existence throughout the whole timespan that a simulation of the system represents. In that case, the model that the simulator performing the simulation is based on should have an initialization point for that process identified as being at, or before, the point where the simulated timeline starts.

It is possible that the entire occurrence of the process takes no time within the timeline of the simulator instantiating the model. If this is the case, then that sort of process can be referred to as an "instant" process. This is typical, but should not be limited to, processes that represent state changes. It is also possible that the occurrence of a process will take place over some span of time within the timeline of the simulator. This type of process can be referred to as an "extant" process.

The existence of multiple processes within a model offers up some consideration of the relationship (temporally) between those processes. For this purpose we can consider two processes, named X and Y. If both processes are instant, then there are three possibilities:

- Process X occurs before process Y
- Process X occurs at the same time as process Y
- Process X occurs after process Y

In the these instances, it is possible to have one of the processes have their initialization condition be represented as some configuration based on the initialization (which is the same as the halting point, in an instant process) of the other process. This is in following the definition of initialization being defined based on some element (in this case, a characteristic of another process) being in a certain configuration.

The existence of two or more processes that are extant brings up a richer set of definitions of the temporal relationship between each. This is in following (Allen 1983) (Allen 1984) and (Allen an Hayes 1990). The possible temporal relationships between extant processes are these:

- Process X occurs before Process Y
- Process X meets Process Y
- Process X overlaps Process Y
- Process X is during Process Y
- Process X starts Process Y
- Process X finishes Process Y

And the inverse of these. . .

- Process Y occurs after Process X
- Process Y is met by Process X
- Process Y is overlapped by Process X
- Process Y contains Process X
- Process Y is started by Process X
- Process Y is finished by Process X

And finally . . .

- Process X equals Process Y (in timing)

The complete timing of a extant process isn't apparent from just the initialization characteristic, but can be inferred if both the initialization characteristic, as well as the halting condition characteristic are known. These relationships also do not consider what happens when a modeled process has an effect that may affect or change one of its own characteristics (for instance, the halting condition). They are presented here, merely as consideration of what sorts of temporal relationships dynamic activities within a modeled system can be represented with regards to each other, by using the characteristics suggested in this work.

4 PROCESS EFFECT

If a process represents change, then there must be something that is being changed. As mentioned already, this can be something that changes instantly, or something that changes over time. As one of the characteristics of a process, depicting what the change the modeled process depicts of the referent system is of high utility in understanding the model and how it represents that system.

4.1 Process Effect Definition

The change that a process component defines is referred to here as the effect characteristic. The effect of the process is defined as some change to one or more defining qualities within the model. This includes any member of the set of all defining qualities (attributes, characteristics, or rules) or any member of the set of values affiliated with a defining quality. So an effect can be a change to any characteristic, attribute or rule – or it's associated value. The possible changes that an effect can include are of three types – creation, alteration and destruction. As an effect can be change to more than one defining quality, it can be defined as also more than one type of change. For instance, a process might have the effect of both creating one attribute (or some other defining quality) and destroying another. When creation of a defining quality, it is up to the definition of the effect to describe if a value will be affiliated with it, and what that value might be. When change to more than one defining quality is part of what the process effect describes, it may be that all of the changed defining qualities or their associated values are from the same component, or they may be associated with more than one component from the model. The way to either create a new component, or to destroy an existing component (that is, an entire object, process or relation), is through its paired identity defining quality. If an identity defining quality (every component has an identity defining quality) is destroyed, then the component that it granted identity to is also destroyed.

4.2 Process Effect Representation

As the process effect characteristic is the case where the process is defined as having some alteration on some other elements of the model, it is necessary here to rely on the relation component to associate the effect characteristic with the element being altered, created, or destroyed by the effect. This allows for the effect representation to be directly defined as the element being affected, or to be subjectively defined as "the element the relation is associating with". This allows the model to have processes enact changes to the associating relation or its defining qualities in order to show how the effect may or not be allowed at different times in the execution of the simulation timeline, without having to redefine the process in order to do so.

This system, of representation of the effect by relying on the association of a relation component, gives great flexibility in how the model is representing the referent system. If the relation is only available under certain conditions, then that can be part of the effect definition, or it can be part of the relation's rules (defining qualities) and their definition. In both cases, these can be affected by the effects of the same or other processes, as defined in order to make the behavior of the system apparent through the model. As OPR was originally devised in the dissertation work of (Turnitsa 2012) in order to make the formal description of modeling techniques as well as models a possibility, this flexibility was required in order to accommodate the different types of modeling techniques available. Using OPR as the framework for understanding the elements of a model, the role those elements play, and as the neutral way of comparing elements between models, still relies on this flexibility.

4.3 Process Effect Consideration

One of the things helpful to explain here is why identity has been elevated to a defining quality for each of the components within OPR. With identity as a defining quality for every component, and as the defining quality required to identify the identity of each component (meaning that a component isn't a component without an identity), then the destruction of the component is accomplished by a process effect destroying the identity. Equally, the creation of a component is accomplished by creating the identity.

The discussion of creation or destruction may be somewhat semantically misleading. After all, if these components or defining qualities are mentioned as part of the model, then they are already "created" as part of the model. However, when destruction or creation is described as part of process effect, it means that before a component or defining quality is "created" in the executed timeline of a simulation that is implementing the model, then that component or defining quality cannot be interacted with by other components, other than as the effect of creation (or associated to by a relation connecting the creation effect with the to-be-created element). Equally, if the component or defining quality is "destroyed" in the executed timeline of a simulation that is implementing the model, then after that point in the timeline, other elements of the model can no longer interact with that element (except to, perhaps, "create" it again).

As mentioned in the definition, there is no reason why a process, as its defined effect, need to be limited to a single element being created, destroyed, or altered. A process may (and many will) affect multiple elements in some way. A single element being affected only once by the defined effect characteristic can classify the process as a "simple process". Any other type of process may be considered a "multiple process". If there are multiple instances of the same effect that occur multiple times, but otherwise share definition, then that can be considered a compound process. Of a compound effect is on the same element, but recurring through the timeline of the simulator, then it is a serial compound process. If the compound effect is occurring all at the same point in the timeline of the simulator, but to different elements, then it is a parallel compound process. If there are multiple effects going on, but they are defintionally different (for instance, some creation, some destruction, to different defining qualities, in different components), then this sort of multiple process is a complex process. Again, a complex process where all of the effects are simultaneous in the timeline is a parallel complex processes, and one where they are somehow separated in the timeline, has the process classified as a serial complex process.

It is possible that each of the different effects within a multiple process could be teased out of that defined process, and in turn, used to create a different (separate) process with its own characteristics, and identity. This may be desirable in some cases, but as the purpose of the model is to enhance understanding of the system being modeled, and in describing systems, a single "process" might be mentioned, or semantically identified, but once modeled it can lead to multiple effects. For that reason, the possibility of one defined process component, with multiple effects as part of its definition, is allowed by the OPR method.

These definitions are not necessary for the understanding, but are presented here to show the possible breadth of representation of a process using the system of characteristics presented here. In order to capture the widest possible number of aspects of systems that may need to be modeled, this system of representation is as open as possible.

5 PROCESS BEHAVIOR

The description of how the effect takes place over the time spanned by a process is an important characteristic of processes, especially for extant processes. It is also important for instant processes in one specific way – it should be described if the effect takes place at the instant of the initialization and halting of the instant process, or if the effect is in place after the instant of the initialization and halting. With the case of an extant process, unless all process effects are uniformly distributed over the timespan of a the process (occurring regularly, and evenly spaced between initialization and halting), then it becomes necessary to define exactly how such effects do take place.

5.1 Process Behavior Definition

In the case of an extant process, where an effect takes place over an interval of time between initialization and halting, it is not necessary that the effect should take place at a single point during that interval. Because of this, an additional defining quality is required. The characteristic that describes the behavior of the process – that is, how the extant effect (or multiple instant or extant effects) occurs during the interval existence of an extant process during the simulation implementation of the model – is called the behavior characteristic. As mentioned, this is only necessary to describe the nature of the effect over the time of the process when the process is extant in nature. When a process is instant, then the effect should be described by the behavior characteristic as taking place at the point of initialization, or after the instant of instantiation.

5.2 Process Behavior Representation

Each effect in a multiple process requires its own representation by the behavior characteristic, so that there could be said to be multiple effect characteristics for a multiple process, and for each, a separate behavior characteristic. The description of the change that the effect brings about (creation, destruction, or alteration) should be described as part of the behavior characteristic, and defined as to how they take place within the streaming of time between initialization and halting. When a single element is being created or destroyed, then the single point at which this is considered to be accomplished, should be captured. When there is an alteration of a component's defining quality (or associated value), then the behavior of the change (effect) needs to be described as part of the behavior characteristic. In the case where a continuous effect takes place over time, this is very important for the model's ability to represent behavior.

5.3 Process Behavior Consideration

One of the things that the defined behavior characteristic can describe about a process is how the process effect in an extant process may be temporarily interrupted. Such cases have been identified and categorized in (Haller, Oren, and Kotinurmi 2006) and (Haller and Oren 2006). In the literature of systems modeling, this is also seen in (ISO 2004). These are presented here:

- Active – the ongoing state of an extant process that is progressing
- Suspended – an extant process that has temporarily halted
- Resumed – an extant process that, after a halt, has re-started progressing
- Cancelled – an extant process that was planned to enter either Active or Resumed, but that planning is curtailed
- Aborted – an extant process that has stopped before completing normally
- Halted – an extant process that has completed normally

When the behavior condition of the described process effect is considered to be *Active* then the changes to the timeline object that the behavior process refers to is progress normally by the timeline advancing process that is affecting its current_time attribute and value. This means that the relation between the effect characteristic and the affected element is true, at least for the rule that defines the relation to be true when time is progressing, so that the effect described by the behavior can proceed normally.

When the time state of the time Sub-model is considered to be *Suspended*, then the subjective rule of the temporal relation is evaluated to false so that the relationship between the behavior characteristic and the timeline current_time attribute is not allowed, and does not proceed as the behavior characteristic would normally dictate.

If the subjective rule of the temporal relation is restored to true, then the behavior characteristic is active for the effect once again – time for that behavior is *Resumed*. Operationally, there is little difference between Active and *Resumed*, however it is here because of additional information that may be captured when a *Suspension* of Time is planned as part of the model's operation.

Some process that has its initialization point defined in terms of the timeline current_time attribute being in a particular value, but the timeline is somehow prohibited from reaching that state can be said to be *Cancelled*. Note that *Cancelling* applies to process effect as described by the initialization characteristic, and not necessarily the entire model, although it could affect all of the processes within the model. A process could be *cancelled* by somehow affecting the relation between the process's behavior characteristic and the timeline component if it is defined that way, or other processes if defined subjective to their operation.

If either the timeline, the current_time attribute, or the time process that advances the value of the current_time attribute are destroyed by some process, then the time sub-model that they were part of is considered to be *Aborted*. There is little difference between the *Aborting* taking place because of destruction by a modal cause or an extra-modal cause, however with a modal cause it can be defined as part of the model. The model is not aware of extra-modal causes.

Finally, if the process that advances the timeline current_time attribute value is somehow stopped from advancing, either because it has completed its proscribed behavior, or some modal change to the relation between the process and the object, or some other reason – then the time sub-model is considered to be *Halted*. When the time sub-model of a model is *Halted* is the expected means of indicating that a Model that has its processes based on the time sub-model (for initialization, behavior, and halting) has reached its end.

Of course, the cases where it may be known in the model if it is aborted, halted, or cancelled refer to modal cases of interruption (as do all of these). Extra modal causes (such as the simulator instantiating the model being stopped, for some reason, or redefined outside of the scope of the representation of the system being simulated) cannot be accounted for.

6 PROCESS HALTING

If a process begins, then it follows also that a process also ends. The definition of when the process stops having an effect is important to understand, although depending on the behavior characteristic of the process, this is the one characteristic whose meaning may be redundant, given other characteristics. The definition and representation of halting is almost identical to that of initialization, so is not covered here.

6.1 Process Halting Consideration

Process halting may be during the temporal life of a simulation based on the Model, or it may be when that simulation ends. If it is something that can be anticipated within the system, then the model describing that system should describe halting characteristics for the process that make up that model. The halting characteristic is in form identical to an initialization characteristic, in that it can be either objectively defined based on some configuration of elements within the model, or subjectively defined based on some other process's characteristic. It is not necessary that a process have a halting characteristic. The temporal behavior of when the effects of the process take place are described by the behavior characteristic, and do not necessarily coincides with the halting characteristic of the process, especially where it is a multiple process which may have multiple effects that start or stop at different points. If a process has its initialization or halting characteristics defined as being subjective to the halting of another process, then however that defined process halts will be the indicator to the subjective characteristics.

7 IMPLICATIONS

There are a number of implications and corollary observations that can arise out of the set of characteristics described here, and as the dynamic nature of systems is a complex topic, the modeling of those systems may tend to get into these areas. Some coverage of the implications that are immediately apparent is warranted here, but there will likely arise other situations where these simple characteristic definitions can get into complex representations and may require further explanation.

7.1 Concurrent Process Effects

Consider the fact that two processes can have effects that modify the value of the same element within the model.

The following table shows an example: Suppose that process P1 (with effect characteristic C1) and process P2 (with effect characteristic C2) affect object O1, and its attribute A1. P1 has the effect of adding +3to the value of A1. P2 has the effect of doubling A1. Both of these processes are state changes to A1, so are instant processes. Now suppose that P1 and P2 occur at the same time, both affecting A1. Depending on how this gets adjudicated by the simulator (lacking any further definition by the model), we can get four different possible outcomes – all based on how the implementation of the model handles the temporal juxtaposition of P1 and P2. Here is a chart showing those four different outcomes, assuming that the attribute A1 started with a value of 4, before P1 and P2 are enacted.

Table 1: Example of Process and Object Interaction

Juxtaposition of P1 and P2	**Resulting value of A1, assume start value of 4**
P1 supersedes P2	A1 change to 7
P1 precedes P2	A1 changes to 14
P2 supersedes P1	A1 changes to 8
P2 precedes P1	A1 changes to 11

If just the state to state change of A1 is considered, then there is little understanding of P1 and P2 to give the implementer knowledge of how the relationship between P1 and P2 should manifest in an implementation.

7.2 Process models more than just Effects

There is a variety of different approaches to using process algebras, and situation calculus methods for showing the change of entities and their parameterization within a system. These have been defined as equivalent to a modern algebra, that is some method of representation for variables (sets of members), some additive function where the members can affect each other, and some identity altering function (multiplication), where the members can change each other. This has been presented well in (Baeten 2005).

Modeling and simulation requires more than this, for methods of modeling and the evaluation of models, to include the possibility that the model will be implemented in a simulator, and used as part of a simulation study, and therefore will have temporal considerations as part of the representation of the referent system. The characteristics presented here satisfy Baeten's treatment of a process algebra - the components and their defining qualities are the sets of members; association through relations to give a bigger meaning (affected identity through association) to any one of the members of the set of components and/or their defining qualities; and finally the method for altering members of the set through the process effect characteristic. However, the nature of a time metered implementation of the model, with the specific timing qualities that require definition of initialization, halting, and details of process behavior are beyond what a classic process algebra addresses itself with – unless time itself is considered to be part of the model. So from this consideration, we can see that either we are in new territory that process algebra's and situation calculi do not cover, or it is validation for the idea that time is a part of the model, and not some external control feature.

7.3 Sequential but Separate Process lead to Indistinguishable Results

Similar to the problems in consideration 7.1, above, this consideration concerns itself with modeling techniques that represent systems for digital computers by representing the discrete event changes to the values and states of system variables. With such techniques, if there are subsequent processes that have similar (or negating of each other) effects on the same defining quality (for instance, multiplying it by x.5 for one process, and multiplying it by x2 for the other process), then the state change for each time advance would show that effect taking place on the defining quality. Consider what happens when one of those processes halts running, and now the change to the state variable is different. If the modeling representation that is describing the system does not do a clear job of identifying the two processes as distinct from each other, it will proved difficult to understand how the ongoing effect to the state based variable has changed once one of the processes is no longer in effect. If the processes are modeled, with their defining characteristics and individual identity, then it is clear in the model what has occurred when the state changes abruptly alter due to changes to one but not both of those.

7.4 The model's definition is a degree of freedom of the model

So models produce useful information about dynamic systems, by handling the things that change within the system as degrees of freedom. What can change in the implementation of a model (representing the running of the referent system) is the degree of freedom. As a consideration, take the instance of when the definition of the model is the degree of freedom. This would occur whenever the referent system is one where it modifies itself, or alters itself (and its behavior) based on the dynamic activity it is performing. So, as with a differential equation, each time the model operates over time as a simulation, it may change its own defined behavior, as the normal outcome of its operation. In this case, the freedom to express the change that processes can bring about must be treated as being able to have effect on any part of the model (any component, in the terms of OPR). So if a process has the effect of changing the definition of itself, or some other processes, then when that process initializes and its effect takes place, then the overall behavior of the system will change. The model is change from time T1 to time T2, and at time T2 it now behaves differently. For this reason, treating processes as entities of the model that can be changed, just as objects and their attributes (and attribute-values) can be changed.

8 CONCLUSION

As mentioned in the introduction, the characteristics introduced here are derived from a functional decomposition of what it means to have dynamic change, and definitions of that change, within a modeled system that operates over time. Understanding what these characteristics are can benefit the methodology of modeling by providing the beginnings of a language for describing dynamic activity for modeling purposes.

Having a depiction, within a model, of the various processes that are part of the system being modeled, giving the four characteristics of those processes as defined here – initialization, effect, behavior, and halting – gives enough detail to enable a developer to produce suitable software methods in a simulator, and also gives enough detail that an engineer relying on the model to understand or analyze a system can also understand or analyze the dynamic nature of that system.

In addition, the taxonomy of different processes and process combinations suggested from examining these four characteristics begins to give us a way to describe the dynamic behavior of systems within models with greater depth than treating all such activities as a simple state change.

It is not assumed here that these are the only characteristics that will be found to be important for defining processes, nor that all of these characteristics are important all the time. The results from the research are offered up, however, as an indication of where this thinking into methodology led, and hopefully to assist others working in this area.

REFERENCES

Allen, J.F. 1983. “Maintaining Knowledge about Temporal Intervals.” *CACM* 26 (11): 832-843

Allen, J.F. 1984. “A General Model of Action and Time." *Artificial Intelligence* 23: 2.

Allen, J.F. and P.J. Hayes. 1990. ``Moments and Points in an Interval-Based Temporal Logic." *Computational Intelligence* 5(3):225-238.

Baeten, J.C.M. 2005. “A Brief History of Process Algebra.” *Theoretical Computer Science* 335:131-146.

Barwise, J., and R. Cooper. 1991. “Simple situation theory and its graphical representation.” In *Partial and Dynamic Semantics III*, Edited by J. Seligman. Edinburgh: Center for Cognitive Science, University of Edinburgh.

Haller, A. and E. Oren. 2006. "m3pl: A work-FLOWS ontology extension to extract choreography interfaces." In *Proceedings of the 2006 ESWC Workshop on Semantics for Business Process Management*. Lecture Notes in Computer Science 4011. Heidelberg: Springer.

Haller, A., E. Oren and P. Kotinurmi. 2006. An ontology for internal and external business processes. In *Proceedings of the 2006 International World-Wide Web Conference* 1055–1056.

International Standards Organization. 2004. *Industrial Automation Systems and Integration - Process Specification Language Standard*. Published as ISO Standard 18629.

McCarthy, J. and P.J. Hayes. 1969. “Some philosophical problems from the standpoint of artificial intelligence.” *Machine Intelligence* 4: 436-502.

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

Tolk, A. and C. Turnitsa. 2012. “Conceptual Modeling with Processes.” In *Proceedings of the 2012 Winter Simulation Conference*, Edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, 2641-2653. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Turnitsa, C. 2012. *Exploring the Components of Dynamic Modeling Techniques*. Doctoral dissertation from Old Dominion University, Modeling, Simulation and Visualization Engineering Department, College of Engineering. Norfolk, Virginia: Old Dominion University.

AUTHOR BIOGRAPHY

CHARLES TURNITSA is Assistant Professor for Modeling and Simulation at the TSYS School of Computer Science at Columbus State University, in Columbus Georgia. He also serves as the director of the GEMS (Gaming, Education, Modeling and Simulation) Institute at CSU. He received his M.S. in Electrical and Computer Engineering, and his PhD in Modeling and Simulation both from Old Dominion University. His B.S. in Computer Science is from Christopher Newport University. His email address is cturnitsa@gmail.com.