A Quantum of Continuous Simulated Time

Submission #4

Affiliation
City, Country
Email

ABSTRACT

In the context of discrete-event simulation, time resolution pertains to the time points at which events actually occur, whereas time precision constrains the time points at which events may possibly occur. Time precision is generally associated with the rounding of time values that takes place when a simulation is executed on a computer, yet here we study precision on a theoretical level. We find that while some models truly require a continuous representation of simulated time, a surprisingly diverse set of continuous-time models can be regarded as having an inherent level of time precision. We explore this concept by introducing the notion of an optimal time quantum, a simulation model property useful for establishing the set of durations which evenly divide all mathematically possible event times. A hierarchical method is presented for deriving the optimal time quantum from a model specification. The proposed theory compliments past and ongoing research on simultaneity, time representation, and formal reasoning.

Author Keywords

Simulated time; discrete-event simulation; time precision; hierarchical model analysis; simultaneous events.

ACM Classification Keywords

I.6.1 Simulation and Modeling: Simulation Theory.

1 INTRODUCTION

This paper presents new theory for analyzing the temporal properties of discrete-event simulation models. Such models are generally associated with continuous time. Yet surprisingly, we find that a wide variety of continuous-time models feature an inherent level of time precision, such that event times can only occur at multiples of some positive time quantum. This quantization of time is not merely the result of the rounding that takes place when a simulation is executed on a computer. Rather, a time quantum can often be derived from a model's mathematical specification.

Our contributions include the notion of an optimal time quantum, as well as a hierarchical method for deriving this property given a particular model. The analysis begins with components at the bottom of a model hierarchy, then progresses upward until it encompasses all models and sources of data. When deriving the optimal time quantum of a component model, the key idea is to assume the time points of all inputs to be quantized; later, when moving up one level in the hierarchy, this assumption becomes a constraint on simulated time. Upon reaching the top of the hierarchy, the overall optimal time quantum is either (*a*) positive, in which case it evenly divides all mathematically possible event times, or (*b*) zero, in which case the model truly requires continuous time for exact results. After applying the method to classic examples from Zeigler et al. [30], we find that a diverse set of simulation models do exhibit a positive optimal time quantum.

The paper is organized as follows. Section 2 reviews several decades of progress in understanding the role of time in simulation. Topics covered include the DEVS formalism, time systems, simultaneous events, time approximation in parallel simulation, computer representations of simulated time, and temporal logic. Section 3 provides an informal overview of time precision and other concepts associated with the optimal time quantum. It also outlines our hierarchical method for inferring this property. Section 4 formalizes these concepts. Three mathematical tests are proposed for analyzing time quanta inherent in atomic models from which hierarchies are constructed. As discussed in Section 5, the proposed theory complements past and ongoing research on simultaneity, time representation, and formal reasoning.

2 RELATED WORK

The concept of time is pervasive throughout the modeling and simulation literature. It also figures prominently in distributed computing and mathematics. Accordingly, our work touches on a broad range of research areas in which the role of time has been elaborated.

2.1 DEVS

In *Theory of Modeling and Simulation*, Zeigler et al. [30] explain how simulations imply an indexing of events based on some representation of time. Such a representation need not be a direct abstraction of physical time (i.e. measurable in seconds); but if it is, we refer to it as *simulated time*. Chapter 4 of the book reveals how two types of simulated time durations, denoted e and $ta(s)$, can be used in conjunction with the current state s to specify the intended behavior of any discrete-event simulation model. The associated formalism, called *DEVS*, is demonstrated in the same chapter using 15 example models chosen for their educational value. We base our method on DEVS theory, and illustrate it using models from Chapter 4.

2.2 Time Systems

We describe a *time system* as a means of associating mathematical values, called *time points*, to events. Our focus is on simulated time, but this is just one of many time systems relevant to simulation. Rooted in the work of Lamport [15],

TMS/DEVS 2016 April 3-6 Pasadena, CA, USA

logical time refers to time systems which capture one or more aspects of *causality*: the fact that one event may influence another through a state change or message [20]. Logical time is neither proportional nor necessarily consistent with simulated time. *Virtual time* [12] refers to a time system used to detect conflicts between optimistically processed events. A simple time system that accommodates discrete-event simulation employs time points of the form $[t, c]$, where t is simulated time and c is an integer counter. As explained by Nutaro and Sarjoughian [19], event times of this form are compared lexicographically with the c elements breaking the tie in cases where t elements are equal. In other words, c helps maintain an ordering of events within a single instant of simulated time. Variants of this time system, such as the $t^{+c} = t + c{\cdot}\epsilon$ notation proposed by Barros [1] and "superdense time" applied by Lee [17], differ partly in the types of operations that require c to be incremented.

2.3 Simultaneous Events

When simultaneous events are ordered using time point operations, there is often a choice of whether to augment simulated time with an integer counter—as in the $[t, c]$ time systems described above—or to simply offset an event time t by some short Δt. This choice has been observed by Kim et al. [14], while Barz et al. [2] criticize simulated time offsets for their potential impact on simulation results. It is true that an integer counter helps honor causal relationships [19, 1] and can address undesirable "chattering" conditions [17]. However, Wieland [28] argues that simulated time carries a degree of uncertainty that can be advantageously exploited by randomly offsetting event times. The random offsets are bounded by a duration parameter δ, called the "threshold of event simultaneity", and simulations are repeated with different offsets in the hopes of achieving robust statistics. An interesting comment is made that for "analytic simulations, δ can represent the *precision of the model* [our emphasis]". Our work elaborates on the concept of model precision.

2.4 Time Approximation in Parallel Simulation

In addition to statistical benefits, Wieland [28] asserts that exploiting temporal uncertainty by shifting event times can improve performance in parallel and distributed simulation. The idea is demonstrated by Zeigler et al. [29], who increase parallelism by quantizing simulated time according to a time granule d. Although delaying events by as much as d introduces rounding error, it allows greater numbers of events to be executed in parallel by synchronous simulators such as those based on the Parallel DEVS formalism [4]. Fujimoto [5] approximates time points in a similar manner to increase concurrency in the context of asynchronous distributed simulation algorithms. In this case simulated time is not quantized, but event times are independently deferred up to some interval size based on the relative progress made by communicating logical processes. Performance is shown to improve with interval size. Although these works explain and justify the use of an arbitrary duration parameter such as the time granule or the interval size, little guidance is provided on how to choose the parameter value. We show that reasonable options may be implied by the specification of the model being simulated.

2.5 Computer Representations of Simulated Time

A time system may have a number of plausible computer representations, and simulated time has several. Vicino et al. [26] observe that (*a*) the majority of DEVS-based simulators use a floating-point representation, and (*b*) rounding errors arising from floating-point time can cause events to be shifted or even reordered. Varga [25] cites limited precision and non-associativity of addition as reasons why OMNeT++ switched from a floating-point to a fixed-point representation of simulated time. A downside to fixed-point time is the need to prescribe a level of precision. CD++ [27], one of the few DEVS-based simulators supporting fixed-point time, features a 1-millisecond precision level suitable for real-time applications. In OMNeT++, the user can choose a level of precision to be imposed on a simulation run. Our work helps reveal which precision levels, if any, result in error-free event time computations for a given model specification. This motivates a discussion on the application areas which most benefit from fixed-point time as opposed to floating-point time (see Section 5). A third option for representing simulated time, based on rational number data types [26], is also discussed.

2.6 Temporal Logic

Looking beyond time systems, temporal logic encompasses theories that aid in reasoning without necessarily quantifying event times. In *A Catalog of Temporal Theories*, Hayes [7] presents a taxonomy of time-related concepts including "tense", "time interval", "temporal position", and notably "time quantum". Portions of the associated theory have been applied in a context closely related to simulation [18]. A classic essay by Lamport [16] advocates the use of temporal logic for reasoning about concurrent programs. Although we forgo temporal logic and focus little on concurrency, Lamport's work is relevant to our own for two reasons. First, the desire for a hierarchical method shapes the entire theory. Second, we take inspiration from the philosophy that formal methods can be applied manually—or automatically once suitable tools are available—to better understand the behavior of a system. Techniques exist for analyzing the temporal behavior of discrete-event simulation models; one example is the exploration of temporal structures by Traoré [23]. The temporal property we introduce is unusual in that it can be calculated without a formal representation of state, at least for many common models. Thus given only a partial model specification, a degree of formal reasoning becomes practical.

3 CONCEPTUAL ANALYSIS OF TIME PRECISION

This section explores how simulation models can have inherent levels of precision. We begin by contrasting time precision with the related concept of time resolution. The definitions below are identical except for the phrases in **bold**.

Time resolution *characterizes the frequency of time points* ***at which similar events occur***[1]*. In* ***discrete-time*** *simulation,* ***time resolution*** *may be expressed using a fixed* ***time step*** *that* ***separates consecutive*** *events times. A simulation with a* ***5-second time step*** *has a* ***5s time resolution****. The longer the*

[1]By *similar events*, we mean all events which affect a particular state variable or other time-varying observation.

***time step**, the coarser*[2] *the **resolution** and the larger the **discretization errors**.*

***Time precision** characterizes the frequency of time points **to which event times are rounded**. In **discrete-event** simulation, **time precision** may be expressed using a fixed **time quantum**[3] that **evenly divides all** event times. A simulation with a **1-nanosecond time quantum** has a **1ns time precision**. The longer the **time quantum**, the coarser the **precision** and the larger the **rounding errors**.*

In short, time resolution is often associated with time steps and discretization error, whereas time precision pertains to time quanta and rounding error.

If a time duration Δt has a particular time precision, one that is associated with finite positive time quantum δt, then Δt is a multiple of δt. Importantly, there are two ways that a nonzero δt can arise. First, it can be imposed by a simulator that uses a fixed-point representation of simulated time; that is, a representation where all time values can be expressed as $m{\cdot}\delta t$ for some integer multiplier m and some preselected δt. Alternatively, δt may be implied by the specification of a simulation model. We refer to this second type of δt as a *conformant time quantum*, and we call its associated precision level a *conformant time precision*. If a fixed-point representation is based on a conformant time quantum/precision, then the durations encountered in a simulation may be free of rounding error.

A conformant time quantum divided by a positive integer is also a conformant time quantum of the same model. We call a model's longest such quantum the *optimal time quantum*, and the associated precision is the *optimal time precision*. A fixed-point time representation based on an optimal time quantum δt not only conforms with the model specification, but also minimizes the multiplier component m of all time values $m{\cdot}\delta t$. Once we have determined a model's optimal time quantum, we can obtain a (likely complete) set of conformant time quanta by dividing by positive integers.

To obtain a model's optimal time quantum, a number of assumptions are necessary. Before listing them, let us distinguish among three types of time durations: elapsed durations, planned durations, and duration parameters. An *elapsed duration* Δt_e is the finite amount of simulated time separating two consecutive events of the same model instance. A *planned duration* Δt_p is the possibly infinite amount of simulated time separating one event from the same instance's most imminent scheduled event. If the scheduled event occurs, the preceding Δt_e and Δt_p are equal. If the scheduled event is preceded by an incoming message, then $\Delta t_e \leq \Delta t_p$. The third type of time duration is a *duration parameter*, which is known at the outset of a simulation and affects the elapsed and planned durations encountered during the simulation. A fixed time step is one of many types of duration parameters.

[2]The phrase *higher resolution* is confusing because it corresponds to a shorter time step; we prefer the phrase *coarser resolution* (the opposite being *finer resolution*), and the same applies to precision.

[3]Our definition of *time quantum* is not be confused with the Global Virtual Time computation algorithm of the same name [3], which temporally partitions events using numbers attached to messages.

When seeking a model's optimal time quantum, our assumptions are as follows:

Assumption 1: All duration parameters represent a rational number of seconds. Note that we are not referring to rational number data types, as in [26], but rational numbers in a mathematical sense. Thus even the computed value `sqrt(7.0)` is rational, since floating-point numbers have a finite number of binary digits. We assume neither rational elapsed durations, as in Rational Time-Advance DEVS (RTA-DEVS), nor rational planned durations, as in RTA-DEVS and Finite & Deterministic DEVS (FD-DEVS) [11]. If the optimal time quantum turns out to be positive and rational, then all elapsed durations and finite planned durations will be rational as well, but this is something we must calculate on a per-model basis.

Assumption 2: All model instances undergo an initialization event at simulated time point $t = 0$. This assumption is not fundamental to the proposed theory, but rather a matter of convenience. It guarantees that if all elapsed durations and finite planned durations are multiples of some δt, then so are all event times. In the future, our method could be generalized to accommodate other initialization patterns.

Assumption 3: The model hierarchy encompasses all sources of time-varying data, including input sequences or information received in real-time from an embedded system or interactive user interface. This assumption can be seen as a step in the direction of the "model everything" philosophy [24]. Although it is good practice to keep a domain-specific model separate from any associated *experimental frame* [30], we assume the frame is itself a component in an upper level of the model hierarchy. In future work, the domain-specific model and the experimental frame can be explicitly differentiated.

There is a fourth assumption, but it applies only in the context of a particular model and a potential time quantum. Since it enables further reasoning, we refer to it as the *Premise*.

Premise: When considering whether some time quantum δt conforms with a model specification, the simulated time points associated with inputs are *assumed to be multiples of* $\delta t/j$ for any positive integer j. This concept is best explained in steps. First, choose any $j \in \{1, 2, \ldots\}$. Second, assume input times are multiples of $\delta t/j$. It follows from Assumption 2 that elapsed durations are also multiples of $\delta t/j$. Now we ask whether all finite planned durations are guaranteed to be multiples of $\delta t/j$. If the answer is yes for all j, then δt is a conformant time quantum.

The rationale for the Premise is to enable hierarchical analysis. Instead of considering a complex model as a whole to determine whether its event times are quantized, the Premise allows an optimal time quantum to be determined for each component in a bottom-up fashion. When analyzing a single component, one temporarily ignores the temporal properties of any source of time-varying data. These data sources are themselves components on account of Assumption 3, so their temporal properties will eventually be accounted for as part of the encompassing analysis.

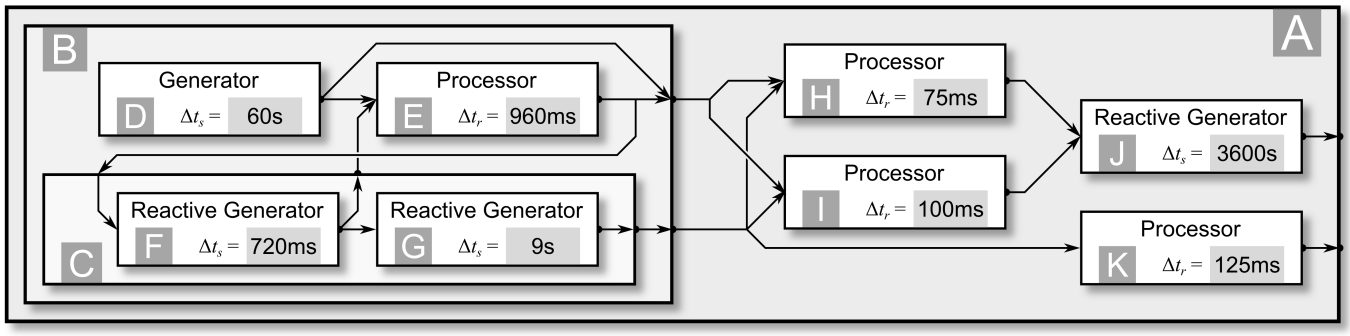

Figure 1. A hierarchy of model instances labeled A through K. Time steps Δt_s for [reactive] generators and response durations Δt_r for processors are indicated.

Let us illustrate the hierarchical method with an example. Figure 1 shows a model hierarchy consisting of 11 model instances labeled A through K. Three of these are instances of coupled models: Instance A at the top of the hierarchy; Instance B one level down; and Instance C nested within B. Each of these levels contains instances of atomic models: D is an instance of the Generator model; F, G, and J are instances of the Reactive Generator model; and E, H, I, and K are instances of the Processor model. These three classic examples of atomic models are described below.

Generator: This model produces outputs at regular intervals spaced according to the *time step* duration parameter Δt_s. The temporal behavior of a generator model is described by a DEVS model in Chapter 4 of *Theory of Modeling and Simulation* [30]. Although DEVS models traditionally exclude initialization events, our method compels us to make reasonable assumptions about the beginning of a simulation run. For generator models, we assume the first output occurs at $t = \Delta t_s$, one time step beyond the initialization event. This type of convention is typical of DEVS model implementations. It conveniently ensures all events times are divisible by the time step.

Reactive Generator: This model (*a*) produces outputs at regular intervals according to time step Δt_s, and (*b*) can accept an input at any time to influence subsequent output values. An example of a reactive generator is the "ramp" model in Chapter 4 of [30]. Again, we assume the first output occurs at $t = \Delta t_s$. This ensures that each *planned event*—an event triggered by the elapsing of a planned duration (and possibly associated with an output)—occurs at a time point divisible by the time step. However, each *unplanned event*—an event triggered by an "unplanned" input—may occur at any real-valued point in simulated time.

Processor: This model either (*a*) waits indefinitely for an input to process, or (*b*) produces an output in response to an input after a delay equal to the *response duration* parameter Δt_r. As in Chapter 4 of [30], we assume that after a waiting processor receives an input, all subsequent inputs are ignored until the output is produced and the processor returns to a waiting state. We further assume a processor is initially waiting. In the case of a processor model, both planned and unplanned events may occur at any real-valued point in simulated time.

A reactive generator's unplanned events and all of a processor's events occur at real-valued time points that need not adhere to any particular level of precision. Yet when they serve as components in a model hierarchy such as that of Figure 1, the overall model may feature a time quantum. Figure 1 indicates the values of the duration parameters Δt_s and Δt_r of all atomic model components. In milliseconds, the parameter values are 60000, 960, 720, 9000, 75, 100, 3600000, and 125. The greatest common divisor of these values is 5. We observe that a duration of 5ms will evenly divide all event times for this model hierarchy, including those associated with the processor instances and with the inputs of the reactive generators. We consider 5ms to be the optimal time quantum of the overall model, while 5ms, $\frac{5}{2}$ms, $\frac{5}{3}$ms, $\frac{5}{4}$ms, etc., are conformant time quanta that will also divide the event times evenly.

The example reveals that after coupling atomic models which individually permit real-valued event times, we may counterintuitively end up with a hierarchical model that constrains event times to some inherent level of precision. In the case of Figure 1, we observe an optimal time quantum of 5ms.

In practice, model hierarchies may be broader and deeper than that in Figure 1, and atomic model components are often much more complicated than generators, reactive generators, and processors. Thus rather than analyzing the entire composition of a model as a whole, we should seek a hierarchical method which starts at the atomic components and progresses upward until all atomic and coupled models have been included in the analysis.

For each of the three particular atomic models in Figure 1, we assert that its optimal time quantum is simply its duration parameter. Thus the optimal time quantum of Reactive Generator G is 9 seconds, and that of Processor H is 75 milliseconds. For any coupled model in general, we adopt the heuristic that the optimal time quantum is the greatest common divisor (GCD) of that of its components. So the optimal time quantum of C is the GCD of 720ms and 9s, which is 360ms. The optimal time quantum of B is the GCD of 360ms, 60s, and 960ms, which is 120ms. Finally, The optimal time quantum of A is the GCD of 120ms and the duration parameters of H through K, which yields 5ms as expected. This hierarchical method is compelling in its simplicity, but of course we must justify the fact we used the duration parameters as the optimal time quanta of the atomic models.

Consider Reactive Generator G. We assert that its optimal time quantum is 9 seconds. Yet we can see that it will receive an input from Reactive Generator F after 720ms. Thus G will undergo an unplanned event at $t = 720$ms, which is not evenly divisible by its 9s optimal time quantum. This is where the Premise comes into play. We ignore all sources of time-varying input data on the grounds that such sources will be treated separately. So when analyzing G, we ignore F and apply the Premise in its place. The Premise allows us to assume inputs will arrive at time points evenly divisible by $9/j$ seconds for any positive integer j. We can prove that the output times will then also be divisible by $9/j$ seconds, and so we accept 9 seconds as the optimal time quantum of G (see proof in next section). Once we discover that the optimal time quantum of F is 720ms, we should realize that only time quanta of the form $360/j$ milliseconds conform with both components, and so 360ms is the optimal time quantum of C. The Premise allows us to pursue inherent levels of time precision in a hierarchical fashion using the GCD as a heuristic.

Not every atomic model has an optimal time quantum equal to its duration parameter. One can imagine a generator in which each time step is randomly selected between two duration parameters $\langle \Delta t_s \rangle_a$ and $\langle \Delta t_s \rangle_b$. In that case the GCD of $\langle \Delta t_s \rangle_a$ and $\langle \Delta t_s \rangle_b$ is the optimal time quantum. A time series can be regarded as a specialized generator in which outputs are separated by irregular yet predetermined durations. The GCD of these durations is the optimal time quantum of the time series.

For some models, all finite positive durations are conformant time quanta. One example is a processor with a response duration of zero. Another common example is a queue model, assuming it has no internal delays. We consider the optimal time quantum of such a model to be ∞. If it appears as a component in a model hierarchy, it has no effect on the conformant/optimal time precision of the encompassing model.

Some models have no conformant time quanta at all. Examples include (*a*) a variant of the generator in which delays between outputs are randomly selected from a continuous probability distribution (e.g. exponential), or (*b*) a quantized integrator as defined in Chapter 16 of [30]. On a theoretical level, such a model truly requires continuous time. It has an optimal time quantum of zero, and imposes this property on all encompassing coupled models. A hierarchy with an optimal time quantum of zero is all but certain to produce temporal rounding errors, yet Section 5 explains how the proposed theory may still be useful for analyzing a subset of the hierarchy. Furthermore, the theory could be extended to include explicit levels of time precision aimed at controlling rounding error.

4 FORMAL ANALYSIS OF TIME PRECISION

This section provides a formal theory to analyze the time precision inherent in atomic models from which hierarchies are constructed. We previously asserted that generators, reactive generators, and processors have optimal time quanta equal to their respective duration parameters: Δt_s, Δt_s, and Δt_r. Here we prove the assertion for reactive generators.

Our work is greatly influenced by DEVS theory. In particular, elapsed and planned durations Δt_e and Δt_p are essentially e and $ta(s)$ from DEVS [30]. Yet we use our own notations, partly to emphasize the fact that a comprehensive model specification is not a prerequisite for our method. It is good practice to start with a DEVS model, but it is not crucial for our purposes. More importantly, while the concept of state is relevant to us, the conventional state representation s is not needed in our formulas.

Our conventions are as follows. Every instance undergoes an initialization event at $t = 0$. This yields a state, which we ignore, and planned duration $\langle \Delta t_p \rangle_0$. From here on the instance undergoes only planned and unplanned events: a total of n where $n \leq \infty$. Every new event (planned or unplanned) is separated from the previous event (planned, unplanned, or initialization) by an elapsed duration $\langle \Delta t_e \rangle_i$. The new event produces a new state and a new planned duration $\langle \Delta t_p \rangle_{i+1}$. If the new event is planned, the previous planned duration must have elapsed, so $\langle \Delta t_e \rangle_i = \langle \Delta t_p \rangle_i$. If the new event is unplanned, it is possible that a shorter duration of time elapsed, so $\langle \Delta t_e \rangle_i \leq \langle \Delta t_p \rangle_i$. The constraints below apply to all models. Note that whenever we include i, the formula must hold for all $i \in \mathbb{N}_{<n}$. The subscript $i+1$ appears in (1) because there are n elapsed durations and $n+1$ planned durations.

$$\langle \Delta t_p \rangle_0, \langle \Delta t_p \rangle_{i+1} \in \left(\mathbb{R}_{\geq 0} \cup \{\infty\}\right) \tag{1}$$

$$\langle \Delta t_e \rangle_i \in \mathbb{R}_{\geq 0} \tag{2}$$

$$\langle \Delta t_e \rangle_i \leq \langle \Delta t_p \rangle_i \tag{3}$$

We now offer three tests to determine whether a time quantum δt is conformant with a model specification. Recall that the Premise allows us to assume $\delta t/j$ evenly divides all elapsed durations $\langle \Delta t_e \rangle_i$. The question is whether $\delta t/j$ evenly divides the resulting planned durations as well. Wherever we include j, the formula must hold for all $j \in \mathbb{N}_{\geq 1}$. In some cases, $\delta t/j$ can be replaced with δt without affecting the implications of a formula, and so we omit j when possible.

Base Condition: Formula (4) is part of all three tests below. It checks that the first planned duration $\langle \Delta t_p \rangle_0$ is a multiple of δt, which is required if $\langle \Delta t_p \rangle_0$ is finite.

$$\frac{\langle \Delta t_p \rangle_0}{\delta t} \in \left(\mathbb{N} \cup \{\infty\}\right) \tag{4}$$

Zero-Step Test: Formula (5) extends the Base Condition by testing the remaining planned durations $\langle \Delta t_p \rangle_{i+1}$. It is an extremely conservative test. That is, any δt that passes the test is a conformant time quantum, but there may be conformant time quanta which fail the test.

$$\frac{\langle \Delta t_p \rangle_{i+1}}{\delta t} \in \left(\mathbb{N} \cup \{\infty\}\right) \tag{5}$$

One-Step Test: Formula (6) extends the Base Condition by testing remaining planned durations while employing the Premise to each event in isolation. We assume the planned and elapsed durations preceding an event are quantized, then require that the next planned duration is also quantized. If this holds for every event, δt is a conformant time quantum. The One-Step Test is conservative. It may reject a δt that is in fact conformant, but it will accept all δt that pass the Zero-Step

Test and will often find additional conformant time quanta.

$$\left(\frac{\langle\Delta t_p\rangle_i}{\delta t/j} \in \big(\mathbb{N}\cup\{\infty\}\big)\right) \wedge \left(\frac{\langle\Delta t_e\rangle_i}{\delta t/j} \in \mathbb{N}\right) \\ \Rightarrow \left(\frac{\langle\Delta t_p\rangle_{i+1}}{\delta t/j} \in \big(\mathbb{N}\cup\{\infty\}\big)\right) \tag{6}$$

General Test: Formula (7) extends the Base Condition by testing remaining planned durations while employing the Premise to the entire history of the system. The history includes all elapsed durations $\langle\Delta t_e\rangle_k$ for $k \leq i$. This test can be regarded as a formal definition of a conformant time quantum. Unfortunately, it is difficult to prove (7) directly. The One-Step Test happens to be a proof of (7) by induction, and some generality is lost.

$$\forall j \in \mathbb{N}_{\geq 1},\ \forall i \in \mathbb{N}_{<n}, \\ \left(\forall k \in \{0,\ldots,i\},\ \frac{\langle\Delta t_e\rangle_k}{\delta t/j} \in \mathbb{N}\right) \\ \Rightarrow \left(\frac{\langle\Delta t_p\rangle_{i+1}}{\delta t/j} \in \big(\mathbb{N}\cup\{\infty\}\big)\right) \tag{7}$$

To apply the theory to a particular model, one may begin with a partial model specification that includes only the relationships among the various types of durations. The relationships are defined as a set of *simulated time constraints*, and below we give the constraints associated with the reactive generator model. As indicated by (8), we require the time step to be positive. As in (9), the first planned duration is always one time step. Formula (10) covers the two cases in which corresponding elapsed and planned durations are equal: first, an input may be triggering an unplanned event at the end of the time step, in which case the next planned duration is zero to produce an immediate planned event; second, a planned event may be occurring, in which case a future planned event is scheduled after another time step. Finally, (11) covers the case of an input received in the middle of a time step, in which case the elapsed duration must be subtracted from the planned duration to effectively reschedule the planned event.

$$\Delta t_s \in \mathbb{R}_{>0} \tag{8}$$
$$\langle\Delta t_p\rangle_0 = \Delta t_s \tag{9}$$
$$\langle\Delta t_e\rangle_i = \langle\Delta t_p\rangle_i \ \Rightarrow\ \langle\Delta t_p\rangle_{i+1} \in \{0, \Delta t_s\} \tag{10}$$
$$\langle\Delta t_e\rangle_i < \langle\Delta t_p\rangle_i \ \Rightarrow\ \langle\Delta t_p\rangle_{i+1} = \langle\Delta t_p\rangle_i - \langle\Delta t_e\rangle_i \tag{11}$$

Now we are ready to pursue the optimal time quantum of the reactive generator. We begin with the Base Condition (4), but using (9) we substitute the time step for the first planned duration.

$$\frac{\Delta t_s}{\delta t} \in \big(\mathbb{N}\cup\{\infty\}\big) \tag{12}$$

The only δt that satisfy (12) are of the form $\Delta t_s/j$, and the longest is Δt_s. So the time step might be the optimal time quantum, but we must confirm this with one of the tests. The Zero-Step test will simply reject Δt_s, since the subsequent planned durations $\langle\Delta t_p\rangle_{i+1}$ may be less than the time step.

And so we select the One-Step Test, and prove it as follows.

$$\left(\frac{\langle\Delta t_p\rangle_i}{\delta t/j} \in \big(\mathbb{N}\cup\{\infty\}\big)\right) \wedge \left(\frac{\langle\Delta t_e\rangle_i}{\delta t/j} \in \mathbb{N}\right) \tag{13}$$
$$\Rightarrow \left(\frac{\langle\Delta t_p\rangle_i}{\Delta t_s/j} \in \big(\mathbb{N}\cup\{\infty\}\big)\right) \wedge \left(\frac{\langle\Delta t_e\rangle_i}{\Delta t_s/j} \in \mathbb{N}\right) \tag{14}$$
$$\Rightarrow \left(\frac{\langle\Delta t_p\rangle_i}{\Delta t_s/j} \in \mathbb{N}\right) \wedge \left(\frac{\langle\Delta t_e\rangle_i}{\Delta t_s/j} \in \mathbb{N}\right) \tag{15}$$
$$\Rightarrow \left(\frac{\langle\Delta t_p\rangle_i}{\Delta t_s/j} \in \mathbb{N}\right) \wedge \left(\frac{\langle\Delta t_p\rangle_i - \langle\Delta t_p\rangle_{i+1}}{\Delta t_s/j} \in \mathbb{N}\right) \tag{16}$$
$$\Rightarrow \frac{\langle\Delta t_p\rangle_{i+1}}{\Delta t_s/j} \in \mathbb{N} \tag{17}$$
$$\Rightarrow \left(\frac{\langle\Delta t_p\rangle_{i+1}}{\Delta t_s/j} \in \big(\mathbb{N}\cup\{\infty\}\big)\right) \tag{18}$$
$$\Rightarrow \left(\frac{\langle\Delta t_p\rangle_{i+1}}{\delta t/j} \in \big(\mathbb{N}\cup\{\infty\}\big)\right) \tag{19}$$

Let us explain the above proof. We begin with the left-hand side of (6), then substitute Δt_s for δt since we are testing the time step. In (15), we exclude $\{\infty\}$ on the grounds that (8)–(11) disallow infinite planned durations. At this point we acknowledge that if $\langle\Delta t_e\rangle_i = \langle\Delta t_p\rangle_i$, as in (10), then $\langle\Delta t_p\rangle_{i+1}$ is either 0 or Δt_s and in either case we arrive at the right-hand side of (6) quite trivially. For this reason the proof follows the $\langle\Delta t_e\rangle_i < \langle\Delta t_p\rangle_i$ case, and we arrive at (16) with a substitution based on the right-hand side of (11). The simplification to (17) is a key step in the proof, and it is based on a property of subtraction: if $\delta t/j$ evenly divides both $\langle\Delta t_p\rangle_i$ and $\langle\Delta t_p\rangle_i - \langle\Delta t_p\rangle_{i+1}$, then it must evenly divide $\langle\Delta t_p\rangle_{i+1}$ as well. From here we generalize the expression by including $\{\infty\}$, then substitute δt for Δt_s to arrive at the right-hand side of (6). Thus Δt_s passes the One-Step Test and is necessarily a conformant time quantum. Since the Base Condition has already informed us there is no coarser conformant precision, we conclude that the optimal time quantum of a reactive generator model is equal to its time step.

The same theory can be applied the generator and processor models. The generator is a trivial case. Its simulated time constraints include (8), (9), and $\langle\Delta t_p\rangle_{i+1} = \Delta t_s$. The time step Δt_s passes both the Base Condition and the Zero-Step test, and is the optimal time quantum. The processor's simulated time constraints permit infinite planned durations, since a processor may "wait". Yet one can prove, in a manner quite similar to the (13)–(19) analysis, that Δt_r is the longest duration that passes the Base Condition and the One-Step Test. In other words, the optimal time quantum of a processor model is its response duration.

Such proofs are tedious, but one develops an intuition that a model's optimal time quantum is likely either (*a*) its sole fixed delay of whatever type, or (*b*) the GCD of all possible delays.

5 DISCUSSION

Our theory compliments past work in a number of areas related to simulation and time. Exploring these areas reveals potential applications and opportunities for future research.

5.1 Simultaneity

Optimal time quanta may be useful in the selection of statistical and concurrency-related duration parameters such as the threshold of event simultaneity δ of Wieland [28] or the time granule d of Zeigler et al. [29]. Wieland's δ threshold has two possible effects. If δ is extremely short, it will order simultaneous events randomly instead of imposing an arbitrary order. If δ is somewhat longer, it may reverse the supposedly unreliable order of nearly simultaneous events. The optimal time quantum is significant in this context, as it seems to separate the two effects. Choosing δ less than the optimal time quantum overwhelming affects simultaneous events, whereas a δ greater than the quantum will tend to encompass and reorder events with different simulated time points. With regard to Zeigler et al.'s time granule d, the optimal time quantum establishes a threshold below which d will have virtually no effect. Experimentation is needed to evaluate how well any d balances parallelism gains with sacrificed accuracy. Our theory suggests an obvious set of d values to test: multiples of the optimal time quantum, if it exists. The same heuristic may be useful for investigating the interval sizes of Fujimoto [5].

5.2 Time Representation

To some extent, our work supports the arguments of Varga [25] and Vicino et al. [26] against floating-point time values. A floating-point representation has no quantum, and may therefore cause temporal rounding errors even for a model with an inherent precision level. By contrast, a simple fixed-point representation based on a conformant time quantum δt can express all theoretically possible event times in the form $m{\cdot}\delta t$ with no rounding error. The use of an optimal time quantum is not critical, but will minimize the multipliers m.

Multiscale modeling and education are two areas where temporal rounding errors should cause concern. When integrating models of multiple scales, rounding errors associated with long durations (from a large-scale model) may be harmful in the presence of short durations (from a small-scale model). Thus a multiscale simulation should be run at a fine precision level, as in OMNeT++ [25], or a mechanism is needed to accommodate different fixed-point precisions at different scales. Progress has been made for diverse time steps (resolution) [6], but less so in the context of time quanta (precision). With regard to education, an ongoing challenge is to promote domain expert adoption of theory-based modeling and simulation techniques. When learning a formalism such as DEVS using simple examples, it helps immensely when a simulator produces the expected results. Fortunately, if a fixed-point time representation is used, simple models often do permit exact event times. In fact, all 15 example models in Chapter 4 of [30] have a positive optimal time quantum.

The relationship between our theory and rational number data types, as in [26], is complicated. If a model has a positive, rational, but unknown optimal time quantum, then a rational data type should automatically conform to any duration or event time. On the other hand, if the optimal time quantum is known, then the overhead of a rational data type seems unjustified since integer multipliers should suffice. It is important to recognize that a set of time values which share a quantum are not necessarily rational, and a set of time values that are rational do not necessarily share a quantum. Consider a nonterminating simulation with rational but quantum-less planned durations $\langle t_p \rangle_i = \frac{i}{i+1}$. As the durations are accumulated, a fixed-point current time variable will incur rounding error, but this may be preferable to the unbounded memory growth of an exact rational number representation. One compromise is the C++ Chrono Library [13], which combines rational precision levels resolved at compile time with integer multipliers computed at runtime.

5.3 Formal Reasoning

The following illustrates how a formal analysis of model behavior could be constructed from the proposed theory. Consider Instance J of Figure 1, and suppose we wish to know whether its inputs alternate between the two sources H and I in a ...HIHIHI... pattern. Processors H and I are triggered simultaneously from B, and the H output will precede the corresponding I output by 25ms. The question is whether H and I might receive inputs during this 25ms period, as this could disrupt the alternation pattern. Instead of considering the internal structure of B, we simply observe that its optimal time quantum of 120ms is greater than the response duration of I. After triggering both processors, B lacks the time precision necessary to intervene, so H and I outputs will alternate. Now suppose the response duration of E is increased from 960ms to 1s. In that case the optimal time quantum of B decreases from 120ms to 40ms, and we lose our guarantee that H and I outputs alternative. In fact, experimentation confirms that this small change to Figure 1 causes an occasional ...HH... pattern. Note that our analysis focuses on a subset of the model hierarchy, and does not require downstream components such as J and K to have any conformant time quanta.

The omission of s, or any other formal representation of state, simplifies all the analyses presented in this paper. That said, incorporating state into the proposed theory is a promising future research direction that could build on advancements in time-state representation for validation [8], verification [9], and general purposes [10]. Even in its current form, the theory may expand the utility of existing model-checking and behavior analysis techniques. The Figure 1 example shows how atomic models with real-valued durations and infinite state sets can form a hierarchy with rational durations, as assumed by RTA-DEVS [21], and a finite number of states, as required by FD-DEVS [11].

6 CONCLUSION

Recent work by Sarjoughian and Sundaramoorthi [22] on visualizing superdense time attests to the enduring importance of time representation in discrete-event simulation. Superdense time allows events to be ordered even if they share the same simulated time point t. Our work shows that when t does increase from one event to the next, the duration between the events is surprisingly often constrained—on a theoretical level—to be a multiple of a quantum δt. In fact, all 15 example models in Chapter 4 of *Theory of Modeling and Simulation* [30] feature an inherent level of precision in the form of a positive optimal time quantum. We define this property such that it can be (*a*) derived formally from an atomic model

specification, and (*b*) analyzed hierarchically to determine if a complex model treats time as quantized. The theory has several potential applications, including behavioral analyses that can be used even when some components feature optimal time quanta of zero and thus require continuous time.

REFERENCES

1. Barros, F. J. Event ordering in discrete flow models. In *IMACS World Congress: Scientific Computation, Applied Mathematics and Simulation* (2005).
2. Barz, C., Göpffarth, R., Martini, P., and Wenzel, A. A new framework for the analysis of simultaneous events. In *Proceedings of the Summer Computer Simulation Conference (SCSC)* (2003).
3. Chen, G., and Szymanski, B. K. DSIM: Scaling time warp to 1,033 processors. In *Proceedings of the Winter Simulation Conference (WSC)* (2005).
4. Chow, A. C. H., and Zeigler, B. P. Parallel DEVS: a parallel, hierarchical, modular modeling formalism. In *Proceedings of the Winter Simulation Conference (WSC)* (1994).
5. Fujimoto, R. M. Exploiting temporal uncertainty in parallel and distributed simulations. In *Proceedings of the Parallel and Distributed Simulation Workshop (PADS)* (1999).
6. Guo, S., Hu, X., and Wang, X. On time granularity and event granularity in simulation service composition (WIP). In *Proceedings of the Symposium on Theory of Modeling & Simulation (TMS/DEVS)* (2012).
7. Hayes, P. A catalog of time theories. Tech. Rep. UIUC-BI-AI-96-01, University of Illinois, 1996.
8. Hollmann, D. A., Cristiá, M., and Frydman, C. A family of simulation criteria to guide DEVS models validation rigorously, systematically and semi-automatically. *Simulation Modelling Practice and Theory 49* (2014), 1–26.
9. Hong, K. J., and Kim, T. G. Timed I/O test sequences for discrete event model verification. In *Artificial Intelligence and Simulation*, T. G. Kim, Ed., vol. 3397 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005, 275–284.
10. Hwang, M. H. Revisit of system variable trajectories (wip). In *Proceedings of the Symposium on Theory of Modeling & Simulation (TMS/DEVS)* (2013).
11. Hwang, M. H., and Zeigler, B. Reachability graph of finite and deterministic DEVS networks. *IEEE Transactions on Automation Science and Engineering 6*, 3 (2009), 468–478.
12. Jefferson, D. R. Virtual time. *ACM Transactions on Programming Languages and Systems (TOPLAS) 7*, 3 (1985), 404–425.
13. Josuttis, N. M. *The C++ Standard Library: A Tutorial and Reference*, second ed. Addison-Wesley, New York, NY, USA, 2012.
14. Kim, K. H., Seong, Y. R., Kim, T. G., and Park, K. H. Ordering of simultaneous events in distributed DEVS simulation. *Simulation Practice and Theory 5*, 3 (1997), 253–268.
15. Lamport, L. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM 21*, 7 (1978), 558–565.
16. Lamport, L. What good is temporal logic? *Information Processing 83* (1983), 657–668.
17. Lee, E. A. Constructive models of discrete and continuous physical phenomena. *IEEE Access 2* (2014), 797–821.
18. Manna, Z., and Pnueli, A. *The Temporal Logic of Reactive and Concurrent Systems*. Springer-Verlag, New York, NY, USA, 1992.
19. Nutaro, J. J., and Sarjoughian, H. H. A unified view of time and causality and its application to distributed simulation. In *Proceedings of the Summer Computer Simulation Conference (SCSC)* (2003).
20. Raynal, M., and Singhal, M. Logical time: Capturing causality in distributed systems. *Computer 29*, 2 (1996), 49–56.
21. Saadawi, H., and Wainer, G. Principles of Discrete Event System Specification model verification. *Transactions of the Society for Computer Simulation International 89*, 1 (2013), 41–67.
22. Sarjoughian, H. S., and Sundaramoorthi, S. Superdense time trajectories for DEVS simulation models. In *Proceedings of the Symposium on Theory of Modeling & Simulation (TMS/DEVS)* (2015).
23. Traoré, M. K. Analyzing static and temporal properties of simulation models. In *Proceedings of the Winter Simulation Conference* (2006).
24. Vangheluwe, H., Sun, X., and Bodden, E. Domain-specific modelling with AToM-3. In *International Conference on Software and Data Technologies (ICSOFT)* (2007).
25. Varga, A. *OMNeT++ User Manual (Version 4.6)*, 2014.
26. Vicino, D., Dalle, O., and Wainer, G. A data type for discretized time representation in devs. In *Proceedings of the International Conference on Simulation Tools and Techniques (SIMUTools)* (2014).
27. Wainer, G. CD++: A toolkit to develop DEVS models. *Software: Practice and Experience 32*, 13 (2002), 1261–1306.
28. Wieland, F. The threshold of event simultaneity. *Transactions of the Society for Computer Simulation International 16*, 1 (1999), 23–31.
29. Zeigler, B. P., Moon, Y., Kim, D., and Ball, G. High performance modelling and simulation: Progress and challenges. Tech. rep., University of Arizona, 1996.
30. Zeigler, B. P., Praehofer, H., and Kim, T. G. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*, second ed. Academic Press, San Diego, CA, USA, 2000.