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

A QUANTUM OF CONTINUOUS SIMULATED TIME

NameFirstAuthor
NameLastAuthor

Institution
Street Address
City, Zip, COUNTRY

ABSTRACT

Although discrete-event simulation models are typically associated with the notion of time as a continuous quantity, many such models can be simulated exactly using a uniform discretization of time based on a carefully selected time unit. This paper contributes general formulas and related tests for calculating the optimal time unit of a discrete-event simulation model from its specification. The proposed theory is applied to various types of models. An ability to reason about a model's optimal time unit provides a deeper understanding of several preexisting methods for handling simultaneous events, and lends credibility to the use of simulated time representations based on integers as opposed to floating-point numbers.

1 INTRODUCTION

> *Time flows not like a river but like the ticking of a clock, with "ticks" that are about as long as the Planck length: 10^{-43} second. Or, more precisely, time in our universe flows by the ticking of innumerable clocks—in a sense, at every location in the spin foam where a quantum "move" takes place, a clock at that location has ticked once.*
>
> – Lee Smolin, Atoms of Space and Time

Smolin (2014) explains that if the theory of loop quantum gravity is correct, neither space nor time is fundamentally continuous but rather they are both "made of discrete pieces". If these "pieces" exist, they would go unnoticed due to their immeasurably small size (10^{-99} cm^3 in volume and 10^{-43} s in duration). The idea is interesting in that it challenges such an intuitive and widespread assumption as the continuous nature of physical time. Our intention is to raise an analogous issue pertaining to simulated time.

In this paper we scrutinize the assumption that discrete-event simulations tend to treat simulated time as fundamentally continuous. Our contradictory observation is that many such simulations can in fact be performed with a uniform discretization of time without introducing any error into the timing of events. To properly understand this concept, there are two important points to note. First, we are speaking of a different type of "uniform discretization of time" than one encounters in discrete-time simulation, where every pair of consecutive events is separated by a common time step. In discrete-event simulation, what we observe is that there is often a *common time unit*, a nonzero duration of simulated time that evenly divides all durations between events. The second point is that our observation is theoretical in nature, not technological. It is well understood that quantities regarded as continuous, such as physical time, are often discretized out of necessity when represented on a computer. A floating-point time variable, for example, is a discretization in the sense that the number of distinct representable time values is finite. What we show is that time units are often implicit in specifications of discrete-event simulation models.

Our main contribution is a set of general formulas and related tests for calculating a model's *optimal time unit*, a property that helps one identify the longest common time unit for all possible simulation runs performed using the model. We demonstrate the theory by applying it to notable classes of discrete-event

simulation models, including the generator and processor models described in *Theory of Modeling and Simulation* (Zeigler, Praehofer, and Kim 2000). For models specified via the coupling of multiple submodels, the theory is applied in a modular fashion from the simplest models at the bottom of a hierarchy to the increasingly complex models at higher levels.

Conveniently, we find that intuition, guided by a general understanding of the proposed theory, can in many cases help one identify the optimal time unit without rigorously applying the formulas. The generator model is a trivial case: a discrete-time model in which the fixed time step serves as the optimal time unit. Yet even the processor model has a nonzero optimal time unit, despite its ability to receive a message at any point in simulated time. The explanation for this apparent neglect of received messages is that they can be analyzed separately. One must recognize that (*a*) a processor's input messages come from a source, (*b*) the source can be treated as a model with its own optimal time unit, and (*c*) the source and processor together constitute a third model with yet another optimal time unit. For example, consider a processor with a 15-second response duration. Its optimal time unit is 15 seconds. If it receives messages from a generator with a 10-second time step, the optimal time unit of the generator is 10 seconds and that of the overall model is 5 seconds. In many cases, this simple greatest common divisor heuristic works well.

In addition to advancing the current body of theoretical knowledge pertaining to the analysis of simulation model specifications, this work has two noteworthy implications. First, there are a number of preexisting methods in which event times are deliberately offset as a means of handling simultaneous events. Optimal time units clarify the point at which these offsets are long enough to cause events to be reordered. Second, computer representations of simulated time may be based on integers or binary floating-point numbers. Although the floating-point option seems most consistent with the notion of time as a continuous quantity, integer-based time representations gain credibility from the counterintuitive observation that a diverse set of discrete-event simulation models treat time as if it were quantized.

2 SIMULATED TIME DURATIONS AND CONSTRAINTS

To reason effectively about time units, we must identify a set of formal conventions for characterizing the behavior of any discrete-event simulation model with respect to simulated time. One option would be to use the Discrete Event System Specification (DEVS) formalism (Zeigler, Praehofer, and Kim 2000), which provides the desired level of generality. However, we choose simpler conventions for two reasons. First, we do not want to give the impression that one must specify his/her model using DEVS in order to calculate its optimal time unit. It is helpful to begin with a DEVS model, but one may also start from a conceptual model, or from a model specified according to an event-scheduling procedure. The second reason we avoid conventional DEVS nomenclature is that functions and transitions involving input, output, and state values, which figure prominently in the formalism, are unnecessary for our purposes; we are concerned with only the constraints imposed by a model on the durations which govern the timing of events. That said, the conventions presented here are heavily inspired by DEVS theory.

When simulated, an instance of a discrete-event simulation model produces a sequence of events $[\mathsf{A}, \mathsf{B}, \mathsf{C}, \ldots]$, with corresponding event times $[t_\mathsf{A}, t_\mathsf{B}, t_\mathsf{C}, \ldots]$. The first event, which may neither receive nor send messages, is the *initialization event*. Each of the remaining events is either a *planned event*, triggered by the instance itself and capable of sending a message, or an *unplanned event*, triggered by the receiving of a message and capable of reading that message.

Although state plays only an indirect role in the analysis of a model's time units, it is worthwhile noting that each state s_i occurring between successive events coincides with two types of durations. A *planned duration* $\langle \Delta t_p \rangle_i$ represents a duration of simulated time that, if uninterrupted by an incoming message or the end of the simulation run, would be followed by a planned event. For a typical event-scheduling simulation algorithm, the planned duration is the time that must elapse before the most imminent scheduled event occurs, regardless of whether the event actually does occur. For a DEVS model, where one has defined the time advance function ta, we have simply $\langle \Delta t_p \rangle_i = ta(s_i)$. The other type of duration is an *elapsed duration* $\langle \Delta t_e \rangle_i$, which measures the simulated time between two consecutive events of any type. When

a simulation run begins, the first state s_0 occurs after initialization event A and coincides with durations $\langle \Delta t_p \rangle_0$ and $\langle \Delta t_e \rangle_0$. A total of n state transitions follow before the simulation ends with the model instance in state s_n. At this point the final planned duration $\langle \Delta t_p \rangle_n$ is known, but the elapsed duration $\langle \Delta t_e \rangle_n$ does not exist because the subsequent event never occurred.

Figure 1 illustrates the notation we have established using a scenario involving an initialization event followed by a hypothetical sequence of planned and unplanned events. On the timeline at the top of the diagram, solid black dots represent events. Planned events such as B and C are shown with a dashed arrow pointing out, representing the possibility of an outgoing message. Unplanned events such as D and F are shown with a solid arrow pointing in, representing the incoming message that triggered the event. The dot with the circle around it indicates that the simulation ends at simulated time t_G with the instance in state s_5. Note that $n = 5$ for this scenario. Also note that any two consecutive event times may be equal, in which case the elapsed duration measured between them would be zero.

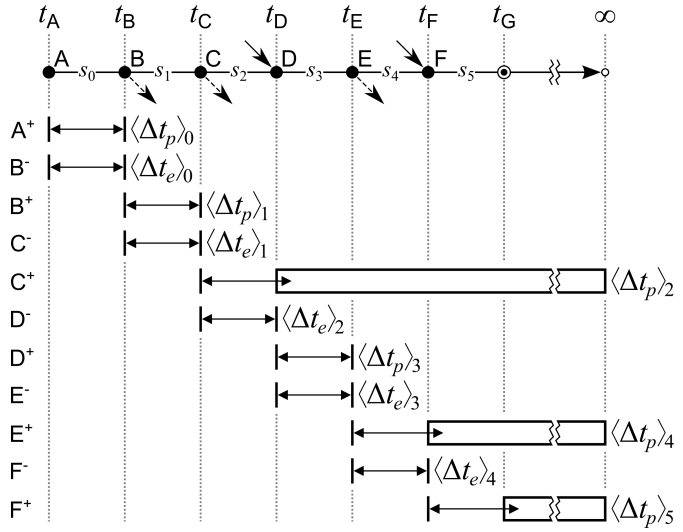

Figure 1: Constraints on planned durations $\langle \Delta t_p \rangle_i$ and elapsed durations $\langle \Delta t_e \rangle_i$ for the following sequence of event types: initialization (A), planned×2 (B, C), unplanned (D), planned (E), unplanned (F).

Beneath the timeline in Figure 1, each planned and elapsed duration is depicted from top to bottom in the order in which it is encountered during the simulation. The first row is labeled A^+, indicating that the first planned duration $\langle \Delta t_p \rangle_0$ is encountered immediately after event A is processed. Because event B is a planned event, it is clear that $\langle \Delta t_p \rangle_0$ elapses uninterrupted by any incoming message, and therefore $\langle \Delta t_p \rangle_0 = t_\mathsf{B} - t_\mathsf{A}$. The second row labeled B^- indicates that the first elapsed duration $\langle \Delta t_e \rangle_0$ is encountered immediately before event B is processed, and that its magnitude must also be $t_\mathsf{B} - t_\mathsf{A}$.

We use diagrams similar to Figure 1 largely to illustrate constraints on planned durations. For example, consider planned duration $\langle \Delta t_p \rangle_2$ that is encountered immediately after the processing of event C. While the diagram clearly shows that the duration is measured starting from time t_C, the box on the right of the interval is meant to indicate that the end point can be anywhere in the box or on either edge. The end point might be t_D, in which case the incoming message would have coincided with the elapsing of $\langle \Delta t_p \rangle_2$. If not t_D, the end point might be any subsequent point in time, including but not necessarily one of the later event times such as t_E or t_F. The end point might also be infinity, in which case $\langle \Delta t_p \rangle_2 = \infty$, meaning that no events had been scheduled. Observe that whenever the next event is planned, the planned duration is equal to its corresponding elapsed duration; whenever the next event is unplanned, the planned duration is at least the elapsed duration but possibly longer.

To express simulated time durations and constraints independent of any particular scenario, we transition from diagrams to formulas. We start by arranging the durations $\langle \Delta t_p \rangle_0, \langle \Delta t_e \rangle_0, \langle \Delta t_p \rangle_1, \langle \Delta t_e \rangle_1, \ldots$ in a duration

vector, where planned and elapsed durations alternate in a manner consistent with Figure 1. A simulation run with n state transitions will have n elapsed durations and $n+1$ planned durations, meaning that the duration vector has $2{\cdot}n+1$ elements in total. Now we let $\mathscr{T}_M$ be the set of all possible duration vectors permitted by discrete-event simulation model M. For any model M, $\mathscr{T}_M$ is partially defined by (1).

$$\mathscr{T}_M \subseteq \left\{ \left[\langle \Delta t_p \rangle_0, \langle \Delta t_e \rangle_0, \langle \Delta t_p \rangle_1, \langle \Delta t_e \rangle_1, \ldots, \langle \Delta t_p \rangle_{n-1}, \langle \Delta t_e \rangle_{n-1}, \langle \Delta t_p \rangle_n \right] \;\middle|\; n \geq 0 \right\} \tag{1}$$

Let us refine the definition of $\mathscr{T}_M$. We assume that a simulation can be terminated at any point after the completion of the initialization event and zero or more subsequent events. What this means is that for any possible simulation run with at least one state transition, there is an otherwise identical simulation run with one fewer transition. Omitting the last state transition means trimming the last pair of durations, one elapsed and one planned, from the end of the duration vector. This gives us (2), which will play a key role in how we formalize optimal time unit calculations.

$$\left(\left[\langle \Delta t_p \rangle_0, \ldots, \langle \Delta t_p \rangle_n, \langle \Delta t_e \rangle_n, \langle \Delta t_p \rangle_{n+1} \right] \in \mathscr{T}_M \right) \quad \Rightarrow \quad \left(\left[\langle \Delta t_p \rangle_0, \ldots, \langle \Delta t_p \rangle_n \right] \in \mathscr{T}_M \right) \tag{2}$$

To complete the general definition of $\mathscr{T}_M$, we observe constraints on the planned and elapsed durations in the duration vectors. These general constraints may be further refined for a specific model. The first constraint is that a planned duration must be a nonnegative real number or infinity, as in (3).

$$\forall i \in \{0, \ldots, n\}, \quad \langle \Delta t_p \rangle_i \in \mathbb{R}_{\geq 0} \cup \{\infty\} \tag{3}$$

An elapsed duration must be a nonnegative real number, but may not be infinity. Furthermore, it must be at most as long as its corresponding planned duration. These constraints, which should be evident from Figure 1, are expressed in (4) and (5). For formulas such as these where i is not explicitly quantified, the condition must hold for all $i < n$. We omit the quantification $\forall i \in \{0, \ldots, n-1\}$ to improve readability.

$$\langle \Delta t_e \rangle_i \in \mathbb{R}_{\geq 0} \tag{4}$$

$$\langle \Delta t_e \rangle_i \leq \langle \Delta t_p \rangle_i \tag{5}$$

3 PERMISSIBLE AND OPTIMAL TIME UNITS

Here the guiding objective is to be able to take any simulation that has yet to be performed, examine its configuration and underlying model, and determine the longest time unit that is guaranteed to divide all elapsed durations and all finite planned durations encountered during the run. We will work toward a practical method for obtaining these common time units, but we begin with a general albeit unwieldy formula. Let $\langle \Delta \mathbb{T}_u \rangle_M$, defined in (6), be the set of permissible time units associated with model M.

$$\langle \Delta \mathbb{T}_u \rangle_M = \left\{ \Delta t_u \in \mathbb{R}_{>0} \;\middle|\; \begin{array}{l} \forall \left[\langle \Delta t_p \rangle_0, \langle \Delta t_e \rangle_0, \langle \Delta t_p \rangle_1, \langle \Delta t_e \rangle_1, \ldots, \langle \Delta t_p \rangle_n \right] \in \mathscr{T}_M, \\ \left(\forall i \in \{0, \ldots, n-1\}, \; \dfrac{\langle \Delta t_e \rangle_i}{\Delta t_u} \in \mathbb{N} \right) \Rightarrow \left(\langle \Delta t_p \rangle_n = \infty \right) \vee \left(\dfrac{\langle \Delta t_p \rangle_n}{\Delta t_u} \in \mathbb{N} \right) \end{array} \right\} \tag{6}$$

A *permissible time unit* is a duration Δt_u such that if all previously encountered elapsed durations are multiples of Δt_u, the next planned duration will be either infinity or a multiple of Δt_u. This is exactly what is expressed in (6). Because the stated condition pertains to all possible simulation runs for model M, the formula includes a quantification over all duration vectors in $\mathscr{T}_M$. Beneath is an implication ($\ldots \Rightarrow \ldots$) with a left-hand side (LHS) and a right-hand side (RHS). The LHS formalizes the assumption that all previous $\langle \Delta t_e \rangle_i$ are multiples of Δt_u, while the RHS captures the requirement that $\langle \Delta t_p \rangle_n$ is infinity or a multiple of Δt_u. Although the RHS explicitly focuses on the last planned duration, it pertains to all planned

durations for the following reason. Recall from Section 2 that one can take a simulation run and omit the final state transition. As in (2), if the original duration vector is in the set of possibilities $\mathscr{T}_M$, then so are the abbreviated variations of this vector. Thus because the expression in (6) involves a quantification over all duration vectors in $\mathscr{T}_M$, all planned durations wind up on the RHS.

Let us now focus on the LHS of the implication in (6), which essentially assumes that elapsed durations are conveniently quantized. The rationale for this assumption relates to remarks made in Section 1: input messages come from a source, and the time constraints associated with the source can be analyzed separately from the receiving model. Since (*a*) we want to defer the analysis of the timing of received messages, and (*b*) the arrival of these messages impacts elapsed durations, we therefore (*c*) make a convenient assumption about elapsed durations. The practical significance of this assumption will become clear when we explain how time units of coupled models and simulation runs are calculated.

We turn our attention to the task of applying (6) to a given model specification. To ease this task, we propose a simple *zeroth-order permissibility test* and a somewhat more complicated *first-order permissibility test* that indicate whether a positive, finite time unit Δt_u is permissible. Both tests are strictly conservative, meaning that any Δt_u which passes either test is definitely permissible. A time unit that fails either test should be considered not permissible only if that test is appropriate for the model under analysis.

The zeroth-order test, shown in (7), is simply the unqualified restriction on the last planned duration. The test is appropriate for any model for which past planned and elapsed durations provide no information that would help to constrain future planned durations. Models in this category tend to exhibit extremely simple behavior with respect to simulated time. Discrete-time simulation models are an obvious example.

$$\left(\langle \Delta t_p \rangle_n = \infty\right) \vee \left(\frac{\langle \Delta t_p \rangle_n}{\Delta t_u} \in \mathbb{N}\right) \tag{7}$$

The first-order test is a proof by mathematical induction based on (6). The proof consists of a base case given by (8) and an inductive step given by (9). The first-order test is appropriate for any model in which the next planned duration $\langle \Delta t_p \rangle_{i+1}$ may be constrained in part by the current planned duration $\langle \Delta t_p \rangle_i$ and the current elapsed duration $\langle \Delta t_e \rangle_i$, but not directly by preceding planned and elapsed durations such as $\langle \Delta t_p \rangle_{i-1}$ and $\langle \Delta t_e \rangle_{i-1}$. A wide range of typical discrete-event simulation models fall into this category.

$$\left(\langle \Delta t_p \rangle_0 = \infty\right) \vee \left(\frac{\langle \Delta t_p \rangle_0}{\Delta t_u} \in \mathbb{N}\right) \tag{8}$$

$$\left(\left(\langle \Delta t_p \rangle_i = \infty\right) \vee \left(\frac{\langle \Delta t_p \rangle_i}{\Delta t_u} \in \mathbb{N}\right)\right) \wedge \left(\frac{\langle \Delta t_e \rangle_i}{\Delta t_u} \in \mathbb{N}\right) \quad \Rightarrow \quad \left(\langle \Delta t_p \rangle_{i+1} = \infty\right) \vee \left(\frac{\langle \Delta t_p \rangle_{i+1}}{\Delta t_u} \in \mathbb{N}\right) \tag{9}$$

We have now established the mathematical theory needed to identify the permissible time units of a given model. It turns out that the longest of these time units is particularly useful because it can be divided by positive integers to yield the other permissible time units. This leads to the concept of an optimal time unit $\langle \Delta \hat{t}_u \rangle_M$ for model M. As defined in (10), this property is zero for models with no permissible time units, and infinity for models where all time units are permissible.

$$\langle \Delta \hat{t}_u \rangle_M = \sup\left(\langle \Delta \mathbb{T}_u \rangle_M \cup \{0\}\right) \tag{10}$$

4 APPLICATION TO NOTABLE CLASSES OF SIMULATION MODELS

Here we apply the theory introduced in the previous section. First, we identify the permissible and optimal time units of the generator, ramp, and processor models as specified in Chapter 4 of *Theory of Modeling and Simulation*. We then expand on the theory in relation to models with infinite optimal time units, those with optimal time units of zero, and coupled models.

4.1 Generator Model

The *generator model* "outputs a 1 in a periodic fashion" (Zeigler, Praehofer, and Kim 2000). The period between successive output messages is simply the time step Δt_s, a model parameter. The output message values have no bearing the model's timing patterns, so we need not regard the value as always equal to 1. In fact the analysis below applies to any model with no time-dependent inputs and a fixed time step between outputs. Such discrete-time simulation models are pervasive in a multitude of domains.

Figure 2 illustrates the progression of a simulation run using the generator model. Since there are no input messages, all events after the first are planned events. As in Figure 1, the planned durations are shown in the order in which they are encountered, though to avoid redundancy we will no longer plot elapsed durations in such diagrams. Note that elapsed durations are always $\langle \Delta t_e \rangle_0 = t_\mathsf{B} - t_\mathsf{A}$, $\langle \Delta t_e \rangle_1 = t_\mathsf{C} - t_\mathsf{B}$, etc.

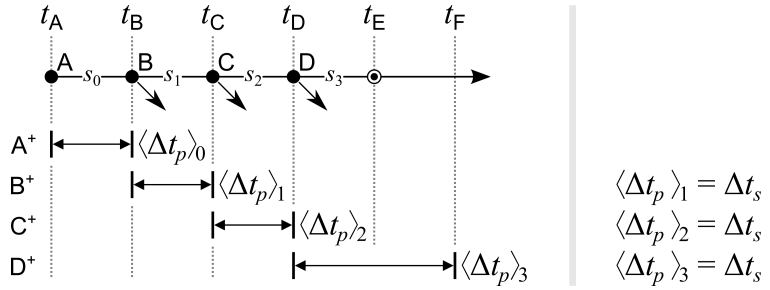

Figure 2: Constraints on planned durations $\langle \Delta t_p \rangle_i$ for the generator model over the following sequence of event types: initialization (A), planned×3 (B–D).

The first step in deriving permissible and optimal time units is to formally express all model-specific constraints on planned and elapsed durations, purposely omitting any representation of state. We start with the constraints on the first planned duration $\langle \Delta t_p \rangle_0$. Inconveniently for the task at hand, DEVS models typically omit the initial state since it has more to do with the desired experiment than the modeled system. Nevertheless, we observe that constraints on the first planned duration tend to reflect those on subsequent durations, at least to some extent. A generator with a time step of 10 seconds is unlikely to begin in such a state that its first output is produced π seconds after the beginning of a simulation run. In fact only two initial planned durations strike us as normal for a generator-like model: zero, implying that the first output coincides with the start of the simulation; and Δt_s, meaning that time advances one step before the first output is produced. Let us assume that either initial duration is possible, as expressed in (11).

$$\langle \Delta t_p \rangle_0 \in \{0, \Delta t_s\} \tag{11}$$

Because the generator never receives messages, the elapsed duration is never cut short. In other words, it is always equal to the corresponding planned duration, as in (12).

$$\langle \Delta t_e \rangle_i = \langle \Delta t_p \rangle_i \tag{12}$$

Finally, all planned durations after the first are necessarily equal to the time step, as in (13). Recall that $i < n$ unless otherwise specified, so $\langle \Delta t_p \rangle_{i+1}$ expresses planned durations $\langle \Delta t_p \rangle_1$ through $\langle \Delta t_p \rangle_n$.

$$\langle \Delta t_p \rangle_{i+1} = \Delta t_s \tag{13}$$

Here the zeroth-order test is appropriate for finding the permissible time units, since no planned durations depend on preceding planned or elapsed durations. We begin with the test expression in (7) and manipulate it using the model-specific constraints (11)–(13) as needed.

$$\begin{aligned} & \left(\langle \Delta t_p \rangle_n = \infty \right) \vee \left(\frac{\langle \Delta t_p \rangle_n}{\Delta t_u} \in \mathbb{N} \right) && \text{zeroth order test (7)} \\ \Rightarrow \quad & \left(\frac{\langle \Delta t_p \rangle_n}{\Delta t_u} \in \mathbb{N} \right) && \langle \Delta t_p \rangle_n < \infty \text{ from (11) and (13)} \end{aligned}$$

$$\begin{aligned}
&\Rightarrow \quad \left(\frac{0}{\Delta t_u} \in \mathbb{N}\right) \wedge \left(\frac{\Delta t_s}{\Delta t_u} \in \mathbb{N}\right) && \langle \Delta t_p \rangle_n \in \{0, \Delta t_s\} \text{ from (11) and (13)} \\
&\Rightarrow \quad \left(\frac{\Delta t_s}{\Delta t_u} \in \mathbb{N}\right) && 0 \in \mathbb{N} \text{ from definition of } \mathbb{N} \\
&\Rightarrow \quad \Delta t_u \in \left\{\Delta t_s, \tfrac{\Delta t_s}{2}, \tfrac{\Delta t_s}{3}, \dots\right\}
\end{aligned}$$

We formally discover that the time step is a permissible time unit of the generator model, and dividing the time step by any positive integer yields another permissible time unit. In this case the optimal time unit is simply the longest permissible time unit, which happens to be the time step itself.

$$\langle \Delta \hat{t}_u \rangle_M = \sup\left\{\Delta t_s, \frac{\Delta t_s}{2}, \frac{\Delta t_s}{3}, \dots, 0\right\} = \Delta t_s \quad \text{from definition of} \langle \Delta \hat{t}_u \rangle_M \text{ in (10)}$$

To summarize, the generator model can be simulated with a uniform discretization of time based on its time step or any other time unit that evenly divides the time step. This result is rather obvious, and it is why such models are referred to as discrete-time simulation models. The reason we formally calculate the result is to show that the proposed theory holds for this simple case. In practice one need only glance at such a model to conclude that its optimal time unit is its time step.

4.2 Ramp Model

The *ramp model* is similar to the generator in that outputs are evenly spaced and separated by the time step Δt_s, but different in that at any time an input message may be received. These inputs do not affect the timing of planned events. They may however influence the trajectory of the ramp's state, which likely alters the output values. The word "ramp" pertains to simple cases where the state trajectory has a linear relationship with elapsed time, though this is of little concern for the purpose of calculating permissible and optimal time units. In general, models with ramp-like timing patterns are useful for integrating discrete-time simulation models with different time steps. A ramp-based simulation scenario is illustrated in Figure 3.

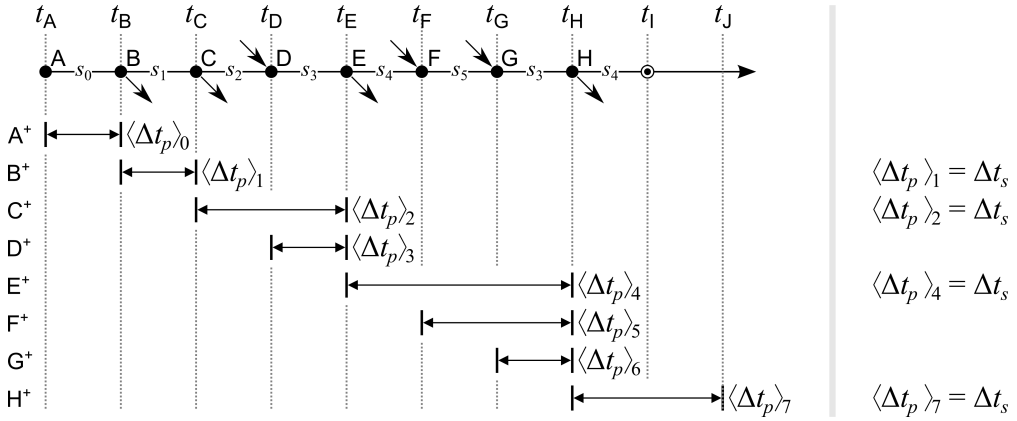

Figure 3: Constraints on planned durations $\langle \Delta t_p \rangle_i$ for the ramp model over the following sequence of event types: initialization (A), planned×2 (B, C), unplanned (D), planned (E), unplanned×2 (F, G), planned (H).

Since a message may theoretically be received at any time, even after π or some other irrational number of seconds, one would expect the ramp model to treat time as fundamentally continuous. In other words, one expects that there are no permissible time units. Yet we will find that the ramp does have permissible time units according to the proposed theory, and that they are the same as those of the generator.

We begin by enumerating the constraints on the planned and elapsed durations. The ramp's initial planned duration is either zero or the time step, similar to that of the generator.

$$\langle \Delta t_p \rangle_0 \in \{0, \Delta t_s\} \tag{14}$$

Recall that for the generator model, $\langle \Delta t_e \rangle_i = \langle \Delta t_p \rangle_i$ holds everywhere. For the ramp model, it is only a possibility that corresponding elapsed and planned durations are equal. If they are equal, it means that either (*a*) an incoming message triggered an unplanned event just when the planned event was scheduled, in which case the planned event is imminent ($\langle \Delta t_p \rangle_{i+1} = 0$), or (*b*) the planned event just occurred, in which case the next planned event is scheduled after a time step ($\langle \Delta t_p \rangle_{i+1} = \Delta t_s$).

$$\langle \Delta t_e \rangle_i = \langle \Delta t_p \rangle_i \quad \Rightarrow \quad \langle \Delta t_p \rangle_{i+1} \in \{0, \Delta t_s\} \tag{15}$$

If an incoming message triggers an unplanned event at an earlier time than that of the planned event, then the elapsed duration $\langle \Delta t_e \rangle_i$ must be subtracted from the previous planned duration $\langle \Delta t_p \rangle_i$ to produce the new planned duration $\langle \Delta t_p \rangle_{i+1}$. In other words, since the current time has advanced by $\langle \Delta t_e \rangle_i$, the remaining time must be reduced by $\langle \Delta t_e \rangle_i$ to maintain the time point of the next output.

$$\langle \Delta t_e \rangle_i < \langle \Delta t_p \rangle_i \quad \Rightarrow \quad \langle \Delta t_p \rangle_{i+1} = \langle \Delta t_p \rangle_i - \langle \Delta t_e \rangle_i \tag{16}$$

Having formulated the constraints in (14)–(16), we now wish to apply one of the permissibility tests. Since the tests are strictly conservative, it does no harm to start with the zeroth-order test. However, in light of the fact that planned durations fall anywhere in the range $0 \leq \langle \Delta t_p \rangle_{i+1} \leq \Delta t_s < \infty$, the zeroth-order test will not provide any permissible time units. The first-order test is more appropriate for the ramp model since (15) and (16) indicate a relationship between planned and elapsed durations $\langle \Delta t_p \rangle_i$ and $\langle \Delta t_e \rangle_i$ and the next planned duration $\langle \Delta t_p \rangle_{i+1}$.

To apply the first-order test, we should start with the base case expression in (8). But let us simply observe that the manipulations on (8) for the ramp model would be almost identical to manipulations on the zeroth-order test expression (7) for the generator model. As demonstrated in Section 4.1, the outcome is that any permissible time unit Δt_u must be the time step Δt_s divided by a positive integer. So we proceed to refine this set of candidate time units by focusing on the inductive step (9). But now we notice that the special case of (15) applied to the RHS of (9) leads to the same manipulations yet again, giving us no additional information. Therefore we move on to the final case, expressed in (16), and we proceed by manipulating the LHS of (9).

$$\begin{aligned}
& \left(\left(\langle \Delta t_p \rangle_i = \infty \right) \vee \left(\frac{\langle \Delta t_p \rangle_i}{\Delta t_u} \in \mathbb{N} \right) \right) \wedge \left(\frac{\langle \Delta t_e \rangle_i}{\Delta t_u} \in \mathbb{N} \right) && \text{LHS of (9)} \\
\Rightarrow \quad & \left(\frac{\langle \Delta t_p \rangle_i}{\Delta t_u} \in \mathbb{N} \right) \wedge \left(\frac{\langle \Delta t_e \rangle_i}{\Delta t_u} \in \mathbb{N} \right) && \langle \Delta t_p \rangle_i \neq \infty \text{ for the ramp model} \\
\Rightarrow \quad & \left(\frac{\langle \Delta t_p \rangle_i}{\Delta t_u} \in \mathbb{N} \right) \wedge \left(\frac{\langle \Delta t_p \rangle_i - \langle \Delta t_p \rangle_{i+1}}{\Delta t_u} \in \mathbb{N} \right) && \text{substitution using (16)} \\
\Rightarrow \quad & \frac{\langle \Delta t_p \rangle_{i+1}}{\Delta t_u} \in \mathbb{N} && \text{property of arithmetic} \\
\Rightarrow \quad & \left(\langle \Delta t_p \rangle_{i+1} = \infty \right) \vee \left(\frac{\langle \Delta t_p \rangle_{i+1}}{\Delta t_u} \in \mathbb{N} \right) && \text{generalization yielding RHS of (9)}
\end{aligned}$$

This calculation shows that the inductive step is always satisfied for the ramp model, meaning that there are no additional constraints on permissible time units. Thus we may now conclude that the permissible time units for the ramp model are identical to those found in Section 4.1 for the generator model, and that the optimal time unit is again the time step Δt_s.

To illustrate the utility of this result, consider a ramp model instance with a 10-second time step. Suppose that this instance receives a message 4 seconds after a planned event. Clearly a 10-second time unit is too coarse, but a 2-second time unit will suffice as it (*a*) evenly divides the elapsed duration of 4 seconds, and (*b*) is a permissible time unit of the ramp model ($\Delta t_s/5$). Now suppose that a message is received $2{\cdot}\pi$ seconds after a planned event. In this case π will not serve as a time unit because, although it evenly

divides the elapsed duration, it is not among the permissible time units. Knowing that the optimal time unit is 10 seconds makes it easy to remember the permissible time units of $10/m$ seconds ($m \in \{1,2,\ldots\}$). The ramp model's permissible time units help one identify appropriate time units for the overall simulation, though timing patterns external to the ramp must be considered as well.

4.3 Processor Model

The *processor model* "outputs the number it received after a processing time" (Zeigler, Praehofer, and Kim 2000). We refer to the "processing time" as the *response duration* Δt_r, a model parameter. The processor is representative of a class of models that respond to input messages after fixed durations, regardless of how output values are determined. Figure 4 illustrates such models under simulation.

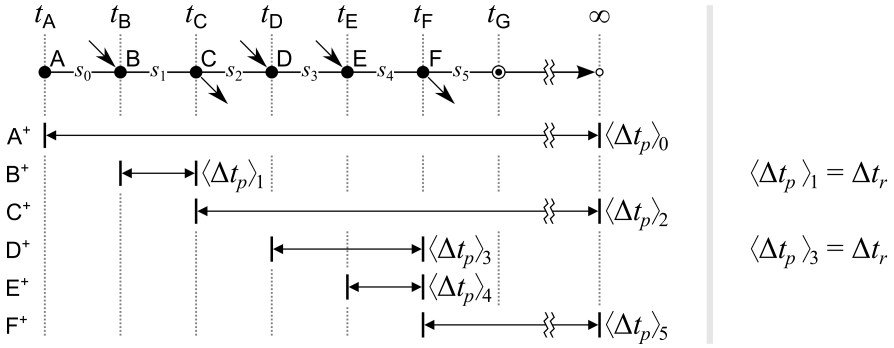

Figure 4: Constraints on planned durations $\langle \Delta t_p \rangle_i$ for the processor model over the following sequence of event types: initialization (A), unplanned (B), planned (C), unplanned×2 (D, E), planned (F).

The processor model differs from the ramp in that, not only can unplanned events occur at any point in simulated time, but planned event times are also not restricted to multiples of any time step. Yet despite this freedom in the timing of events, the optimal time unit of the processor model again happens to be nonzero and finite. We will briefly outline the proof as we formalize the constraints on planned and elapsed durations. As with the ramp model, the first-order test is needed.

We assume that a processor instance is initially waiting for a message, so the first planned duration is infinite as in (17). Because this constraint always satisfies the base case expression of (8), we have yet to identify any restrictions on the permissible time units.

$$\langle \Delta t_p \rangle_0 = \infty \tag{17}$$

If a waiting processor receives a message, then as in (18) its next planned duration is Δt_r. The RHS of the inductive step expression (9) will then restrict permissible time units to $\Delta t_r/m$ for $m \in \{1,2,\ldots\}$.

$$\langle \Delta t_p \rangle_i = \infty \quad \Rightarrow \quad \langle \Delta t_p \rangle_{i+1} = \Delta t_r \tag{18}$$

If a planned event is about to occur, then the next planned duration will be zero or infinity depending on whether a message is received at exactly that instant. Both cases are covered by (19). Since the RHS of (19) satisfies the RHS of (9), there are no additional restrictions on permissible time units.

$$\langle \Delta t_e \rangle_i = \langle \Delta t_p \rangle_i \quad \Rightarrow \quad \langle \Delta t_p \rangle_{i+1} \in \{0, \infty\} \tag{19}$$

If a message is received while a planned event is to later occur, we encounter the familiar $\langle \Delta t_p \rangle_i - \langle \Delta t_e \rangle_i$ on the RHS of (20). This will lead to the same calculation as performed in Section 4.2 for the ramp model, which results in no further restrictions on permissible time units.

$$\langle \Delta t_e \rangle_i < \langle \Delta t_p \rangle_i \leq \Delta t_r \quad \Rightarrow \quad \langle \Delta t_p \rangle_{i+1} = \langle \Delta t_p \rangle_i - \langle \Delta t_e \rangle_i \tag{20}$$

The outcome is that the processor model's optimal time unit is Δt_r, in much the same way that the ramp's optimal time unit turned out to be Δt_s. Dividing by a positive integer yields a permissible time unit.

4.4 Time-Unit-Neutral Models

As a processor model's response duration Δt_r approaches zero, its optimal time unit approaches zero as well since they are equal. If Δt_r actually reaches zero, however, then the optimal time unit becomes infinity. The reason is that the argument outlined in Section 4.3 assumes a nonzero response duration. If $\Delta t_r = 0$, then the RHS of (18) would satisfy the RHS of (9), and consequently any positive real number would be a permissible time unit. It would then follow from (10) that the optimal time unit is infinity.

We refer to a model with an infinite optimal time unit as a *time-unit-neutral model*. When such a model is integrated with another simulation model, it accommodates the other model's timing patterns by imposing no restrictions of its own on common time units. The simplest example is the passive model in *Theory of Modeling and Simulation*, which does nothing ($n = 0$; $\langle \Delta t_p \rangle_0 = \infty$). A more useful example is the *instant processor* described above: a processor with a response duration of zero.

If a model's planned durations are always either zero or infinity, then its optimal time unit is always infinite. But the converse is not necessarily true. It is possible to design a model with an infinite optimal time unit that can produce positive and finite planned durations. Consider a *time-unit-adopting model* that receives messages, stores the durations between messages, and then schedules future events using these stored durations. This is one of the few cases where both the zeroth- and first-order tests are inadequate; only the general expression in (6) would reveal this model's optimal time unit to be infinite.

4.5 Time-Unit-Averse Models

The opposite of a time-unit-neutral model—which has an infinite optimal time unit—is a *time-unit-adverse model*: a model with no permissible time units and an optimal time unit of zero. These are the only models that truly require a continuous representation of time in order to produce theoretically exact event times. One of the simplest examples is a *Poisson process model*, which outputs messages at time points separated by planned durations randomly sampled from an exponential distribution. Probabilistically, these durations are irrational numbers with no common divisors. It follows from (6) that there are no permissible time units, and according to (10) the optimal time unit is zero.

Another noteworthy example of a time-unit-adverse model is the *quantized integrator model* defined in Chapter 16 of *Theory of Modeling and Simulation*. This model is similar to the ramp model in that input messages can be received at any time, and between these messages a scalar property is typically assumed to vary linearly with time. But whereas a ramp constraints the time points of output messages to multiples of a time step, a quantized integrator constrains output values to multiples of a non-temporal quantum. To achieve quantized values, planned durations must be completely unconstrained, and as a consequence there are generally no permissible time units.

4.6 Coupled Models

A *coupled model* contains components which communicate through the instantaneous transfer of messages. Each component is described by a submodel, which allows complex models to be developed in a modular fashion. Given a coupled model M composed of p submodels M_0, M_1, …, M_{p-1}, its set of permissible time units is simply the intersection of the submodel-specific sets, as in (21).

$$\langle \Delta \mathbb{T}_u \rangle_M \;=\; \langle \Delta \mathbb{T}_u \rangle_{M_0} \cap \langle \Delta \mathbb{T}_u \rangle_{M_1} \cap \ldots \cap \langle \Delta \mathbb{T}_u \rangle_{M_{p-1}} \tag{21}$$

A convenient alternative to (21) is to determine a coupled model's optimal time unit using the following set of rules. The first rule is simple: if any of the submodels are time-unit-averse, the coupled model is also time-unit-averse. Otherwise, proceed as follows: first, ignore all time-unit-neutral submodels, as they impose no constraints on time units; second, list the positive, finite optimal time units of the remaining submodels; third, determine the greatest common divisor of all of these time units, as this is the optimal time unit of the coupled model. Recall that a coupled model containing a 10-second generator and a 15-second processor has an optimal time unit of 5 seconds, 5 being the greatest common divisor of 10

and 15. If we include an instant processor (time-unit-neutral) as a third component of the coupled model, the optimal time unit remains unchanged. But if we add a Poisson process model (time-unit-averse) as a fourth component, the coupled model's optimal time unit becomes zero. Note that when coupled models are nested within other coupled models, optimal time units can be calculated by working upwards from the bottom of the hierarchy. Also note that as with any model, one simply divides a coupled model's optimal time unit by any positive integer to obtain a permissible time unit.

For the purpose of choosing a common time unit for a simulation run—a time unit supporting theoretically exact event times—the simulation run can be conceptually represented by a coupled model with its own optimal time unit. One of the components of this "simulation-level coupled model" is necessarily the actual "model" of the system of interest. The other components include "sources" of time-dependent information. These information sources might not be specified as discrete-event models, yet they often feature constraints on planned durations between output messages. Such constraints can be used to determine a source's optimal time unit, which in turn influences the optimal time unit of the encompassing simulation run. To illustrate, suppose the "model" is a 15-second processor. The simulation injects messages at time points 03:51.683, 16:06.277, and 16:13.954. This injection of messages constitutes a "source" with planned durations of 231683 ms, 734594 ms, and 7677 ms, which have no common factors other than 1 ms. And so, although the model has a 15-second optimal time unit, the source has a 1-millisecond optimal time unit for the experiment at hand. Thus for exact results the simulation must be run with a common time unit of 1 ms, or more generally 1 ms divided by any positive integer. Note that the "sources" mentioned here may be regarded as part of an *experimental frame* (Zeigler, Praehofer, and Kim 2000), which could be described as having its own optimal time unit.

5 IMPLICATIONS FOR SIMULTANEOUS EVENTS AND TIME REPRESENTATION

Two implications of the contributed theory deserve attention, the first of which concerns preexisting methods in which simultaneous events are handled in part by the offsetting of event times. Time offsets introduce the possibility that nearly simultaneous events, with close but distinct time points, will occur at the same time or in the opposite order as planned. A question arises: for each method, how large can these offsets be before such reordering effects occur? Having defined optimal time units, we are now in a position to provide answers. We start with the ε-delay scheme (Kim, Seong, Kim, and Park 1997), in which a unique nonnegative multiple of some ε is added to each simultaneous event time to order the events. To avoid reordering nearly simultaneous events, it is suggested that the largest such multiple be less than the minimum of any planned duration. We can now be more accurate: these multiples of ε must be less than the optimal time unit of the simulation run. Another method involves the "threshold of event simultaneity" (Wieland 1999), a parameter δ that bounds randomly generated positive or negative event time offsets. If δ is sufficiently small, only exactly simultaneous events are affected. If δ is larger, the order of nearly simultaneous events may also be randomized, which is claimed to improve robustness in certain contexts. But what value of δ separates the two effects? We can now give the answer as half the optimal time unit. Finally, we consider the time granule (Zeigler, Moon, and Kim 1996), a parameter d that promotes parallelization by equating the event times of nearly simultaneous events. We observe that d must be at least the optimal time unit to have the desired effect.

The second implication pertains to computer representations of simulated time. Discrete-event simulations are typically associated with continuous time, suggesting that binary floating-point numbers be used as a basis for simulated time points and durations. With one or two exceptions such as CD++ (Wainer 2002), most DEVS-based simulators rely on floating-point time values (Vicino, Dalle, and Wainer 2014). Nevertheless, the presented theory lends credibility to the following integer-based time representation: a duration-specific integer multiplier m times a common time unit Δt_u. The rationale is as follows. Both floating-point-based and integer-based time values will introduce error into event times for time-unit-adverse models. Yet a diverse class of discrete-event simulation models feature nonzero optimal time units, and for these models one is better able to produce exact event times using an integer-based representation.

Incidentally, not only do the generator, ramp, and processor models feature positive optimal time units, but so do all ten atomic model examples in Chapter 4 of *Theory of Modeling and Simulation*. In fact all six examples of coupled models also have positive optimal time units, provided duration parameters and initial values are rational numbers. These sixteen models were selected for their educational value, and learning discrete-event simulation is one endeavor for which exact results are highly desirable when possible. Hence while Varga (2014) and others have observed practical advantages of integer-based time values, there is now a complementary argument supported by theory.

6 CONCLUSION

This paper has introduced theory for reasoning about time units as inherent properties of discrete-event simulation models. We have shown that all models have an optimal time unit, that some have permissible time units, and that first- and second-order permissibility tests can be used to formally derive these units. An understanding of the theory leads to an intuition about time units; notably, in the absence of time-unit-adverse models, the optimal time unit is often the greatest common divisor of all duration parameters. The optimal time unit plays an informative yet previously unrecognized role in several preexisting methods for handling simultaneous events. Also, the high prevalence of nonzero optimal time units suggests the use of integer-based time representations that reflect the surprisingly quantized nature of simulated time.

REFERENCES

Kim, K. H., Y. R. Seong, T. G. Kim, and K. H. Park. 1997. "Ordering of simultaneous events in distributed DEVS simulation". *Simulation Practice and Theory* 5 (3): 253–268.

Smolin, L. 2014. "Atoms of space and time". *Scientific American* 23 (4): 94–103.

Varga, A. 2014. *OMNeT++ User Manual (Version 4.5)*.

Vicino, D., O. Dalle, and G. Wainer. 2014. "A data type for discretized time representation in DEVS". In *Proceedings of the International Conference on Simulation Tools and Techniques (SIMUTools)*. Lisbon, Portugal.

Wainer, G. 2002. "CD++: A toolkit to develop DEVS models". *Software: Practice and Experience* 32 (13): 1261–1306.

Wieland, F. 1999. "The threshold of event simultaneity". *Transactions of the Society for Computer Simulation International* 16 (1): 23–31.

Zeigler, B. P. and Y. Moon and D. Kim 1996. "High performance modelling and simulation: Progress and challenges".

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

AUTHOR BIOGRAPHIES

NAMEFIRSTAUTHOR is [...] at [...] [...] [His/her] email address is emailfirstauthor.

NAMELASTAUTHOR is [...] at [...] [...] [His/her] email address is emaillastauthor.