A Taxonomy of Event Time Representations

Conference Paper · April 2017

CITATIONS
0

READS
8

2 authors:

Rhys Goldstein
Autodesk
38 PUBLICATIONS **177** CITATIONS
SEE PROFILE

Azam Khan
Autodesk Research
101 PUBLICATIONS **1,451** CITATIONS
SEE PROFILE

A TAXONOMY OF EVENT TIME REPRESENTATIONS

Rhys Goldstein
Azam Khan

Autodesk Research
210 King Street East, Suite 500
Toronto, ON, Canada
{rhys.goldstein, azam.khan}@autodesk.com

ABSTRACT

Terms such as "simulated time", "simulation time", "virtual time", "logical time", and "real time" appear throughout the modeling and simulation literature as a means of describing the timing, ordering, and/or processing of events. Unfortunately, this vocabulary can become a source of confusion due to subtle inconsistencies in how the terms are interpreted. Here we categorize mathematical representations of event times, review their formal properties such as causal consistency and causal characterization, identify formal relationships among the representations, and present a taxonomy to clarify the proper meanings of the most common notions of time in a simulation context. A thorough look at the various event time representations suggests new research opportunities, such as the repurposing of distributed computing techniques for debugging both parallel and sequential discrete event simulations.

Keywords: time representation, discrete event systems, causality, simulation debugging

1 INTRODUCTION

To successfully learn, design, implement, or communicate state-of-the-art modeling and simulation methods, it is beneficial for researchers and practitioners to have a common and precise understanding of the various notions of time found in the literature. Fujimoto (2000) points out that failure to distinguish among time-related concepts "*is perhaps one of the greatest sources of confusion when beginning to learn about parallel and distributed simulations*", and we believe the same can be said of complex sequential simulations. Confusion can occur when terms such as "simulated time", "simulation time", "virtual time", "logical time", and "real time"—which describe the timing, ordering, and/or processing of events—are used interchangeably or inconsistently. Although these time-related terms have distinct meanings, no single previous reference precisely differentiates all of the prominent mathematical representations of event times.

This paper proposes consistent terminology, definitions, and formal constraints expressing the numerous ways in which event times have been represented over the past 40 years in the fields of distributed computing and simulation research. Specific event time representations including "vector time" and "discrete event time" are organized in a taxonomy featuring categories such as "logical time" and "simulation time". The taxonomy also includes well-established properties such as "causal consistency", and relationships including "event time characterization" that are identified for the first time in this paper.

The scope of the presented taxonomy is narrower than those found in mathematical overviews of temporal logic (Knight and Ma 1993, Hayes 1996), yet broader than a number of simulation-oriented works which

SpringSim-TMS/DEVS, 2017 April 23-26, Virginia Beach, VA, USA;

review time-related concepts in support of a particular representation (Nutaro and Sarjoughian 2003, Lee 2014). The mathematics underlying our taxonomy is largely inspired by the classic paper "*Time, clocks, and the ordering of events in a distributed system*" (Lamport 1978) and subsequent work on logical time in which timestamps are used to capture potential causal relationships. We speculate that many of these classic distributed computing techniques could be repurposed in a modern simulation-oriented context. As an example, we discuss the potential use of logical time for debugging discrete event simulations regardless of whether events are processed in parallel or in sequence.

2 BACKGROUND

A multitude of advances from a number of disciplines have led to an enriched understanding of how time can be represented both mathematically and computationally. We review these ideas and discuss their relevance to our proposed taxonomy.

2.1 Temporal Logic

Temporal logic encompasses theories that aid in reasoning without necessarily quantifying event times. In *A Catalog of Temporal Theories*, Hayes (1996) presents a taxonomy of time-related concepts including "tense", "time interval", and "temporal position". A similar taxonomy by Knight and Ma (1993) emphasizes the distinction between discrete vs. continuous and linear vs. non-linear time representations. Pnueli (1977) was among the first to advocate for the use of temporal logic in computer science, and a subsequent essay by Lamport (1983) explains how temporal logic can aid in reasoning about concurrent programs. Aspects of temporal logic were applied by Maler, Manna, and Pnueli (1991) and Manna and Pnueli (1992) in a context closely related to simulation.

Unlike classification schemes for temporal logic as a whole, our taxonomy focuses specifically on representations that quantify event times. Although the formal relationships we present may possibly be useful for reasoning, our primary intent is to promote consistent use of terminology (Sections 3–5) and encourage new practical tools for applications such as simulation debugging (Section 6).

2.2 Time Granularity

Bettini et al. (1998) define *time granularity* as a mapping from the integers to a set of contiguous, non-overlapping subsets of the time domain. The concept is motivated in part by the need to accommodate computer technology by discretizing time.

Two manifestations of time granularity in a simulation context are the distinct concepts of resolution and precision. *Time resolution* pertains to nonzero time durations that separate consecutive events. Related to this is the challenge of how to integrate models with different levels of resolution (Guo, Hu, and Wang 2012, Santucci, Capocchi, and Zeigler 2016). *Time precision* pertains to the time durations that separate consecutive points at which events are permitted to occur. Goldstein, Breslav, and Khan (2016) show that a theoretically significant precision level can often be formally derived from a model's specification, even if time is treated as a continuous quantity. This finding supports the use of fixed-point over floating-point computer representations of time. More exact digital representations have recently been explored as a possibility for discrete event simulation (Vicino, Dalle, and Wainer 2014, Vicino, Dalle, and Wainer 2016).

The quantities which separate one possible event time from the next are a key aspect of the time representations covered in this paper. However, the above works on time granularity all focus on simulated time (see Section 4.2), which is just one of the many notions of time we review.

2.3 Simultaneous Events

We define *simultaneous events* informally as a set of events for which the order of occurrence is at some point unknown and merits attention. It is a prominent issue faced by nearly all researchers and practitioners of discrete event simulation, and has led to a number of interesting research topics. Wieland (1999) proposes randomly offsetting event times by as much as a duration parameter δ called the "threshold of event simultaneity". Simulations are repeated with different offsets in the hopes of achieving robust statistics. Small changes in event times can also improve performance in parallel and distributed simulation. Zeigler, Moon, and Kim (1996) quantize time according to a granule d, allowing greater numbers of events to be executed concurrently by synchronous simulators such as those based on the Parallel DEVS modeling formalism (Chow and Zeigler 1994, Zeigler, Praehofer, and Kim 2000). Fujimoto (1999) approximates time points in a manner that increases concurrency in asynchronous simulations.

We avoid defining simultaneous events as a set of events with a common timestamp. The question of whether two events have equal timestamps depends not only on the ordering of the events, if there is any, but also on the event time representation used. For example, two events at the same point in simulated time may occur at different points in virtual time (see Section 4.2), and might not be considered simultaneous.

2.4 Potential Causality

In his foundational paper, Lamport (1978) elaborates on the concept of *potential causality*, a one-way relationship in which a path exists for information to flow from one event A to another event B. If such a path exists, we say "A potentially causes B" and apply Lamport's notation $\mathsf{A} \rightarrow \mathsf{B}$. If neither $\mathsf{A} \rightarrow \mathsf{B}$ nor $\mathsf{B} \rightarrow \mathsf{A}$, we say "A and B are causally independent" and write $\mathsf{A} \parallel \mathsf{B}$. It is conventional to focus on *discrete event systems* in which there are a set of communicating logical processes, or *instances*, with their own internal states. An *event* is a self-contained set of computations associated with a single instance, and it is only at such an event that the instance's state may change. A message may be transmitted from a sending event of one instance to a receiving event of another. In this context, $\mathsf{A} \rightarrow \mathsf{B}$ is equivalent to stating that at least one of the following conditions is true:

1. A and B are associated with the same instance, and A directly precedes B.
2. A and B are associated with different instances, and A sends a message to B.
3. There exists an event C such that $\mathsf{A} \rightarrow \mathsf{C}$ and $\mathsf{C} \rightarrow \mathsf{B}$.

A guiding objective of Lamport's paper was to describe a distributed system as a state machine (Lamport 2016). Although the concepts he developed toward this end remain fundamental to discrete event systems, his terminology can be adapted to suit a wider range of objectives. Lamport (1978) used the term "clock" to emphasize the symmetry between two elements he combined: logical timestamps and measurements output by a real-work clock. This symmetry loses its relevance in the broader context of our review, so we drop the term "clock" in favor of "event time representation". Thus "logical clock" becomes "logical time", as in Raynal and Singhal (1996) and other works. Also, Lamport named $\rightarrow$ the "happened before" relation. Although this temporal phrase makes sense for the timestamps he prescribed, it contradicts other event time representations. For example with simulated time (see Section 4.2), it is possible for A and B have a common timestamp even if $\mathsf{A} \rightarrow \mathsf{B}$. Thus we read $\rightarrow$ not as "happened before", but rather "potentially causes".

Potential causality was inspired by early work on *replicated databases*, which attempt to maintain consistent information while serving different regions (Johnson and Thomas 1975). This application continues to draw attention to the concept due to the prevalence of large-scale social networks (Lloyd et al. 2014).

3 PROPERTIES OF EVENT TIME REPRESENTATIONS

Three well-established formal properties help disambiguate specific event time representations. Some refer to these properties as "minimal", "weak", and "strong" consistency (Rönngren and Liljenstam 1999, Nutaro and Sarjoughian 2003), though these terms do not quite align with the similarly named consistency guarantees of replicated databases (Gotsman et al. 2016). With the aim of presenting a taxonomy that spans disciplines, we take inspiration from Schwarz and Mattern (1994) and adopt terms such as "non-contradiction" and "characterization" that attempt to describe the property itself. All three properties constrain the time values t_{A} and t_{B} of events A and B according to their causal relationship.

Causal non-contradiction. If an event time representation exhibits *causal non-contradiction*, we say that the representation "does not contradict" causality, which means that time values cannot oppose any causal relationship in the events they label. The condition below is then satisfied for all pairs of events A and B:

$$\mathsf{A} \rightarrow \mathsf{B} \quad \Rightarrow \quad t_{\mathsf{A}} \leq t_{\mathsf{B}} \tag{1}$$

As the weakest of the three, this property can be described as "minimal consistency" (Rönngren and Liljenstam 1999). However, the term "non-contradiction" better expresses the fact that the causal ordering $\mathsf{A} \rightarrow \mathsf{B}$ cannot be opposed by the timestamp order (i.e. we cannot have $t_{\mathsf{B}} < t_{\mathsf{A}}$). Simulated time (see Section 4.2) is the most widely used event time representation that (*a*) does not contradict causality, and (*b*) fails to satisfy the stronger properties of causal consistency and causal characterization.

Causal consistency. If an event time representation exhibits *causal consistency*, we say it "is consistent with" causality (Schwarz and Mattern 1994), meaning time values are ordered in alignment with any causal relationships among their associated events. Formally, the following condition is always satisfied:

$$\mathsf{A} \rightarrow \mathsf{B} \quad \Rightarrow \quad t_{\mathsf{A}} < t_{\mathsf{B}} \tag{2}$$

This property may be referred to as "weak consistency" (Rönngren and Liljenstam 1999), though this downplays its importance as the key requirement for preserving causal orderings (Lamport 1978). Moreover, "causal consistency" has an analogous meaning in the context of replicated databases (Lloyd et al. 2014). Aside from simulated time, most of the event time representations we review are consistent with causality.

Causal characterization. Any event time representation that exhibits *causal characterization*, or equivalently one that "characterizes" causality (Schwarz and Mattern 1994), allows one to infer any potential causal relationship between events by comparing their timestamps. Causal characterization is formally expressed by a condition similar to (2), except with a bidirectional implication $\Leftrightarrow$:

$$\mathsf{A} \rightarrow \mathsf{B} \quad \Leftrightarrow \quad t_{\mathsf{A}} < t_{\mathsf{B}} \tag{3}$$

Though sometimes referred to as "strong consistency" (Rönngren and Liljenstam 1999), the term "causal characterization" inspired by Schwarz and Mattern (1994) avoids confusion with the strong consistency guarantee of replicated databases (Gotsman et al. 2016), which is similar but not quite analogous. Causal characterization requires a sophisticated event time representation such as vector time (see Section 4.1) that allows both events and their time values to be partially ordered.

4 REVIEW OF EVENT TIME REPRESENTATIONS

An *event time representation* determines how a time point t_{A} may be assigned to an event A to express the event's order relative to other events, and possibly additional information about the timing of real-world causes and effects. As illustrated in Figure 1, the same set of events can be assigned different timestamps depending on which event time representation is used. For example, the 7$^{\text{th}}$ event of Instance 3 has a scalar

time point of 9, a vector time point of $[6,3,7]$, a virtual time point of 15, a simulated time point of t_5, and a discrete event time point of $[t_5,2]$ (formed by pairing the simulated time point with the integer counter at the bottom). A single application may employ any of these representations or a combination of them. The arrows in Figure 1 are messages from sending events to receiving events, as described in Section 2.4.

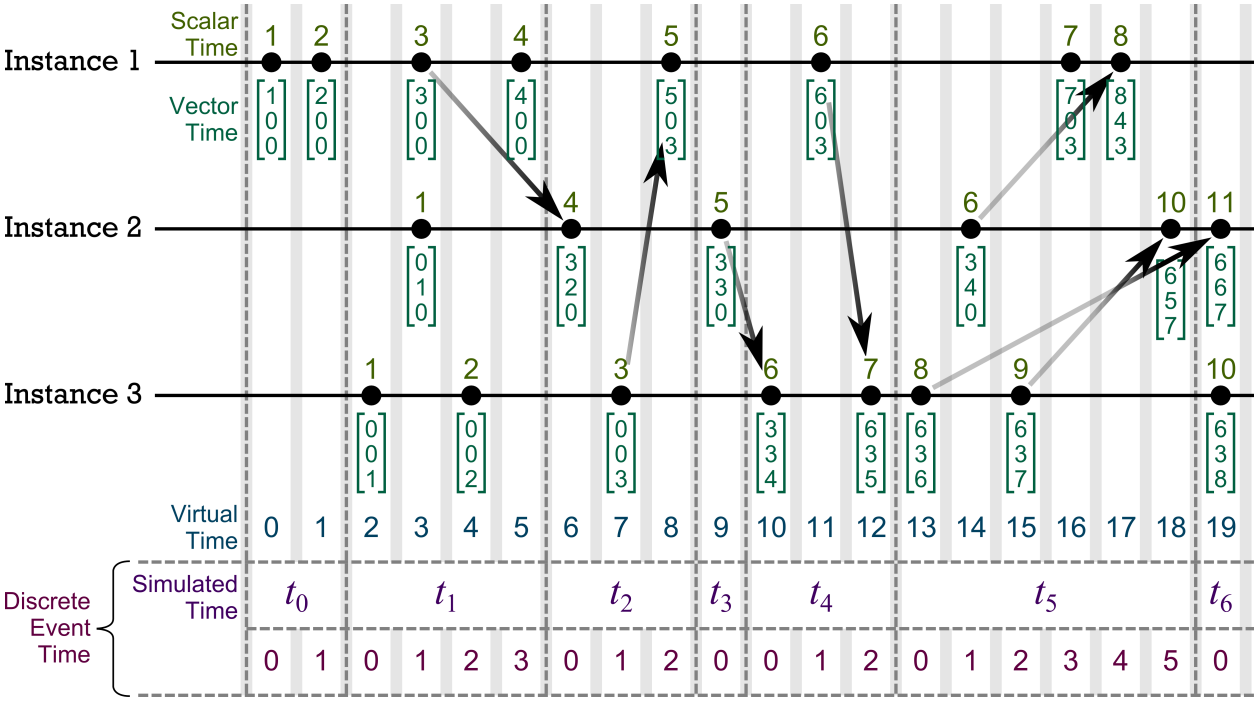

Figure 1: A timeline of events serving as an example of how timestamps are assigned according to various event time representations: scalar time, vector time, virtual time, simulated time, and discrete event time.

4.1 Logical Time

Logical time refers to any event time representation with the primary intent of using timestamps to provide information about the potential causal relationships among events. Any information relevant to physical time, but unrelated to causality, is typically discarded. Raynal and Singhal (1996) identify three types of logical time: scalar time, vector time, and matrix time.

Scalar time. *Scalar time* is the most basic form of logical time, and arguably the simplest of all event time representations consistent with causality. In the absence of messages, scalar time points simply number an instance's events in sequence. If the instance does receive a message, however, then several numbers in the sequence may be skipped to ensure the receiving event is given a greater time point than the sending event where the message originated. In the simplest form of scalar time, each integer-valued event time is the maximum timestamp of all causally preceding events, plus 1. This scheme is illustrated in Figure 1 by the integer timestamps that appear above each event.

Lamport (1978) specifies the conditions on which scalar time is based. The same paper also presents a means of totally ordering all events, though this amounts to a separate representation that could be called *prioritized scalar time*. Although scalar time preserves causal orderings, it discards information about causal independence (Schwarz and Mattern 1994). The events at virtual time points 6 and 16 in Figure 1 are

causally independent and can be processed in either order under asynchronous execution, but this is not evident from their scalar time values of 4 of 7.

Vector time. *Vector time* is the best known event time representation that characterizes causality. Whereas scalar time can help sort causally related events, vector time can also ensure causally independent events remain unordered. This is achieved by representing each event time as a vector with one element per instance. At instance i, the i^{th} element of the vector is simply a count of that instance's events. When a message is sent, the time vector of the sending event is merged into that of the receiving event such that the maximum of each corresponding pair of elements is retained. These rules produce the labels underneath each event in Figure 1.

The key to vector time is how two time points are compared. If every element of t_{A} is at most that of the corresponding element of t_{B}, and at least one is less, then $t_{\mathsf{A}} < t_{\mathsf{B}}$. One can then conclude that $\mathsf{A} \rightarrow \mathsf{B}$. But if at least one corresponding element is larger and at least one is smaller, then t_{A} and t_{B} have no order and one concludes that $\mathsf{A} \parallel \mathsf{B}$. In Figure 1, one can infer that the events at virtual time points 6 and 16 are causally independent by comparing their vector time values $[3,2,0]$ and $[7,0,3]$ and finding no clear order. However the events at virtual time points 6 and 17 are seen as having a potential causal relationship, as the time vector $[8,4,3]$ of the latter event is greater than $[3,2,0]$ in at least one corresponding element, and lesser in none.

Raynal and Singhal (1996) give credit to Fidge (1991), Mattern (1988), and Schmuck (1988) for the invention of vector time. Its weakness is that, when implemented, the potentially large number of vector elements may significantly increase the amount of data required by each message.

Matrix time. *Matrix time* is similar to vector time, but event times are expanded into n-by-n matrices where n is the number of instances. At instance i, the i^{th} row of the matrix contains exactly the elements of a vector time point. Other rows provide additional causal information that may be useful in certain applications. Further details, including the origins of matrix time, are provided by Raynal and Singhal (1996).

4.2 Simulation Time

We interpret *simulation time* as any event time representation intended as a basis for simulation. Unfortunately, the term is ambiguous in many cases, especially when used to refer to the current point in time during a simulation run. Although "simulation time" and "simulated time" are often used interchangeably, we hope to discourage this practice. In our view, simulated time is a particular type of simulation time. Other types include virtual time and discrete event time. The more specific terms should be used where possible.

Simulated time. *Simulated time* provides a quantitative representation of physical time. Similar to physical time, it is typically expressed in years, days, hours, minutes, seconds, fractions of seconds, or a combination of these physical time units. The terms *continuous time* and *discrete time* both refer to simulated time, with the latter indicating that there is a uniform duration or *time step* separating consecutive event times. Simulated time does not contradict causality. In other words, no event can be influenced by an event at a later point in simulated time. However, simulated time is not necessarily consistent with causality, since multiple causally related events may share the same time point. In Figure 1, the events at virtual time points 2 and 4 are causally related yet share simulated time point t_1. The events at virtual time points 7 and 8 are also causally related, yet both occur at t_2. Lacking causal consistency, simulated time on its own is often inadequate for describing the progress of a discrete event simulation.

Virtual time. *Virtual time* is an event time representation used to provide an ordering of events consistent with causality while also reflecting some other measure of temporal progress. The term "virtual time" was established by Jefferson (1985) in the context of optimistic distributed algorithms, where a robust measure of progress is crucial for guiding computations in which events can be rolled back based on newly received

information. Simulated time could itself be regarded as virtual time in specific cases where no two events with the same simulated time point have any causal relationship. In general, however, a distinct representation is needed to ensure causal consistency, which is an explicit requirement of Jefferson (1985). In its simplest form, virtual time points are integers extracted from a counter that simply tracks the number of past steps in a simulation. The precise points at which the counter is incremented often depends on the simulator's implementation. A simulator may or may not allow multiple events to share a common virtual time point, depending on implementation details, the underlying modeling paradigm, and whether or not there is an intent to support synchronous parallel execution.

Discrete event time. We refer to *discrete event time* as an event time representation that carries at least as much information as simulated time while exhibiting causal consistency. Discrete event time encompasses *superdense time* as elaborated by Lee (2014), as well as the representation proposed by Nutaro (2011) and the use of hyperreal numbers as described by Barros (2016). These works promote essentially the same idea—the juxtaposition of a simulated time variable t with an integer—but formulate and apply the representation slightly differently. We propose the umbrella term "discrete event time" to avoid choosing one formulation over the others, while at the same time emphasizing the well-established utility of this type of representation for obtaining highly versatile timestamps for discrete event systems.

We submit the following simple formulation of discrete event time. Every event time is expressed as a 2-element vector $[t,c]$ where t is simulated time and c is an integer counter. When a new event occurs after a duration of simulated time Δt, the current time $[t,c]$ is advanced to $[t,c] \triangleright \Delta t$ according to the formula below:

$$[t,c] \triangleright \Delta t = \begin{cases} [t+\Delta t, 0], & \Delta t > 0 \\ [t, c+1], & \Delta t = 0 \end{cases} \tag{4}$$

If simulated time t increases, the counter c is reset. If t remains the same, c is incremented. These rules produce sequences of discrete event time points similar to those formed by combining the bottom two rows of Figure 1. To compare time points $[t_\mathsf{A}, c_\mathsf{A}]$ and $[t_\mathsf{B}, c_\mathsf{B}]$, for events A and B, one uses the formula below:

$$\begin{aligned} [t_\mathsf{A}, c_\mathsf{A}] = [t_\mathsf{B}, c_\mathsf{B}] \quad &\Leftrightarrow \quad t_\mathsf{A} = t_\mathsf{B} \ \wedge \ c_\mathsf{A} = c_\mathsf{B} \\ [t_\mathsf{A}, c_\mathsf{A}] < [t_\mathsf{B}, c_\mathsf{B}] \quad &\Leftrightarrow \quad t_\mathsf{A} < t_\mathsf{B} \ \vee \ (t_\mathsf{A} = t_\mathsf{B} \ \wedge \ c_\mathsf{A} < c_\mathsf{B}) \end{aligned} \tag{5}$$

Discrete event time should not be confused with a category of representations we call *prioritized time*: the juxtaposition of any time point with a priority number (usually the instance ID). Prioritized time is often used to achieve a total ordering of events (Johnson and Thomas 1975, Lamport 1978, Kim et al. 1997).

4.3 Wallclock Time

Wallclock time captures the points or intervals in physical time at which events are processed. It can be considered a form of *measured time*, meaning that its values are derived from physical time measurements. Note that *physical time* is not a representation, but rather "time" itself as experienced in day-to-day life. The related yet distinct concept of *real time* is also not a representation, but rather a condition that indicates a loose correspondence between durations of wallclock time and durations of simulated time. Similarly, *scaled real time* is a condition indicating an approximate proportional relationship between durations of wallclock and simulated time (Fujimoto 2000).

Wallclock time is almost always consistent with causality, with the caveat that optimistic algorithms allow rollbacks to earlier states (Fujimoto 1990). Since events take time to process, we provide an alternative to (2) that expresses causal consistency while allowing event times to be represented as intervals. In the formula below, T_A and T_B represent the sets of wallclock time points over which events A and B are processed.

$$\mathsf{A} \rightarrow \mathsf{B} \quad \Rightarrow \quad \forall t_\mathsf{A} \in T_\mathsf{A}, \, \forall t_\mathsf{B} \in T_\mathsf{B}, \, t_\mathsf{A} < t_\mathsf{B} \tag{6}$$

5 RELATIONSHIPS AMONG EVENT TIME REPRESENTATIONS

The event time representations defined in Section 4, their properties, and their relationships are organized in the proposed taxonomy shown in Figure 2. The three gray boxes indicate that measured time, simulation time, and logical time are categories encompassing more specific representations. The arrows of different styles pointing toward "Causality" reflect the formal properties defined in Section 3. The arrows connecting pairs of event time representations indicate analogous relationships defined in this section.

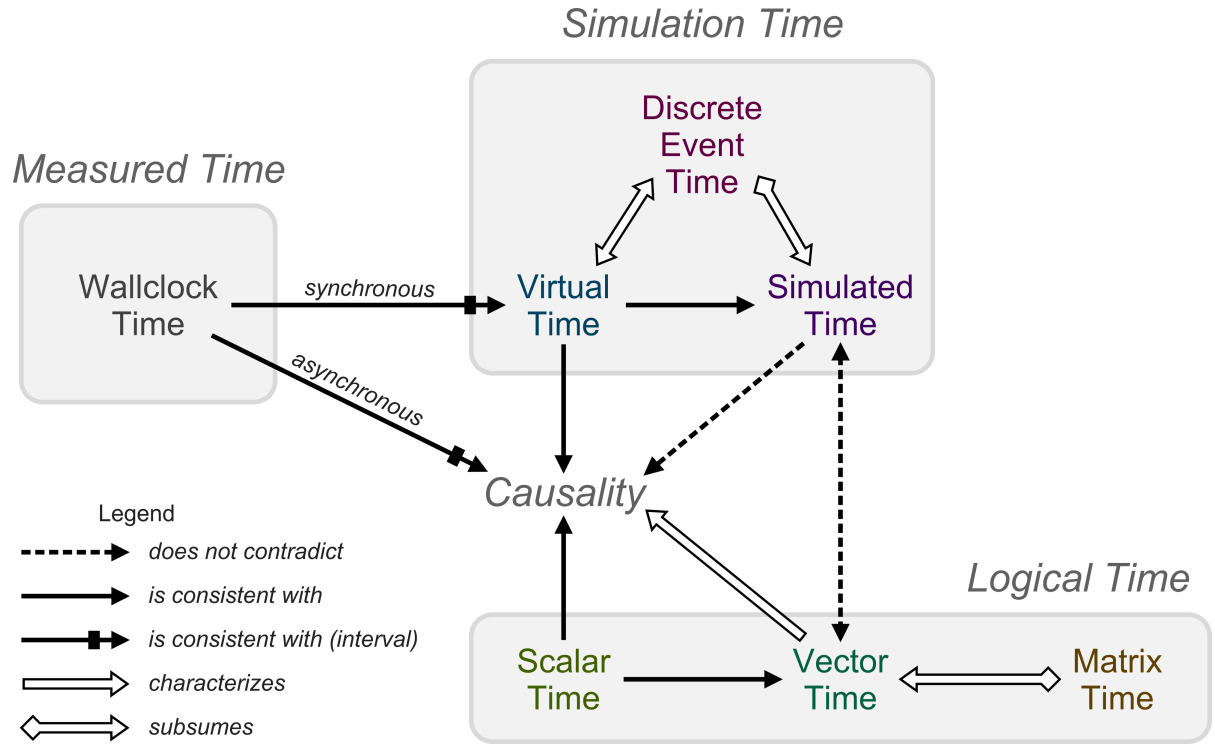

Figure 2: Taxonomy indicating the properties and relationships of event time representations.

The properties and relationships in Figure 2 are interpreted by concatenating the name preceding an arrow, the corresponding phrase in the legend, and the name following the arrow. For example, an arrow near the center of the diagram yields the statement "Simulated Time does not contradict Causality". One then references the formal definition, in this case the property given by (1) in Section 3. Another example is the relationship "Scalar Time is consistent with Vector Time", defined in (8) below. Bidirectional arrows are read in both directions; thus "Discrete Event Time characterizes Virtual Time" as in (10), and vice versa.

Event time non-contradiction. If two event time representations $\mathscr{T}$ and $\mathscr{T}'$ feature *event time non-contradiction*, we say that $\mathscr{T}$ and $\mathscr{T}'$ "do not contradict" one another, which specifies that corresponding time values cannot express strictly opposing orderings of events. Formally, it means that the following relationship is satisfied for any two events A and B with associated timestamps $\mathtt{t}_\mathsf{A}$, $\mathtt{t}_\mathsf{B}$ (according to $\mathscr{T}$) and $\mathtt{t}_\mathsf{A}{}'$, $\mathtt{t}_\mathsf{B}{}'$ (according to $\mathscr{T}'$):

$$t_\mathsf{A} < t_\mathsf{B} \quad \Rightarrow \quad t_\mathsf{A}{}' \leq t_\mathsf{B}{}' \tag{7}$$

The relationship is symmetric. If $\mathscr{T}'$ does not contradict $\mathscr{T}$, then $\mathscr{T}$ does not contradict $\mathscr{T}'$. This follows directly from (7) as demonstrated below. Although the proof swaps A and B, the relationship holds for all

all pairs of events in either order.

$$\begin{array}{lrcll} & t_A < t_B & \Rightarrow & t_A' \leq t_B' & \text{(from (7))} \\ \equiv & \neg(t_A < t_B) & \vee & t_A' \leq t_B' & \text{(equivalent expression for implication)} \\ \equiv & t_B \leq t_A & \vee & \neg(t_B' < t_A') & \text{(logical negation of comparisons)} \\ \equiv & t_B \leq t_A & \Leftarrow & t_B' < t_A' & \text{(equivalent expression for implication)} \end{array}$$

As shown in Figure 2, vector time and simulated time do not contradict one another. This relationship can be derived from the fact that simulated time does not contradict causality ($A \rightarrow B \Rightarrow t_A^{\text{simulated}} \leq t_B^{\text{simulated}}$) and vector time characterizes causality ($A \rightarrow B \Leftrightarrow t_A^{\text{vector}} < t_B^{\text{vector}}$). Substituting the expression for $A \rightarrow B$ from the second property into the first property yields $t_A^{\text{vector}} < t_B^{\text{vector}} \Rightarrow t_A^{\text{simulated}} \leq t_B^{\text{simulated}}$, which has the same form as (7). A number of properties and relationships are omitted from Figure 2 yet can be derived in this manner. For example, virtual time is consistent with vector time (see the definition of event time consistency below), discrete event time is consistent with causality, and wallclock time is consistent with discrete event time under synchronous execution. In essence, the noncontradiction, consistency, and characterization properties/relationships are inherited through characterization.

Event time consistency. The one-way relationship of *event time consistency* states that if one event time representation yields increasing timestamps, the other must as well. If event time representation $\mathscr{T}'$ is consistent with $\mathscr{T}$, then the relationship below holds for all pairs of events and their associated time points:

$$t_A < t_B \quad \Rightarrow \quad t_A' < t_B' \tag{8}$$

Notably, virtual time is consistent with simulated time. We consider this relationship one of the key differentiators between virtual time as discussed by Jefferson (1985) and scalar logical time as introduced by Lamport (1978). Scalar time is not consistent with simulated time, but it is consistent with vector time.

Wallclock time is consistent with virtual time under synchronous execution. If events are processed asynchronously, wallclock time is only consistent with causality. Because events are best described by intervals of wallclock time, we give an alternative formulation of (8) in which T_A, T_B, T_A', and T_B' refer to the sets of time points during which events A and B are processed. If $\mathscr{T}'$ is consistent with $\mathscr{T}$, the following holds:

$$\forall t_A \in T_A,\ \forall t_B \in T_B,\ t_A < t_B \quad \Rightarrow \quad \forall t_A' \in T_A',\ \forall t_B' \in T_B',\ t_A' < t_B' \tag{9}$$

If either $\mathscr{T}$ or $\mathscr{T}'$ is consistent with the other, it follows that $\mathscr{T}$ and $\mathscr{T}'$ do not contradict one another. It is therefore implicit in Figure 2 that scalar time and vector time do not contradict each other, and neither do any two of the three forms of simulation time.

Event time characterization. The symmetric relationship of *event time characterization* means that the ordering of events is preserved if one event time representation is substituted for the other. If the following condition holds for all pairs of events A and B and their associated timestamps according to representations $\mathscr{T}$ and $\mathscr{T}'$, we say that $\mathscr{T}$ and $\mathscr{T}'$ "characterize" one another:

$$t_A < t_B \quad \Leftrightarrow \quad t_A' < t_B' \tag{10}$$

Virtual time and discrete event time characterize one another, even though one cannot necessarily derive a timestamp under one representation from the corresponding timestamp of the other.

Event time subsumption. If two event time representations exhibit the one-way relationship of *event time subsumption*, one can infer an event's exact timestamp according to one representation from the timestamp of the other. Specifically, if $\mathscr{T}'$ "subsumes" $\mathscr{T}$, then as in (11) there exists a function f that derives t_A from t_A' for any event A:

$$\exists f,\ \forall A,\ f(t_A') = t_A \tag{11}$$

As in Figure 2, matrix time subsumes vector time and discrete event time subsumes simulated time.

6 RESEARCH OPPORTUNITIES & CONCLUSION

Based on classic distributed computing concepts, the proposed taxonomy may aid in the development of simulation methods involving both parallel and sequential processing. For example, it may help one identify the event time representation best suited to his/her application, or it may inspire new representations.

To demonstrate the potential for new event time representations, we propose a form of logical time called *genealogical time* that labels causal ancestors and descendants of a focal event using timestamps consistent with causality. The focal event E_F is given a timestamp of 0. Every descendant event E_D for which $E_F \rightarrow E_D$ is assigned a time value 1 greater than the maximum of its causal predecessors, similar to scalar time. Unlike other forms of logical time, every ancestor event E_A for which $E_A \rightarrow E_F$ is assigned a time value 1 less than the minimum of its causal successors. Genealogical time serves as the time axis for the diagram in Figure 3. The timeline is the same as in Figure 1, except only events causally related to the focal event are shown.

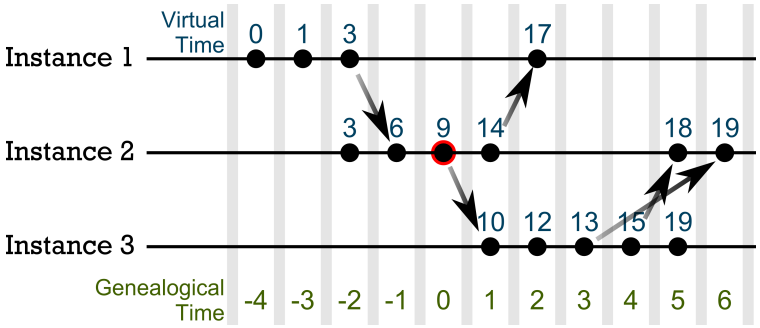

Figure 3: The timeline of Figure 1 re-organized using genealogical time with a focal event at virtual time 9.

Potential applications of genealogical time include the debugging or analysis of simulation models. If an anomaly is detected in the output of a simulation run, the associated event could be selected as a focal event. A visual interface similar to the illustration in Figure 3 might help a modeler systematically observe the likely causes of the anomaly on the left-hand side, and explore its potential effects on the right-hand side. Such analyses could be performed regardless of whether the simulation is executed sequentially or using parallelism. The application of logical time to modern debugging tools has received considerable attention recently in the field of distributed computing (Isaacs et al. 2014, Beschastnikh et al. 2016). New parallel and sequential discrete event simulation tools, including debugging interfaces (Maleki et al. 2015, Van Mierlo et al. 2017), may also benefit from the repurposing of classic distributed computing techniques.

Simulated time, simulation time, virtual time, logical (or scalar/vector/matrix) time, discrete event (or superdense) time, and wallclock time are distinct event time representations defined in this paper and disambiguated by formal properties and relationships indicated in a newly proposed taxonomy (Figure 2). We hope this organizational effort will help researchers and practitioners discuss the various time representations—and related concepts such as real time and physical time—with greater precision and consistency.

REFERENCES

Barros, F. 2016. “Semantics of multisampling systems”. *International Journal of Simulation and Process Modelling* vol. 11 (5), pp. 374–389.

Beschastnikh, I., P. Wang, Y. Brun, and M. D. Ernst. 2016. “Debugging distributed systems”. *Communications of the ACM* vol. 59 (8), pp. 32–37.

Bettini, C., C. E. Dyreson, W. S. Evans, R. T. Snodgrass, and X. S. Wang. 1998. “A glossary of time granularity concepts”. In *Temporal Databases: Research and Practice*. Springer Berlin Heidelberg.

Chow, A. C. H., and B. P. Zeigler. 1994. "Parallel DEVS: a parallel, hierarchical, modular modeling formalism". In *Proceedings of the Winter Simulation Conference (WSC)*.

Fidge, C. 1991. "Logical time in distributed computing systems". *Computer* vol. 24 (8), pp. 28–33.

Fujimoto, R. M. 1990. "Parallel discrete event simulation". *Communications of the ACM* vol. 33 (10), pp. 30–53.

Fujimoto, R. M. 1999. "Exploiting temporal uncertainty in parallel and distributed simulations". In *Proceedings of the Parallel and Distributed Simulation Workshop (PADS)*.

Fujimoto, R. M. 2000. *Parallel and Distributed Simulation Systems*. New York, NY, USA, John Wiley & Sons.

Goldstein, R., S. Breslav, and A. Khan. 2016. "A quantum of continuous simulated time". In *Proceedings of the Symposium on Theory of Modeling & Simulation (TMS/DEVS)*.

Gotsman, A., H. Yang, C. Ferreira, M. Najafzadeh, and M. Shapiro. 2016. "'Cause I'm strong enough: Reasoning about consistency choices in distributed systems". In *Proceedings of the ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*.

Guo, S., X. Hu, and X. Wang. 2012. "On time granularity and event granularity in simulation service composition (WIP)". In *Proceedings of the Symposium on Theory of Modeling & Simulation (TMS/DEVS)*.

Hayes, P. 1996. "A catalog of time theories". Technical Report UIUC-BI-AI-96-01, University of Illinois.

Isaacs, K. E., P.-T. Bremer, I. Jusufi, T. Gamblin, A. Bhatele, M. Schulz, and B. Hamann. 2014. "Combing the communication hairball: Visualizing parallel execution traces using logical time". In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS)*.

Jefferson, D. R. 1985. "Virtual time". *ACM Transactions on Programming Languages and Systems (TOPLAS)* vol. 7 (3), pp. 404–425.

Johnson, P. R., and R. H. Thomas. 1975. "The maintenance of duplicate databases". Network Working Group RFC #677.

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* vol. 5 (3), pp. 253–268.

Knight, B., and J. Ma. 1993. "Time representation: A taxonomy of temporal models". *Artificial Intelligence Review* vol. 7 (6), pp. 401–419.

Lamport, L. 1978. "Time, clocks, and the ordering of events in a distributed system". *Communications of the ACM* vol. 21 (7), pp. 558–565.

Lamport, L. 1983. "What good is temporal logic?". *Information Processing* vol. 83, pp. 657–668.

Lamport, L. 2016. "My writings". http://research.microsoft.com/en-us/um/people/lamport/pubs/pubs.html (accessed 2016 Oct 21).

Lee, E. A. 2014. "Constructive models of discrete and continuous physical phenomena". *IEEE Access* vol. 2, pp. 797–821.

Lloyd, W., M. J. Freedman, M. Kaminsky, and D. G. Andersen. 2014. "Don't settle for eventual consistency". *Communications of the ACM* vol. 57 (5), pp. 61–68.

Maleki, M., R. Woodbury, R. Goldstein, S. Breslav, and A. Khan. 2015. "Designing DEVS visual interfaces for end-user programmers". *Simulation: Transactions of the Society for Modeling and Simulation International* vol. 91 (8), pp. 715–734.

Maler, O., Z. Manna, and A. Pnueli. 1991. "From timed to hybrid systems". In *Proceedings of the Real-Time: Theory in Practice, REX Workshop*.

Manna, Z., and A. Pnueli. 1992. *The Temporal Logic of Reactive and Concurrent Systems*. New York, NY, USA, Springer-Verlag.

Mattern, F. 1988. "Virtual time and global states of distributed systems". In *Proceedings of International Workshop on Parallel and Distributed Algorithms*.

Nutaro, J. J. 2011. *Building Software for Simulation: Theory and Algorithms with Applications in C++*. Hoboken, NJ, USA, John Wiley & Sons.

Nutaro, J. J., and H. H. Sarjoughian. 2003. "A unified view of time and causality and its application to distributed simulation". In *Proceedings of the Summer Computer Simulation Conference (SCSC)*.

Pnueli, A. 1977. "The temporal logic of programs". In *Proceedings of the Annual Symposium on Foundations of Computer Science (SFCS)*.

Raynal, M., and M. Singhal. 1996. "Logical time: Capturing causality in distributed systems". *Computer* vol. 29 (2), pp. 49–56.

Rönngren, R., and M. Liljenstam. 1999. "On event ordering in parallel discrete event simulation". In *Proceedings of the Workshop on Parallel and Distributed Simulation (PADS)*.

Santucci, J.-F., L. Capocchi, and B. P. Zeigler. 2016. "System entity structure extension to integrate abstraction hierarchies and time granularity into DEVS modeling and simulation". *Simulation: Transactions of the Society for Modeling and Simulation International* vol. 92 (8), pp. 747–769.

Schmuck, F. B. 1988. *The Use of Efficient Broadcast Protocols in Asynchronous Distributed Systems*. Ph. D. thesis, Department of Computer Science, Cornell University.

Schwarz, R., and F. Mattern. 1994. "Detecting causal relationships in distributed computations: In search of the holy grail". *Distributed Computing* vol. 7 (3), pp. 149–174.

Van Mierlo, S., Y. Van Tendeloo, and H. Vangheluwe. 2017. "Debugging Parallel DEVS". *Simulation: Transactions of the Society for Modeling and Simulation International* vol. 93 (4), pp. 285–306.

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)*.

Vicino, D., O. Dalle, and G. Wainer. 2016. "An advanced data type with irrational numbers to implement time in DEVS". In *Proceedings of the Symposium on Theory of Modeling & Simulation (TMS/DEVS)*.

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

Zeigler, B. P., Y. Moon, and D. Kim. 1996. "High performance modelling and simulation: Progress and challenges". Technical report, University of Arizona.

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

RHYS GOLDSTEIN is a simulation expert in the Complex Systems group at Autodesk Research, where he investigates tools and techniques for developing discrete-event simulation models and applying them to building architecture and other design challenges. His email address is rhys.goldstein@autodesk.com.

AZAM KHAN is Director, Complex Systems Research at Autodesk, and Founder of the Parametric Human Project Consortium, the Symposium on Simulation for Architecture and Urban Design (SimAUD), and the CHI Sustainability Community. Azam has published numerous articles on simulation, human-computer interaction, architectural design, and sustainability. His email address is azam.khan@autodesk.com.