Event-oriented approaches to geographic phenomena

Michael Worboys
National Center for Geographic Information and Analysis
University of Maine, Orono ME 04469, USA
worboys@spatial.maine.edu

Abstract

This paper is about the information-theoretic foundations upon which useful explanatory and predictive models of dynamic geographic phenomena can be based. It traces the development over the last decade or so of these foundations, from sequences of temporal snapshots, through object life histories, to event chronicles. A crucial ontological distinction is drawn between "things" and "happenings", that is between continuant and occurrent entities. Most of the work up to now has focused on representing the evolution through time of geographic things, whether objects or fields. This paper argues that happenings should be upgraded to an equal status with things in dynamic geographic representations, and suggests ways of doing this. The main research focus of the paper is the application of an algebraic approach, previously developed mainly in the context of computational processes, to real-world happenings. It develops a pure process theory of space and time, and demonstrates its applicability by providing an example of the representation of motion of a vehicle through a region. The paper concludes by noting some of the requirements for scaling this approach to real-world dynamic scenarios, such as might be found, for example, in the automation of coordination of disaster relief.

Keywords: spatiotemporal, event, process, algebra, logic

1 Introduction

The title of this paper makes reference to two previous papers of the author and colleagues. In [36], the object-oriented approach was introduced and applied to spatial data modeling. It has since turned out that seeing the world as a collection of classified objects, with properties, relationships to each other, and definable behavior, is an extremely useful approach to modeling. The theme was continued in [34], where fundamental aspects of the object-oriented paradigm, including identity, classification, inheritance, composition, encapsulation, and operation polymorphism, were introduced. The step forward that the object-oriented paradigm allows us to make is to model our observations of the world, not just as collections of data, but as forming into complex entities, with identity, internal structure and behavior, and capable of relating to other entities. Of course, not every geographic phenomenon can usefully be viewed as a collection of objects. The object-field dichotomy, discussed by Couclelis [5], recognizes the importance of two different kinds of entities: *fields* of variation of properties over a spatial framework (digital elevation models provide the obvious example, where land elevation is the property that varies) and collections of *objects*, relevant, identifiable entities with spatial and non-spatial attributes.

Both objects and fields, at least as conceived above, are static. However, there is a growing body of work showing that in many application domains, a treatment of the dynamic aspects of geographic phenomena is essential for useful explanatory and predictive models. This work goes back at least as far as Hägerstrand [9], emphasizing the importance of time in human activity, and currently exemplified by the work of Miller [20] on transportation and urban analysis, and Yuan [37] on analysis of physical phenomena, such as storms.

This observation leads to the idea of extending the object/field models to allow a temporal variation. So, we can imagine spatiotemporal fields and objects with additional temporal attributes. Spatiotemporal information systems provide the computational embodiment of such conceptions. However, this paper argues that these constructions form merely a half-way house, and that the next real breakthrough in computer modeling of geographic phenomena comes when we move from an object-oriented to an event-oriented view of the world. This view is of course over-simplified, and the details of the argument will show that both temporally indexed snapshots of the world, as well as an event-oriented view, are required for a complete representation.

Our goal in providing approaches to representation and reasoning is to

Date	Start Time	End Time	Place	Description
5 Apr	0700	0720	Home	Get up
5 Apr	0730	0800	Home	Breakfast
5 Apr	0800	0830	Route from home to Department	Walk to work
5 Apr	0845	1000	Office	Work on paper
5 Apr	1000	1100	Graduate seminar room	Class
5 Apr	1100	1130	Office	Meet colleague
5 Apr	1200	1300	Student Union	Lunch with students

Table 1: Relational view of a morning's activities

allow us to explain, make predictions and make planning decisions based on information we have about the world. The argument presented in this paper is that to more effectively perform these function, we need representations, query languages, and techniques for reasoning, where the event-oriented view is explicitly catered for. Other issues, such as event visualization and event-based natural language interfaces are also required, but are not covered in this work.

Consider the following simple scenario. "John got up earlier than usual, had breakfast, walked to the department, worked on a new draft of a paper, took a graduate class, met with a colleague, had lunch with two students, ..." This is a natural and simple description of part of John's day. If set the task of keeping such a diary in a database, we might set up a relation, as shown in Table 1, with columns for date, start time, end time, place, and description of activity. On the face of it, this looks like a perfectly normal table in a relational database, with spatial and temporal references. We traditionally think of a row in a relational database, or an object in an object database, as representing a state of an entity, given by values for its set of attributes, with possible spatial and temporal reference. But notice that Table 1 is concerned with descriptions of occurrences rather than states, and even though structurally similar to a table in a traditional relational database, semantically it is very different. Each row represents the occurrence of an event, specified by its location in space-time and given a description.

This paper describes the concepts underlying a move to incorporate events modeling into our conceptual modeling toolbox. We begin be charting the recent history of dynamic geographic information models.

2 Stages in the development of spatiotemporal information systems

This "brief history of time" (with apologies to Steven Hawking [11]) provides an account of the principal stages in the introduction of temporal capability into geographic information systems.

2.1 Stage Zero: Static GIS

Stage zero is, by and large, where we are now with current proprietary technology. Most systems allow only representation of a single state of knowledge about the application domain. It is usually the case that the state of most interest is that which is as close as possible to the current state, with database updates keeping the state as current as possible. It is possible in stage zero technology to represent the past or future, but only a single moment in time can be represented, and no comparisons between the state of affairs at different times are possible.

2.2 Stage One: Temporal snapshots

The most common approach to spatiotemporal models up to now has been the view of the world as a succession of temporal snapshots of spatial configurations of objects. A *temporal snapshot* is a representation of the state of affairs in a particular domain at a single moment in time. A temporal sequence of snapshots is a collection of temporal snapshots, usually all of the same spatial region, indexed by a temporal variable. One can think of the snapshots as sampling the dynamic phenomena at a sequence of temporal instants. Figure 1 shows the development during the 20th century of part of the region around the University of Maine. (These figures are taken from USGS historical maps, collected as part of a project, headed by historian Christopher Marshall, and hosted on Maptech's web site [18].) The temporal sequence consists of three temporal snapshots, referenced to the years, 1902, 1946 and 1955. It is clear that as time passed many changes have occurred, such as the construction of the airport. This example also shows clearly the importance of untangling changes to the real-world and changes to the database (in this case shown by different cartographic presentation styles for each map).

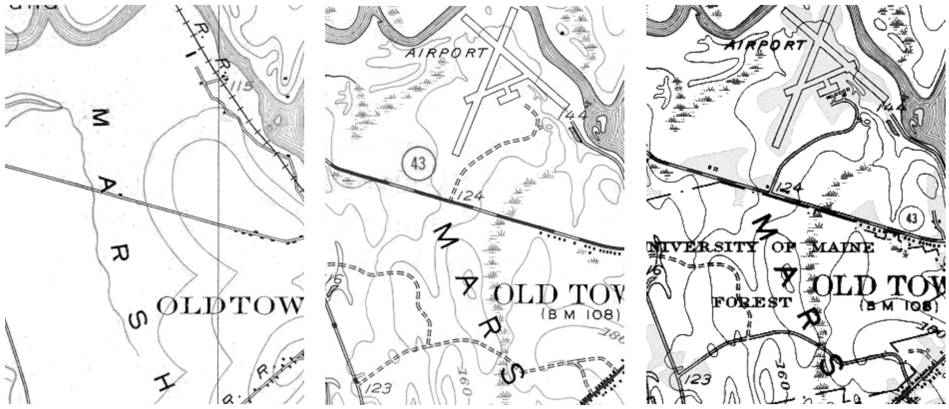

Figure 1: History of part of Old Town, Maine, recorded in snapshot at times 1902 (left), 1946 (center) and 1955 (right)

Stage One snapshot sequences are indexed by a temporal variable, and so the nature and structure of the underlying temporal reference domain influences the structure of the snapshot model. Questions of temporal structure that arise will depend on the application domain, but include whether time is discrete or dense; linear, branching or cyclic; and whether metric and topological properties are relevant. In fact, it is not really the time domain that dictates these properties, but the nature of the geo-phenomena under consideration. If the event to be modeled is continuous (e.g. the movement of a glacier), then the time domain should allow interpolation between measurements. If the event is discrete (e.g., the change in an administrative boundary), then the discrete nature of the temporal domain should reflect this. In some cases, the domain might call for various possible futures or pasts, based on available evidence, in which case, branching time may be required. The metric nature of the temporal index is typified by temporal properties of events such as “lasted 3 days” or “occurred on July 5th”; while an example of a metric relationship between two events is “finished 5 hours before the start of.” An example of a temporal topological property is “the duration of the event had no gaps” (temporal connectedness), while an example of a topological relationship is “event A finished before event B had begun.” A key observation here is that it is not really time that is being structured, but the treatment of the underlying events.

The snapshot approach is by far the most common in current temporal database models, and is linked directly to concepts such as timestamp, temporal granularity, and temporal indexing. The general forms of such

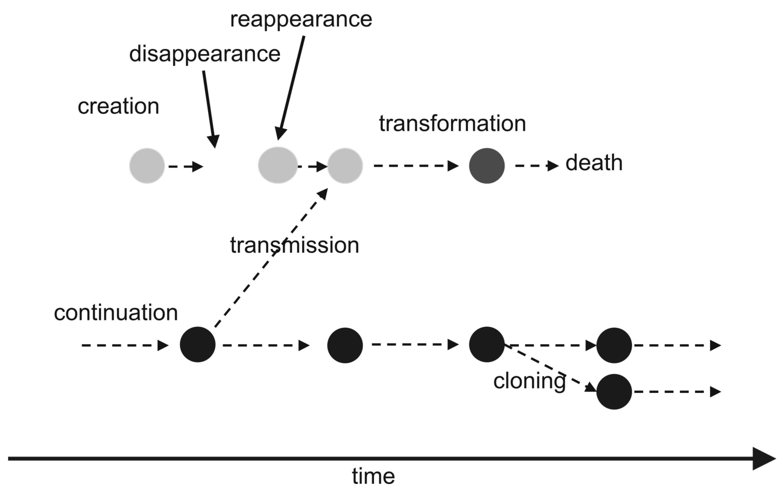

Figure 2: Object change history

temporal queries is "What was the state of this object at that time?" or its converse "At what time did this object have that state." In the case of spatiotemporal information, the query becomes "Where was this object at this time?" and its converse "At what time was the object at this location?" The literature on such temporal and spatiotemporal models and query languages is extensive, and good accounts may be found in [1, 31, 35].

2.3 Stage Two: Object change

Referring again to figure 1, we notice the construction of an airport between 1902 and 1946. This information is only given to us implicitly, through comparison of the 1902 and 1946 snapshots. The snapshot metaphor offers no mechanism for explicitly representing the time or occurrence of events such as the construction or destruction of an airport. In Stage Two, the focus shifts from the temporal sequences of objects, their attributes and relationships, to the changes that can happen to objects, attributes and relationships. This approach has been developed by Hornsby and Egenhofer in a geospatial setting [15]. Figure 2 shows some of the possibilities. In this example, creation, continuation, disappearance, reappearance, transformation, and death, are all operations that can apply to a single object; transmission is an operation performed by one object on another; and cloning allows an object to replicate itself.

The difference between Stage Two (object change) and Stage One (snapshots) can be further explained with reference to figure 3. Ignoring for a moment the annotations, this figure shows a sequence of Stage One snapshots representing the development of a neighborhood from 1908 to 1974. A Stage Two approach focuses on the changes themselves rather than the sequences of static images. The addition of the annotations to figure 3 provides a mixed Stage One–Two approach. A Stage Two representation of our example could be the following list of temporally referenced changes:

1908–1920: The property on lot 2 incorporates lot 3.

1920-1938: A school is built on lot 4.

1938-1958: The house on lot 2 is burnt down.

1958-1964: Lot 2 is divided up, part incorporated into the property on lot 1, part making a new lot 5, and part becoming a path to the school.

1964-1974: The house on lot 1 is extended and a new house is built on lot 5.

While both the figure and the above description refer to the same developing situation, it is clear that the representations are from two very different standpoints.

To further develop the Stage Two approach, we would need to develop a collection of change "primitives," such as creation, destruction, appearance, disappearance, transmission, fission, and fusion. Then, complex changes will be constructed from the primitives using a collection of predefined combinators. The details are not discussed further here because in this paper we will follow a different route to our objective.

The example above involves changes of attributes of objects, some of them spatial. It is sometimes also important to consider a particular case of change, namely *movement*. Movement occurs when a physical object changes its position, for example, when a vehicle is moving along a highway. There are clear cases where change does not involve movement, for example, the change in name of a city. There are also less clear cases. Would we call a change in the position of an administrative boundary a movement? It is certainly not a continuous move. It is also possible to consider mixed cases; an attacking army may have changing aspatial characteristics, such as the number of its soldiers, and spatial characteristics, such as its areal formation, as well as moving towards its objective. Further problems arise from "hybrid" examples, such as a wildfire or a spread of an infectious disease.

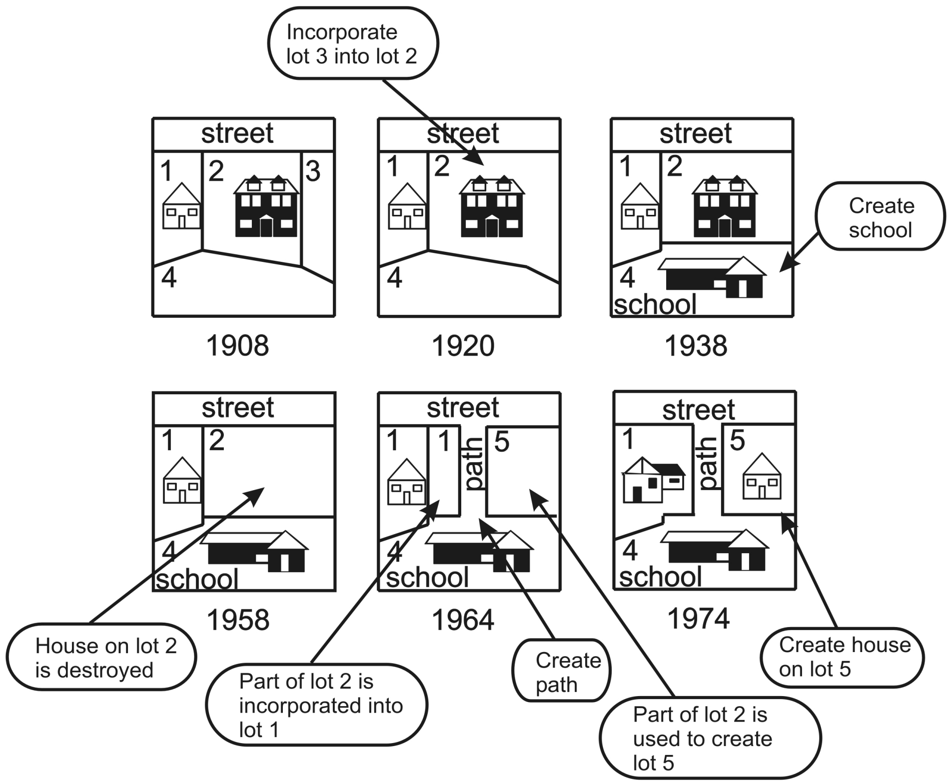

Figure 3: Neighborhood evolution example: snapshot sequence

A model of the world based on the evolution of objects through time, retaining identity but changing spatial and other attributes, seems natural. However, problems arise, particularly related to continuity of identity through time. Here is an example of the kind of tangle than can arise from some seemingly commonsense assumptions. A natural assumption to make about objects in space is that two objects occupying exactly the same space at a particular time must be the same object. However, this assumption can cause difficulties. Suppose that entities occupying physical space can be given an identity, and that at time t the object that is my house has identity H, and the object that is my house except its chimney has identity H^-. Suppose between times t and t' a storm blows the chimney off my house. At time t' the object that is my house continues to have identity H, and the object that is my house except its chimney continues to have identity H^-. However, now the objects with identities H and H^- are the same in all attributes, including filling the same space, and yet have different identities!

These kinds of issue are explored in [12].

Another kind of problem, again related to object identity, is well expressed in a version of the paradox of the ship of Theseus. Theseus, according to Greek mythology, slew the Minotaur in the Labyrinth on the island of Crete. Imagine the following scenario during the course of his voyage to Crete to meet the Minotaur. Theseus's ship began to leak, because the timber needed replacing. Theseus therefore replaced plank by plank every part of his ship and threw the old material overboard. It would be natural to think that the identity of the ship in which Theseus returned should be the same as that of the one in which he left. But suppose further that other people followed Theseus and picked up all the planks that he threw overboard, and reassembled all of those parts into a new ship, identical in physical constitution to the original. Was this reconstruction the ship of Theseus, or was it something else? Both options have their problems.

2.4 Stage Three: Events, actions and processes

The final stage in this evolution is a full-blooded treatment of change, in terms of events, actions, and processes. Galton [7] makes the distinction between *histories* that are functions from a temporal domain to attribute values, or properties of objects, and *chronicles* that treat dynamic phenomena as collections of happenings. In Stage Three we would expect to model complex events, the ways in which objects may participate in them, and relationships between events.

From an ontological perspective, we can make an initial division of entities that exist in the world into entities, *continuants*, that endure through time (e.g., tables, houses, and people) and entities, *occurrents*, that happen or occur and are then gone (e.g., lectures, people's lives, boat races). There is a difference between a city, whose characteristics are recorded by census and survey once each decade, say, and the processes of urban growth and decline, migration, and expansion, that constitute the city in flux. Grenon and Smith [8] call temporal sequences of object configurations the SNAP ontology, and the event/action/process view, the SPAN ontology. It is to SPAN that Stage Three entities of interest belong.

An initial difficulty arises concerning the meaning of terms: almost every account uses different definitions for event, process and action. However, even though terms may be used in a different way, we may pick out certain core concepts of interest in this work. Firstly, there is a distinction to be made about events/processes/actions and their specific occurrences at given times (compare the distinction between types and instances of objects). Secondly, we can distinguish those occurrents that are initiated, and sometimes terminated, by human or non-human agents; often such occurrents are termed *actions*. Thus a murder would probably be classified as an action, while an avalanche, unless caused by an agent, would not.

There is an important distinction to be made between occurrents that can be counted and those that cannot. There is a parallel here with count nouns, such as "lake", that name entities that can be counted, and mass nouns, such as "water", that name entities that are numerically uncountable and may only be quantified by a word that signifies amount. Some occurrents, such as "athletics race", may be counted, while others, such as "running", may not. There is a similar distinction in the classification of verb types presented by Vender [33] and refined by Mourelatos [25]. In this taxonomy, occurrents are either *events*, *accomplishments*, or *achievements*, that may be counted; or *processes* that may not. Mourelatos also observes that some occurrents are *homeomerous*, meaning that their parts are of the same sort as themselves. So, the activity of running, consists of running in all its temporal parts. On the other hand, there are occurrents that are not homeomerous, for example, to say that a region expanded is not to say that the expansion took place in every temporal subperiod. (Note the connection with properties such as downward- and upward-hereditary in section 3.4).

In what follows, we will begin by calling all occurrents *events*. However, we will shortly break this rule, because most of the literature that we will reference on computational occurrents uses the term *process* for a computational event. It should be emphasized that there is no claim here to the "correct" usage of terminology; the most important thing is to understand the different kinds of distinction that can be made between classes of occurrents:

The ontological status of events has been of interest to philosophers. Following Pianesi and Varzi, *et al.* [27], we can divide the philosophical positions on event occurrences into three main classes.

Events as occupations of spatiotemporal regions: In this position, set out by Quine [29], events and objects are not to be distinguished, as both are spatiotemporal entities. At most one event can occupy a given spatiotemporal region, but is capable of many possible different properties and descriptions. Thus the braking and slowing down of a vehicle at a yellow light is one event.

Events identified according to their causes and effects: According to Davidson [6], "Events have a unique position in the framework of causal relations between events in somewhat the same way that

Object-event similarities

Objects	**Events**
object instances	event occurrences
object attributes	event attributes
object taxonomy	event taxonomy
object partonomy	event partonomy
object relationships	event relationships

Object-event differences

Objects	**Events**
object endurance	event perdurance

Table 2: Object-event similarities and differences

objects have a unique position in the spatial framework of objects." In our example, the vehicle slowing event is distinct from and caused by the braking event.

Events as exemplifying a property or relationship at some time: Kim [16] argues that there may be many events occurring in the same portion of space-time. For example, the slowing of the vehicle, the braking of the vehicle, the expletive uttered by the driver, and the pressure of the foot on the brake, are all distinct events.

None of these positions seem to provide a complete account, but all can contribute something to event modeling approaches discussed below.

There are interesting modeling questions about the similarities between events and objects. Certainly, events may have instances (occurrences), attributes, belong to a subsumption hierarchy, have temporal parts, and relationships to other events. Event identifiers may be more problematic, due to the ephemeral nature of events. Table 2 shows some of the similarities and differences between objects and events. Actions have an additional structure, related to the agents that initiate or terminate them. We often speak loosely about the goal of an action, although it might be more precise to speak of the goal of the agent in initiating/terminating an action.

A fully event-oriented framework should allow us to move on from simple snapshot queries of the form "What happened at this location at this time?" to a much richer language involving the interplay between object and events, and event-event relationships.

3 Underlying approaches

This section looks at some of the approaches to temporal and spatiotemporal models and reasoning based upon logic. Formal theories of time go back at least as far as Hamilton [10], who conceived a theory of *moments* (what we may now call time instants) and *moment pairs*. A moment pair can define a time duration, and Hamilton allowed the possibility of both positive and negative durations. He provided an algebra of moments and their pairs, which we would now see as the basis of a structure of instants and durations, where for example durations may be added to moments to give new moments. The basis of many modern approaches is provided by temporal logics, and so we briefly describe the main components of these.

3.1 Tense and temporal logics

It is possible to extend both propositional and first order logic to include temporal capability. Tense logic was introduced by Prior [28] as a way of using modal operators to account for tense (past, present and future). A modal operator is a means of qualifying a proposition or first-order formula. For example, if p is the proposition "Washington is the capital of USA"; then $\Box p$ might be the proposition "Necessarily, Washington is the capital of USA". How "necessarily" is to be interpreted depends on the particular modal logic. Examples of its interpretation are "in all possible worlds"; "for all times"; "the knowledge base has information that"; or "it is believed that." Thus modal logics may capture knowledge, belief, time, and several other intentional and representational aspects. In temporal logics, there are four basic modal operators:

Gp**:** p will always be the case.

Hp**:** p has always been the case.

Pp**:** p has been the case at some time in the past.

Fp**:** p will be the case at some time in the future.

There are several logical systems based on these modalities. Among the simplest is the proof system called *temporal K*. The axioms of temporal K include all propositional tautologies as well as the axiom schemata:

$$G(p \to q) \to (Gp \to Gq) \tag{1}$$

$$H(p \to q) \to (Hp \to Hq) \tag{2}$$

$$p \to GPp \tag{3}$$

$$p \to HFp \tag{4}$$

and the derivation rules:

From p, deduce Gp
From p, deduce Hp

Thus, for example, axiom 3 ensures that if a proposition p is provable, then it is also provable that it will always be the case that p has been the case at some time in the past.

3.2 Situation calculus

Temporal logics provide a way of reasoning about states of the world as they change through time. However, situations and events are not explicitly represented. The situation calculus was developed in 1969 by McCarthy and Hayes [19]. An application domain is modeled as a collection of static, snapshot *situations*, hence the situation calculus is a basically a Stage Two approach. Each situation has a *state*, and *actions* change one situation to another. The situation calculus is a *change-based* approach, where the actions are instantaneous, have no duration, and have immediate and permanent effect upon situations (delayed effects are not a feature of this approach, neither can actions have only temporary effects upon situations). This allows some Stage Two functionality, but relationships between actions, such as concurrency, cannot be expressed in the situation calculus.

3.3 Event calculus

The event calculus was introduced by Kowalski and Sergot [17] and is discussed in a more recent paper of Miller and Shanahan [21]. This calculus does allow events to be explicitly represented, and is therefore a Stage Three approach. The event calculus is narrative-based, and its principal constituents are *events*, each of which is an instance of an *event type*. A *fluent* is a time-varying property of the domain, expressed by a proposition that evaluates to true or false, depending on the time and the occurrence of relevant events. A fluent is *true* at a time point if it has been *initiated* by an *event* at some earlier time point, and has not since been *terminated* by another event. Otherwise a fluent is *false*. The basic predicates are `Occurs(event, time)`, `HoldsAt(fluent, time)`, `Initiates(event, fluent, time)`, and `Terminates(event, fluent, time)`. Examples of the ensuing theory are: "A fluent is true once it has been initiated by an event," and "A fluent is false once it has been terminated and before it has been initiated." Although this is a Stage Three approach, only instants of time are represented, so the calculus is limited to punctual events.

3.4 Interval temporal logic

When the temporal domain of a temporal logic consists of time intervals, then the logic is referred to as an *interval temporal logic*. An *(interval) temporal proposition* is a statement associated with a time interval, in which it may or may not hold. Thus the temporal proposition "Washington is the capital of USA" holds during the year 2003. Because the temporal reference of a temporal proposition is an interval, and so has duration, its internal structure is more complex than for the event calculus. Following [30], among the ways that a temporal proposition P can be classified are the following:

downward-hereditary: If P holds during an interval, then P hold during all subintervals of the interval. Example: "The vehicle has moved less than five miles from its original station." (Note that any homeomerous event is associated with a downward-hereditary property).

upward-hereditary: If P it holds for all proper, non-point subintervals of an interval, then P holds for the interval. Example: "The vehicle's average speed is faster than 5mph."

clay-like: If P holds during two consecutive intervals, then P holds during their union. Example: "The journey started and ended at a pub".

gestalt: Proposition P never holds over two intervals, one of which properly contains the other. Example: "The moving vehicle covered one mile."

solid: Proposition P never holds over two properly overlapping intervals. Example: "The vehicle travelled directly from Boston to New York."

In [2], Allen developed a calculus of temporal intervals, in which the underlying structure of time is linear and a time interval is a connected temporal duration, such as between 3.00 and 4.00 this afternoon. The calculus provides a pairwise independent and mutually exhaustive collection of

Relation	**Example**
`before`	$[0, 10]$ `before` $[20, 30]$
`equals`	$[0, 10]$ `equals` $[0, 10]$
`meets`	$[0, 10]$ `meets` $[10, 30]$
`overlaps`	$[0, 10]$ `overlaps` $[5, 30]$
`during`	$[10, 20]$ `during` $[0, 30]$
`starts`	$[10, 20]$ `starts` $[10, 30]$
`finishes`	$[20, 30]$ `finishes` $[10, 30]$

Table 3: Allen's temporal interval relations

relations between temporal intervals, as shown in table 3. The right hand column shows examples of the relations using temporal intervals of the form $[i, j]$, where i and j are time points in some unit of measurement.

From the base set, other temporal relations may be derived. For example, the relation `in` is defined as follows:

$$\texttt{in}(t_1, t_2) \equiv (\texttt{during}(t_1, t_2) \vee \texttt{starts}(t_1, t_2) \vee \texttt{finishes}(t_1, t_2))$$

where t_1 and t_2 are temporal intervals. So, $[0, 10]$ `in` $[0, 30]$.

Allen and colleagues [3, 4] have used interval temporal logic to provide an extension of the event calculus of the preceding section using the above calculus of temporal intervals. This gives quite a rich framework for representing and reasoning about events, and is definitely a Stage Three approach. Events now take place over time intervals and can have relationships dependent on their interval relationships. This allows each event to have a rich internal structure. In a similar way to the event calculus, the predicate $\texttt{holds}(\texttt{property}, \texttt{time})$ is introduced, that asserts that a property holds during a time interval.

To illustrate the representational power of interval temporal logic, the following equivalence expresses the fact that property p is downward hereditary.

$$\texttt{holds}(p, t) \equiv \forall t'(\texttt{in}(t', t) \Rightarrow \texttt{holds}(p, t'))$$

Again, as with the event calculus, a predicate `occur(event, time)` is introduced to convey that an event happened over a time interval. If we assume that events are non-homeomerous, then we have the following:

$$(\texttt{occur}(e, t) \wedge \texttt{in}(t', t)) \Rightarrow \neg\texttt{occur}(e, t')$$

In fact, Allen uses the above equation to distinguish between events, satisfying it, and processes, satisfying the equation that follows. Processes and events are taken to be subtypes of the more general type, occurrence. A new predicate, `occurring(process, time)`, is introduced that satisfies:

$$\texttt{occurring}(r,t) \Rightarrow \exists t'((\texttt{in}(t',t) \wedge \texttt{occurring}(r,t'))$$

Actions require agents that initiate them. Therefore, a new function `acause(agent, occurrence)` is introduced that, for an agent and an occurrence, produces the action of the agent causing the occurrence. This leads to a further set of axioms, an example of which is:

$$\texttt{occur}(\texttt{acause}(a,e),t) \Rightarrow \texttt{occur}(e,t))$$

The interpretation of this axiom is that the occurrence of an action resulting in an event e at time t implies the occurrence of event e itself at t. A fuller axiom set is provided in [3], which includes an account of how actions can lead to other axioms, and agents' intentions.

4 Process calculi: formal models of concurrent occurrents

An alternative to logical approaches is to construct algebraic theories in which occurrents are treated as first-class entities. There is a large literature describing this strategy in the case where occurrents are computational processes, traceable back at least to Petri's work on asynchronous flow [26]. Later work on the algebraic approach can be found in [22, 14, 13, 32]. Following this literature we will use the term *process* for occurrents that are computational processes. The motivations for this work are well expressed in Milner's Turing Award lecture [23], which discusses the move from traditional models of a single computation (e.g., a Turing Machine) to models of many computations going on together. A model of a single computation (or, in real–world terms, a single occurrent) can be provided by the mathematical theory of functions; hence the important role of functional programming languages. However, when many occurrents acting together need to be modeled, function theory is insufficient. Database transactions provide a good example. Database transactions, for example to simultaneous updates of the same item, can interfere with each other, resulting in problems such as "lost update". The key missing ingredients that need to be added are *concurrency* and *interaction*, and it is to these primary concepts that we now turn. Both concepts will be important for analysis of real-world occurrents, which will be going on concurrently and impinging upon each other in various ways. This section describes some of the basic elements in the theory of computational processes. Something to keep in mind here is that computational processes are rather like computer programs, which when executed result in occurrents: they have the potential to become occurrents.

4.1 Process definition and basic operations

Complex processes may be constructed using algebraic combinator operations acting on a base collection of atomic actions. The term *action* here is that traditionally used in process calculi. This connects with the discussion in the previous section of actions as being performed by agents; here, the agent is a computer on which the action is executed.

We begin with a collection A of atomic actions. A includes the action 0, indicating stop, or do nothing. In computational processes, as with real-world occurrences, a fundamental notion is one occurrence followed by another. If processes can be viewed as mathematical functions (and we have already noted that this is too limited an abstraction), then the corresponding construction here is function composition. Indeed, if P and Q are processes, and if Q may be thought of as P followed by some atomic action a, we write the simple *transition diagram* $P \xrightarrow{a} Q$ (we try to use upper-case beginnings to process names and lower-case for atomic actions). We may write this as an equation to define Q:

$$Q \stackrel{\text{def}}{=} a.P$$

The summation operator + allows binary choice of process. For processes P, Q, R, and actions a, b, the definitional equation:

$$R \stackrel{\text{def}}{=} a.P + b.Q$$

shows that R is the name for the process that allows either the action a followed by process P or b followed by Q. The transition diagram corresponding to this equation is shown in figure 4. A process of the form $R \stackrel{\text{def}}{=} a.P + a.Q$ is called *nondeterministic*, because it can perform action a and then has to choose between moving to process P or R.

We now begin to show how process calculi constructions, although originally designed to model computational processes, can be applied to real-world occurrents. The full story can only be told when space and time have been incorporated into the model. But, even with the limited techniques so

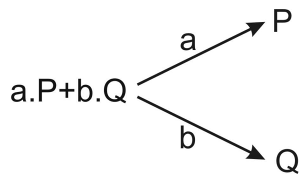

Figure 4: Transition diagram for summation

far described, we can model some complex real-world events, as the following example shows.

Example 1. Four way stop protocol
Some intersections in North America are designated "four way stops". There are no traffic lights at these intersections. At a four way stop, vehicles must come to a complete halt before proceeding one at a time across the intersection, regardless of whether there is any other traffic in sight or not, and also regardless of whether the vehicle is making a turn at the junction or not. If two or more vehicles draw up to such an intersection, then they should proceed in the same order in which they arrived. The first vehicle to arrive is the first to proceed, the second vehicle to arrive is the second to proceed, and so on.

We use the algebra of processes to represent the four way stop protocol. Assume the four roads incoming to the intersection are labelled R_i for $i \in \{1, 2, 3, 4\}$. Let the arrival of a vehicle at the stop line for road R_i be indicated by action a_i. Let the direction to move from the stop line, into the intersection and out by any of the other roads, be indicated by the action $\overline{b_i}$. (The notation a and $\bar{b}$ to represent input and output actions will be discussed in more detail in the next section).

The traffic protocol is given by the equations below, where all subscripts

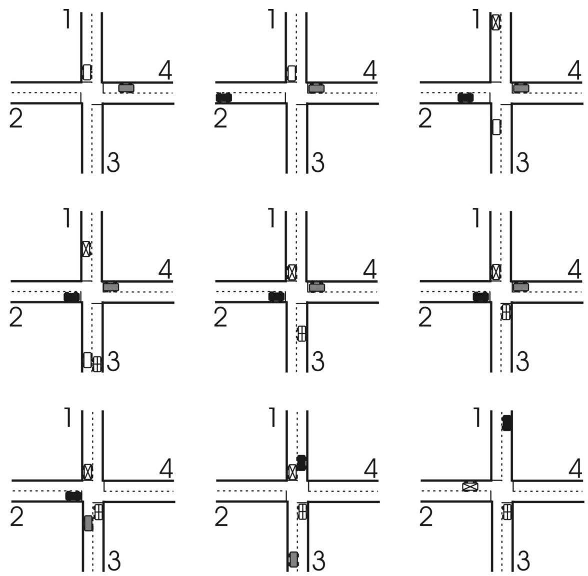

Figure 5: Four way stop example

run over the index set $\{1, 2, 3, 4\}$ with restrictions as given.

$$X = \sum_i a_i X_i$$

$$X_i = \overline{b_i} X + \sum_{j \neq i} a_j X_{ij}$$

$$X_{ij} = \overline{b_i} X_j + \sum_{k \neq i,j} a_k X_{ijk}$$

$$X_{ijk} = \overline{b_i} X_{jk} + \sum_{l \neq i,j,k} a_l X_{ijkl}$$

$$X_{ijkl} = \overline{b_i} X_{jkl}$$

Each process $X_{ij\ldots}$ is associated with state $x_{ij\ldots}$, where there are vehicles waiting at the stop lines of the roads $i, j, \ldots$, having arrived in the order $i, j, \ldots$. Figure 5 shows an example of this. A vehicle arrives at intersection R_1 followed by another at R_4. The vehicle at R_1 moves off and new vehicles arrive in order at R_2, R_1, and R_3. Vehicles then move off in the order, leaving a single vehicle at R_3. The process execution sequence for this is:

$$X \xrightarrow{a_1} X_1 \xrightarrow{a_4} X_{14} \xrightarrow{\overline{b_1}} X_4 \xrightarrow{a_2} X_{42} \xrightarrow{a_1} \cdots$$

$$\cdots \xrightarrow{a_1} X_{421} \xrightarrow{a_3} X_{4213} \xrightarrow{\overline{b_4}} X_{213} \xrightarrow{\overline{b_2}} X_{13} \xrightarrow{\overline{b_1}} X_3$$

4.2 Process concurrency and reaction

The real purpose of process calculi is to formally model not just the sequential execution of a single process but a collection of processes acting together and reacting with each other. This is the case where functional models become insufficient. Several processes are *concurrent* if they are contained within the same time frame. This is not to say that the processes must act in any way in connection, nor that they act at exactly the same time. The fundamental formal operation is *concurrent composition*, and we write that the concurrent composition of processes P and Q is $P|Q$.

To discuss interaction between processes, we need to partition A into two sets, the positive or *input* actions, and the negative or *output* actions. By convention, output actions are indicated with an overline, for example, $\overline{a}$. Such a pair of actions, a and $\overline{a}$, is called a *complementary pair.* A *reaction* can occur between two concurrent processes if they have a complementary pair of actions that can be matched. Formally, if $P \stackrel{\text{def}}{=} \overline{a}.P'$ and $Q \stackrel{\text{def}}{=} a.Q'$, then there is an internal action τ so that:

$$P|Q \xrightarrow{\tau} P'|Q'$$

The idea is that the concurrent execution of complementary actions a and $\overline{a}$ result in a *handshake* between processes P and Q that leads to a momentary interaction between them and after which they proceed to independent concurrent executions of P' and Q'.

4.3 Message passing

Milner's development of the π-calculus [24] extends the internal action τ to allow reacting concurrent processes to pass messages. Actions are generalized to contain parameters, that will carry the messages in the new reactions. So, now we have three kinds of action:

1. $a(x)$ that, under the appropriate reaction, allows the input of message to be substituted for variable x along the channel named a.

2. $\overline{a}(v)$, that, under the appropriate reaction, allows the output of message v along the channel named a.

3. τ, the internal action.

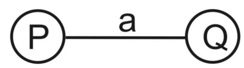

Figure 6: Processes P,Q, and their communication channel a

As before, the actions A are partitioned into two sets, the *input* and *output* actions. Now, we also allow actions to convey data from input to output, so the output action $\overline{a(v)}$ has the capability of passing the data v, which another process with input action $a(x)$ may receive by substituting v for x in free occurrences of variable x. As before, such a pair of actions is called a *complementary pair*. A *reaction* can occur between two concurrent processes in the following way. Suppose we have parameterized processes $P(v)$ and$Q(x)$ defined as follows:

$$P(v) \stackrel{\text{def}}{=} \overline{a(v)}.P'$$
$$Q(x) \stackrel{\text{def}}{=} a(x).Q'(x)$$

Then, there is the possibility of a transition $P(v)|Q(x) \rightarrow P'|Q'(v)$. This reaction permits the passing of parameter v from P to Q. The pair of actions, $a(x)$ and $\overline{a(v)}$ together constitute a *channel of communication*, by means of which, message v may be passed. These processes, and their possibility of communicating by means of channel a, can be shown diagrammatically, as with figure 6. Input and output actions may be interpreted in a more general way than data input and output. The following example illustrates how the name of a process may be passed as a message.

Example 2. Toll booth protocol
To illustrate some of these constructions, we partially model the process of a vehicle passing through a toll booth on a toll highway. The process $\mathsf{ThroughToll}(m)$ takes as a parameter the money deposited into the machine at the booth. The action of the machine is represented by another parametric process, $\mathsf{Machine}(x)$, which takes the money x and if it is the correct amount executes $\mathsf{CorrectMoney}$. The process Light goes green and reacts with $\mathsf{ThroughToll}(m)$ to a go, provided that the correct money has been inserted into the machine. The equations defining these processes are:

$$\mathsf{ThroughToll}(m) = \overline{\mathsf{Pay}(m)}.\mathsf{green}.\mathsf{Proceed}$$
$$\mathsf{Machine}(x) = \mathsf{Pay}(x).x$$
$$\mathsf{CorrectMoney} = \overline{\mathsf{go}}.\mathsf{Machine}$$
$$\mathsf{Light} = \mathsf{go}.\overline{\mathsf{green}}.\mathsf{Light}$$

The situation begins with the following concurrent processes:

$$\mathsf{ThroughToll}(\mathsf{CorrectMoney})|\mathsf{Machine}(x)|\mathsf{Light}$$

The processes $\mathsf{ThroughToll}(\mathsf{CorrectMoney})$ and $\mathsf{Machine}(x)$ can react through the input and output Pay actions (the Pay channel), leading to the substitution of $\mathsf{CorrectMoney}$ for x and the concurrent processes:

$$\mathsf{green.Proceed}|\mathsf{CorrectMoney}|\mathsf{Light}$$

The processes $\mathsf{CorrectMoney}$ and Light can now react through the go channel, leading to:

$$\mathsf{green.Proceed}|\mathsf{Machine(x)}|\overline{\mathsf{green}}.\mathsf{Light}$$

Finally, the processes $\mathsf{green.Proceed}$ and $\overline{\mathsf{green}}.\mathsf{Light}$ can react through the input and output green actions, leading to:

$$\mathsf{Proceed}|\mathsf{Machine(x)}|\mathsf{Light}$$

The vehicle can proceed, and the machine and light are ready for the next vehicle.

5 Real-world actions, events and processes

In this section we take the process calculus briefly outlined above, and show that it may be used to represent and reason about real-world occurrents. The process calculi of Milner and others [22, 14, 13, 32], are concerned primarily with computational occurrents - machine processes, represented by the actions of collections of automata, for example. In order to make this work relevant to real-world occurrences, we must firmly embed these computational processes in space and time. Our key idea is a pure process model of the world: *everything is process*. To develop this idea, we now show how spatial and temporal reference frames can be modeled as processes in the process calculus.

5.1 Clocks, time, and temporal entities

We begin by constructing a process-oriented model of time, and use it to define temporally-referenced entities. The basic idea behind our event-oriented approach to time is that the ticking of a clock is a real-world process linked to the advancement of time. It is quite natural to link time to the process of ticking, and we take the 'tick' processes as atomic, and define the temporal

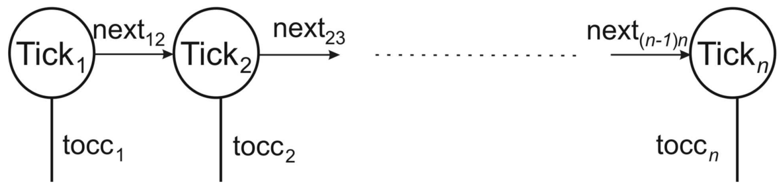

Figure 7: Processes representing linear time by means of a clock

domain (assumed linear) as a sequence of ticks. In our pure event-oriented approach, time *is* the sequence of ticking processes.

There is a choice as to whether to present time as a single tick process that recursively "calls" itself, or as a collection of separate but sequenced tick processes. We choose the latter, and to this end construct a collection of processes, $\mathsf{Tick}_1, \ldots, \mathsf{Tick}_n$. The collection is structured into a linear order by means of a set of channels $\mathsf{next}_{i(i+1)}$, for $i = 1, \ldots, n$, such that two consecutive ticks Tick_i and Tick_{i+1} share the common channel $\mathsf{next}_{i(i+1)}$. Each $\mathsf{next}_{i(i+1)}$ channel is directed: one cannot go back in time, or, more precisely, if backwards time travel is required, then it must be explicitly modeled.

The main purpose of this construction is the need to be able to model temporally-referenced entities. To this end, each tick process, Tick_i, has a channel tocc_i that is capable of being occupied by temporally-referenced entities. Any configuration of Tick processes with the minimum amount of structure as above will be referred to as a *clock*. The arrangement is shown in figure 7. More precisely, the finite process Clock_n consists of the concurrent composition of n linked tick processes, each with tocc channels. The definition of Clock_n is given as:

$$\mathsf{Clock}_n = \mathsf{Tick}_1 | \ldots | \mathsf{Tick}_n$$

A model of the dynamic world will contain at least one clock. We are now ready to state precisely what we mean by a temporally-referenced entity.

Definition A *temporally referenced entity* is an entity that handshakes with at least one clock by means of its tocc channels. A temporally referenced entity may have a *duration*, indicated by *begin* and *end* handshakes with the tocc channels of its (usually the same) reference clock.

Depending on the application, there may also be other channels through which each "tick" of a Clock can communicate; for example, it may need to communicate its clock time to other interested process. Of course, we could also set up other temporal structures, for example, branching or cyclic time, by altering the arrangement of next channels. In *synchronous* domain models, all temporally referenced entities handshake with processes in the same clock. In *asynchronous* domain models, temporally referenced entities may handshake with processes in different clocks. Another feature to note about clocks is that there is no in-built notion of regularity in the time-keeping. Each tick of the basic clock occupies a duration that is unspecified. Clock regularity can of course be built in as an extra feature or assumption.

5.2 Locations, space, and spatial entities

It is natural to represent time as a process, or collection of processes, as we measure time using processes such as the ticking of clocks. A process view of space requires more of a conceptual leap. However, the measurement and communication of location also require processes. (A collection of location-aware cell-phones provides an example of a process-oriented view of location).

For the purposes of this paper, let us assume that space is to be represented as a connected region, partitioned into a set of blocks, termed *locations*. Locations are related to their neighbors by a directed adjacency relation. The idea is to represent each location as a process that handshakes with its neighbors through its adjacency relations, and has the capability of handshaking with an occupying entity, itself represented by a process.

More precisely, construct a collection of location processes, $\mathsf{Loc}_1, \ldots, \mathsf{Loc}_n$. Two adjacent locations, Loc_i and Loc_j share the common channels adj_{ij} and adj_{ji}. The two directional boundaries are modeled explicitly, thus providing more modeling flexibility (e.g., the ability to model one-way-only constraints).

In a similar way to the construction of temporal processes, location processes need to be able to provide spatial references for selected entities. To this end, each location, Loc_i, also has a channel, socc_i, through which it can handshake with occupying entities. Depending on the application, the socc can be parameterized to communicate other information, such as its position, to its occupants. There might also be other channels that can communicate to other non-occupant entities. For example, a geo-sensor location process might communicate to neighboring sensors that it is currently occupied. A Region is now formally defined as the concurrent composition of a collection of constituent locations. To illustrate these concepts, figure 8 shows a process representation of a region, Region_5 divided into locations

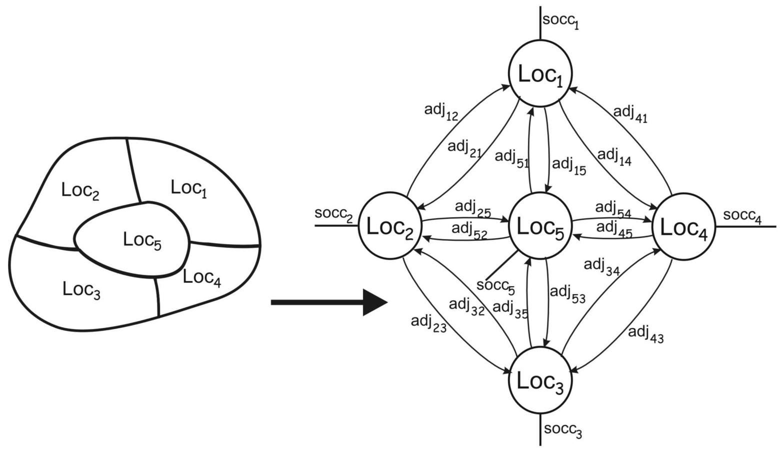

Figure 8: Processes representing locations in a spatial region

with adjacency relations. Formally, we have:

$$\mathsf{Region}_5 = \mathsf{Loc}_1|\ldots|\mathsf{Loc}_5$$

Along with a clock, each model of the dynamic world must contain at least one region process. This gives us the following definition:

Definition A *spatially referenced entity* is an entity that handshakes with at least one region by means of socc channels.

In *syntopic* domain models, all spatially referenced entities handshake with processes in the same region. In *asyntopic* domain models, spatially referenced entities may handshake with processes in different regions. The asyntopic case may happen, for example, when two different collections of sensors are being used as spatial referents.

The final notion in this section is that of a *spatiotemporal entity* (ST-entity), which is both spatially and temporally referenced. The dynamic nature of the world can now be modeled using structured collections of ST-entities.

6 Case study: Motion

To illustrate the ideas of the previous section, we show how motion can be modeled using the process calculi formalism enhanced by the spatial and temporal embeddings described above. We model the motion of a vehicle along a route through a region. To keep the formalism relatively concise,

Location	**Time**
Loc_1	Tick_1
Loc_2	Tick_2
Loc_3	Tick_3

Table 4: Spatiotemporal trajectory of the vehicle

only three times, Tick_1, Tick_2, and Tick_3 are modeled as Clock_3. The spatial region, is the region Region_5 used in the previous section, consisting of 5 locations, $\mathsf{Loc}_1, \ldots, \mathsf{Loc}_5$. The vehicle is preset to follow the spatiotemporal trajectory shown in table 4. The initial state of the vehicle is shown in figure 9, and the constituent processes model is given below in equations 5–13. The complete model is the concurrent composition Motion of all the constituent processes, defined by:

$$\mathsf{Motion} \stackrel{\text{def}}{=} \mathsf{Vehicle}|\mathsf{Clock}_3|\mathsf{Region}_5$$

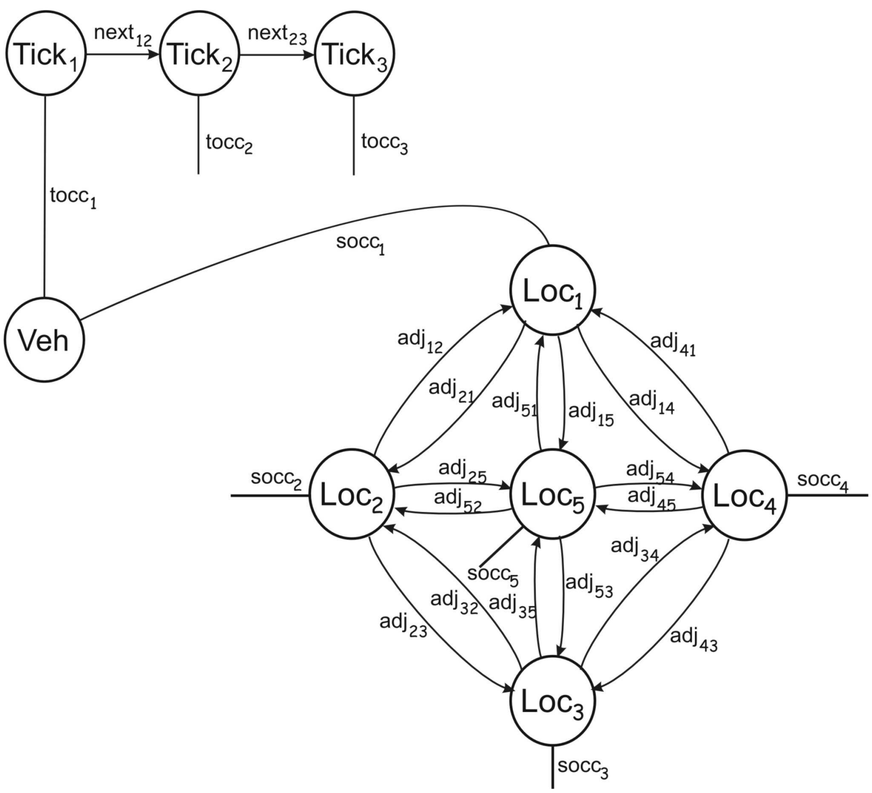

Figure 9: The initial stages of connection between processes for the motion example

$$\begin{aligned}
\mathsf{Tick}_1 &= \mathsf{tstart}_1.\mathsf{tocc}_1.\overline{\mathsf{next}}_{12}.\mathsf{Tick}_1 && (5)\\
\mathsf{Tick}_2 &= (\mathsf{tstart}_2 + \mathsf{next}_{12}).\mathsf{tocc}_2.\overline{\mathsf{next}}_{23}.\mathsf{Tick}_2 && (6)\\
\mathsf{Tick}_3 &= (\mathsf{tstart}_3 + \mathsf{next}_{23}).\mathsf{tocc}_3.\mathsf{Tick}_3 && (7)\\
\mathsf{Loc}_1 &= \\
&\quad (\mathsf{sstart}_1 + \mathsf{adj}_{21} + \mathsf{adj}_{51} + \mathsf{adj}_{41}).\mathsf{socc}_1.\\
&\quad (\overline{\mathsf{adj}}_{12} + \overline{\mathsf{adj}}_{15} + \overline{\mathsf{adj}}_{14}).\mathsf{Loc}_1 && (8)\\
\mathsf{Loc}_2 &= \\
&\quad (\mathsf{sstart}_2 + \mathsf{adj}_{12} + \mathsf{adj}_{52} + \mathsf{adj}_{32}).\mathsf{socc}_2.\\
&\quad (\overline{\mathsf{adj}}_{21} + \overline{\mathsf{adj}}_{25} + \overline{\mathsf{adj}}_{23}).\mathsf{Loc}_2 && (9)\\
\mathsf{Loc}_3 &= \\
&\quad (\mathsf{sstart}_3 + \mathsf{adj}_{23} + \mathsf{adj}_{53} + \mathsf{adj}_{43}).\mathsf{socc}_3.\\
&\quad (\overline{\mathsf{adj}}_{32} + \overline{\mathsf{adj}}_{35} + \overline{\mathsf{adj}}_{34}).\mathsf{Loc}_3 && (10)\\
\mathsf{Loc}_4 &= \\
&\quad (\mathsf{sstart}_4 + \mathsf{adj}_{14} + \mathsf{adj}_{34} + \mathsf{adj}_{54}).\mathsf{socc}_4.\\
&\quad (\overline{\mathsf{adj}}_{41} + \overline{\mathsf{adj}}_{43} + \overline{\mathsf{adj}}_{45}).\mathsf{Loc}_4 && (11)\\
\mathsf{Loc}_5 &= \\
&\quad (\mathsf{sstart}_5 + \mathsf{adj}_{15} + \mathsf{adj}_{25} + \mathsf{adj}_{35} + \mathsf{adj}_{45}).\mathsf{socc}_5.\\
&\quad (\overline{\mathsf{adj}}_{51} + \overline{\mathsf{adj}}_{52} + \overline{\mathsf{adj}}_{53} + \overline{\mathsf{adj}}_{54}).\mathsf{Loc}_5 && (12)\\
\mathsf{Vehicle} &= \overline{\mathsf{tstart}}_1.\overline{\mathsf{sstart}}_1.\overline{\mathsf{tocc}}_1.\overline{\mathsf{socc}}_1.\overline{\mathsf{tocc}}_2.\overline{\mathsf{socc}}_2.\overline{\mathsf{tocc}}_3.\overline{\mathsf{socc}}_3.0 && (13)
\end{aligned}$$

The reactions that take place within Motion are now considered. For readability, we show at each stage only those processes within Motion that can react. We also use the notation $a : P$ to indicate that action a of process P is taking part in the reaction. So,

$$\overline{a} : P \mid a : Q \rightarrow b : P \mid c : Q$$

indicates that within the full concurrent composition, actions $\overline{a}$ and a of processes P and Q react together, resulting in action b and c, respectively, of processes P and Q, to be performed when future reactions become available. The reactions are set out in equations I1–I12 below.

$$\overline{\mathsf{tstart}}_1 : \mathsf{Vehicle}|\mathsf{tstart}_1 : \mathsf{Tick}_1 \to \overline{\mathsf{sstart}}_1 : \mathsf{Vehicle}|\mathsf{tocc}_1 : \mathsf{Tick}_1 \tag{I1}$$

$$\overline{\mathsf{sstart}}_1 : \mathsf{Vehicle}|(\mathsf{sstart}_1 + \mathsf{adj}_{21} + \mathsf{adj}_{51} + \mathsf{adj}_{41}) : \mathsf{Loc}_1 \to \overline{\mathsf{tocc}}_1 : \mathsf{Vehicle}|\mathsf{socc}_1 : \mathsf{Loc}_1 \tag{I2}$$

$$\overline{\mathsf{tocc}}_1 : \mathsf{Vehicle}|\mathsf{tocc}_1 : \mathsf{Tick}_1 \to \overline{\mathsf{socc}}_1 : \mathsf{Vehicle}|\overline{\mathsf{next}}_{12} : \mathsf{Tick}_1 \tag{I3}$$

$$\overline{\mathsf{socc}}_1 : \mathsf{Vehicle}|\mathsf{socc}_1 : \mathsf{Loc}_1 \to \overline{\mathsf{tocc}}_2 : \mathsf{Vehicle}|(\overline{\mathsf{adj}}_{12} + \overline{\mathsf{adj}}_{15} + \overline{\mathsf{adj}}_{14}) : \mathsf{Loc}_1 \tag{I4}$$

$$\overline{\mathsf{next}}_{12} : \mathsf{Tick}_1|(\mathsf{tstart}_2 + \mathsf{next}_{12}) : \mathsf{Tick}_2 \to \mathsf{tstart}_1 : \mathsf{Tick}_1|\mathsf{tocc}_2 : \mathsf{Tick}_2 \tag{I5}$$

$$\overline{\mathsf{tocc}}_2 : \mathsf{Vehicle}|\mathsf{tocc}_2 : \mathsf{Tick}_2 \to \overline{\mathsf{socc}}_2 : \mathsf{Vehicle}|\overline{\mathsf{next}}_{23} : \mathsf{Tick}_2 \tag{I6}$$

$$\overline{\mathsf{next}}_{23} : \mathsf{Tick}_2|(\mathsf{tstart}_3 + \mathsf{next}_{23}) : \mathsf{Tick}_3 \to (\mathsf{tstart}_2 + \mathsf{next}_{12}) : \mathsf{Tick}_2|\mathsf{tocc}_3 : \mathsf{Tick}_3 \tag{I7}$$

$$(\overline{\mathsf{adj}}_{12} + \overline{\mathsf{adj}}_{15} + \overline{\mathsf{adj}}_{14}) : \mathsf{Loc}_1|(\mathsf{sstart}_2 + \mathsf{adj}_{12} + \mathsf{adj}_{52} + \mathsf{adj}_{32}) : \mathsf{Loc}_2 \to (\mathsf{sstart}_1 + \mathsf{adj}_{21} + \mathsf{adj}_{51} + \mathsf{adj}_{41}) : \mathsf{Loc}_1|\mathsf{socc}_2 : \mathsf{Loc}_2 \tag{I8}$$

$$\overline{\mathsf{socc}}_2 : \mathsf{Vehicle}|\mathsf{socc}_2 : \mathsf{Loc}_2 \to \tag{I9}$$

$$\overline{\mathsf{tocc}}_3 : \mathsf{Vehicle}|(\overline{\mathsf{adj}}_{21} + \overline{\mathsf{adj}}_{25} + \overline{\mathsf{adj}}_{23}) : \mathsf{Loc}_2 \tag{14}$$

$$\overline{\mathsf{tocc}}_3 : \mathsf{Vehicle}|\mathsf{tocc}_3 : \mathsf{Tick}_3 \to \overline{\mathsf{socc}}_3 : \mathsf{Vehicle}|(\mathsf{tstart}_3 + \mathsf{next}_{23}) : \mathsf{Tick}_3 \tag{I10}$$

$$(\overline{\mathsf{adj}}_{21} + \overline{\mathsf{adj}}_{25} + \overline{\mathsf{adj}}_{23}) : \mathsf{Loc}_2|(\mathsf{sstart}_3 + \mathsf{adj}_{23} + \mathsf{adj}_{53} + \mathsf{adj}_{43}) : \mathsf{Loc}_3 \to (\mathsf{sstart}_2 + \mathsf{adj}_{12} + \mathsf{adj}_{52} + \mathsf{adj}_{32}) : \mathsf{Loc}_2|\mathsf{socc}_3 : \mathsf{Loc}_3 \tag{I11}$$

$$\overline{\mathsf{socc}}_3 : \mathsf{Vehicle}|\mathsf{socc}_3 : \mathsf{Loc}_3 \to 0 : \mathsf{Vehicle}|(\overline{\mathsf{adj}}_{32} + \overline{\mathsf{adj}}_{35} + \overline{\mathsf{adj}}_{34}) : \mathsf{Loc}_3 \tag{I12}$$

The Vehicle process has built-in control of the spatiotemporal trajectory, through the predetermined sequence of interactions with temporal and spatial processes, as given in Table 4. The Motion process begins with an interaction I1 between constituent processes Vehicle and Tick_1 that starts the vehicle in motion at the determined time, and then I2 between Vehicle and Loc_1 to start at the determined place. The reaction sequence involves the following further relevant kinds of reactions:

Occupation of temporal locations Interactions I3, I6, I10 between Vehicle and Tick_i using the tocc_i channels, that establish the vehicle at a position in time.

Occupation of spatial locations Interactions I4, I9, I12 between Vehicle and Loc_i using the socc_i channels, that establish the vehicle at a position in space.

Temporal transitions Interactions I5, I7, between neighboring Tick processes using the next channels, that move the scenario on in time.

Spatial transitions Interactions I8, I11, between neighboring Loc processes using the adj channels, that move the scenario on in space.

There are also some 'irrelevant' reactions that can take place. For example, reaction I8 between Loc_1 and Loc_2 occurs by means of channel adj_{12}. However, also possible is the following reaction I8* between Loc_1 and Loc_4 using channel adj_{14}. But this chain of reactions leads to a dead-end, as the vehicle's predetermined route cannot now connect with Loc_4.

$$(\overline{\mathsf{adj}}_{12} + \overline{\mathsf{adj}}_{15} + \overline{\mathsf{adj}}_{14}) : \mathsf{Loc}_1 | (\mathsf{sstart}_4 + \mathsf{adj}_{14} + \mathsf{adj}_{34} + \mathsf{adj}_{54}) : \mathsf{Loc}_2 \rightarrow$$
$$(\mathsf{sstart}_1 + \mathsf{adj}_{21} + \mathsf{adj}_{51} + \mathsf{adj}_{41}) : \mathsf{Loc}_1 | \mathsf{socc}_4 : \mathsf{Loc}_4 \qquad \text{(I8*)}$$

As an extension, we can formalize *indeterminate* motion through the region, following only legal adjacencies in space and time. Equations I1–I12 could be modified by removing all subscripts from the socc and tocc channels. In this way, there would no longer be a set track for the vehicle. Now the only fixed points of the motion are that the vehicle starts at location Loc_1 at time Tick_1, and finishes after whatever tocc and socc pairs are built into the vehicle's itinerary process. The starting times and places can also be changed by modifying tstart and sstart. It is also not difficult to build into this model notions of speed and direction of travel.

7 Conclusions

This paper sets out some of the requirement for representation and reasoning about real-world occurrents, where these entities are treated directly and explicitly in the semantics. We provide a staged review of developments towards such a treatment, from sequences of temporal snapshots, through object life histories, to event chronicles. On the way, we note the important ontological distinction between continuants and occurrents, and suggest that treatments of occurrents as 'first-class entities' in the representation need different approaches from the now traditional object-oriented paradigm. The paper also discusses some of the most relevant approaches to representation and reasoning with occurrents. Approaches based on extensions to logic, where occurrent names are allowed as explicit terms in the formalism, include the situation and event calculi. We also describe the added richness that logics involving temporal intervals provide.

The relationship between the stages and approaches, whether based on logic or process calculi, gives rise to two issues:

1. The match between approaches to stages in the evolution of temporal and spatiotemporal models.

2. Translation between stages. For example, how would a process-oriented model of a transportation event be translated into a snapshot-oriented query to a current spatiotemporal database?

The principal new area of development introduced here is the application of calculi, developed mainly in the context of computational processes, to real-world occurrents. We develop a process theory of space and time, and show how this provides a rich semantics that allows us to speak of synchronous *vs.* asynchronous, and syntopic *vs.* asyntopic processes, as well as spatially, temporally, and spatiotemporally referenced entities. To show the power of this approach we provide an example of the representation of motion of a vehicle through a region.

We believe that spatiotemporally extended process calculi have a great deal to contribute to our understanding, representation, and reasoning about the dynamic world. However, the work here provides only the beginning; the theory needs to be scaled up to work with full-scale processes in the world. Also, there are other important connections that have not been developed in this paper. For example, work on agent-based systems and simulations, and active databases, speak relevantly to our key concerns. Theories of granularity will need to be introduced to provide levels of detail at which process models can be viewed and manipulated. We envision applications for process models of geographic space-time that involve representation of coordination and control semantics, for example, in transportation, coordination of relief and rescue resources, and defense.

Some immediate follow-ups to work described in this paper include the following:

- What is the role of objects in a pure process model? Clearly, any useful model will neither be pure object nor pure process, and hybrid representations need to be developed.

- Development of a framework in which process models of specific domains can be developed. Basic process calculi prove to be too cumbersome in themselves to develop large-scale applications. For example, what is the process calculus equivalent of the entity-relationship diagram?

- How can process calculi help in the development of query languages to information systems, where users can directly frame expressions in which events occur as first-class entities?

References

[1] T. Abraham and J. Roddick. Survey of spatio-temporal databases. *GeoInformatica*, 3:61–69, 1999.

[2] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26:832–843, 1983.

[3] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.

[4] J. F. Allen and G. Ferguson. Actions and events in interval temporal logic. Technical Report TR521, Computer Science Department, University of Rochester, 1994.

[5] H. Couclelis. People manipulate objects (but cultivate fields): beyond the raster-vector debate in GIS. In A.U. Frank and I. Campari, editors, *Spatial Information Theory: A Theoretical Basis for GIS*, number 716 in Lecture Notes in Computer Science, pages 65–77. Springer-Verlag, Berlin, 1993.

[6] D. Davidson. The individuation of events. In N. Rescher, editor, *Essays in Honor of Carl G. Hempel*, pages 216–234. Reidel, 1969.

[7] A. Galton. Fields and objects in space, time, and space-time. *Spatial Cognition and Computation*, 2004. forthcoming.

[8] P. Grenon and B. Smith. SNAP and SPAN: Towards dynamic spatial ontology. Forthcoming in the Journal of Spatial Cognition and Computation, 2004.

[9] T. Hägerstrand. What about people in regional science? *Papers of the Regional Science Association*, 24:7–21, 1970.

[10] W.R. Hamilton. Theory of conjugate functions or algebraic couples, with a preliminary and elementary essay on algebra as the science of pure time. *Trans. Royal Irish Academy*, 17(1):293–422, 1837.

[11] S.W. Hawking. *A Brief History of Time: From the Big Bang to Black Holes*. Bantam Books, Toronto, 1988.

[12] M. Heller. *The Ontology of Physical Objects: Four-dimensional Hunks of Matter*. Cambridge Studies in Philosophy. Cambridge University Press, UK, 1990.

[13] M. Hennessy. *An Algebraic Theory of Processes*. Boston: MIT Press, 1988.

[14] C. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.

[15] K. Hornsby and M.J. Egenhofer. Identity-based change: A foundation for spatio-temporal knowledge representation. *International Journal of Geographical Information Science*, 14(3):207–204, 2000.

[16] J. Kim. Events as property exemplifications. In *Action Theory*, pages 159–177. Reidel, Dordrecht, 1976.

[17] R.A. Kowalski and M.J. Sergot. A logic-based calculus of events. *New Generation Computing*, 4:67–95, 1986.

[18] C. Marshall. Historic USGS topographic maps. http://historical.maptech.com/index.cfm, 2003.

[19] J.M. McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Reading in Artificial Intelligence*, pages 431–450. Tioga Publishing Co., Palo Alto, CA, 1969.

[20] H.J. Miller. What about people in geographic information science? In D. Unwin, editor, *Re-Presenting Geographic Information Systems*. Wiley, New Jersey, 2003. In press.

[21] R. Miller and M. Shanahan. The event calculus in classical logic – alternative axiomisations. *Linköping Electronic Articles in Computer and Information Science*, 4, 1999. http://www.ep.liu.se/ea/cis/1999/016.

[22] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer Verlag, 1980.

[23] R. Milner. Elements of interaction(Turing Award Lecture). *Comm. ACM*, 36:78–89, 1993.

[24] R. Milner. *Communicating and Mobile Systems: The π-calculus*. Cambridge University Press, 1999.

[25] A. Mourelatos. Events, processes and states. *Linguistics and Philosophy*, 2:415–434, 1978.

[26] C. A. Petri. Fundamentals of a theory of asynchronous flows. In *Proc. IFIP (Amsterdam '62)*, pages 386–390. North Holland, 1963.

[27] F. Pianesi and A. Varzi. Events and event talk: An introduction. In J. Higginbottom, F. Pianesi, and A. Varzi, editors, *Speaking of Events*, pages 3–48. Oxford University Press, Oxford, UK, 2000.

[28] A. N. Prior. *Time and Modality*. Oxford: Clarendon Press, 1957.

[29] W.V.O. Quine. *Word and Object*. MIT Press, Cambridge, MA, 1960.

[30] Y. Shoham and N. Goyal. Representing time and action in artificial intelligence. *Frontiers of Artificial Intelligence*, 1988.

[31] R. Snodgrass, editor. *The TSQL2 Temporal Query Language*. Kluwer Academic Publishers, 1995.

[32] C. Stirling. *Modal and Temporal Properties of Processes*. Texts in Computer Science. Springer, 2001.

[33] Z. Vendler. Verbs and times. In *Linguistics in Philosophy*, chapter 4, pages 97–121. Ithaca, New York, 1967.

[34] M.F. Worboys. Object-oriented approaches to geo-referenced information. *International Journal of Geographical Information Systems*, 8(4):385–399, 1994.

[35] M.F. Worboys. A unified model of spatial and temporal information. *Computer Journal*, 37(1):26–34, 1994.

[36] M.F. Worboys, H. M. Hearnshaw, and D. J. Maguire. Object-oriented data modelling for spatial databases. *International Journal of Geographical Information Systems*, 4(4):369–383, 1990.

[37] M. Yuan. Representing complex geographic phenomena with both object- and field-like properties. *Cartography and Geographic Information Science*, 28(2):83–96, 2001.