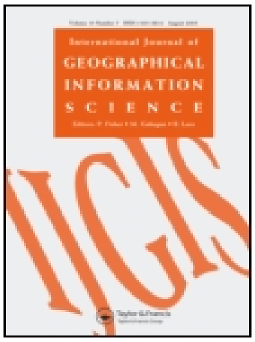

International Journal of Geographical Information Science

ISSN: 1365-8816 (Print) 1362-3087 (Online) Journal homepage: http://www.tandfonline.com/loi/tgis20

Event-oriented approaches to geographic phenomena

Michael Worboys

International Journal of Geographical Information Science
Vol. 19, No. 1, January 2005, 1–28

Taylor & Francis
Taylor & Francis Group

Research Article

Event-oriented approaches to geographic phenomena

MICHAEL WORBOYS
National Center for Geographic Information and Analysis, University of Maine, Orono, ME 04469, USA

(*Received 30 November 2003; in final form 29 April 2004*)

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 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. Up to now, most research 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 event-oriented theory of space and time, and suggests the possibilities that the theory provides by using it to represent the motion of a vehicle through a region. The paper contains a discussion of the scalability and applicability of this model to geographic domains and illustrates some of the ideas by reference to a geo-sensor example. The paper concludes by summarizing its main ideas, relating the research to other germane areas not covered in the developmental survey, and indicating directions for future work.

Keywords: Event-oriented approaches; Geographic domain; Geographic phenomena

1. Introduction

The title of this paper makes reference to two previous papers of the author and colleagues. Worboys *et al.* (1990) the object-oriented approach introduced and applied it to spatial data modelling. It has become clear that seeing the world as a collection of classified objects, with properties, definable behaviour, and relationships to each other, is an extremely useful approach to modeling. The theme was continued by Worboys (1994), 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 behaviour, and which are capable of relating to other entities. Of course, not every geographic phenomenon can usefully be viewed as a collection

*Email: worboys@spatial.maine.edu

ISSN 1365-8816 print/ISSN 1362-3087 online © 2005 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/13658810412331280167

of objects. The object-field dichotomy, discussed by Couclelis (1993), 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 (1970), emphasizing the importance of time in human activity, and is currently exemplified by the work of Miller (2004) on transportation and urban analysis, and Yuan (2001) 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 modelling 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 and an event-oriented view are required for a complete representation.

Our goal in providing approaches to representation and reasoning is to 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 perform these functions more effectively, we need representations, query languages, and techniques for reasoning, that explicitly cater for the event-oriented view. 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

Table 1. Relational view of a morning's activities.

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

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 event modelling into our conceptual modeling toolbox.

1.1 *A note on terminology: events or processes?*

This paper is about happenings in the world, as opposed to objects, their properties, locations, and other enduring entities. We will see later that a general term for these happenings is 'occurrents', as distinct from 'continuants'. Common speech, as well as the linguistic and philosophical literature, has several terms for occurrents, including *event, process*, and *action*, each with a different shade of meaning. For example, we might agree that photosynthesis is a process, whereas the typing by the author of this sentence is an event, although typing is a process. On further investigating these terms and the ways in which they are used, one finds an astonishing variety of usage and definition. One person's process is another's event, and vice versa. Because this work is neither philosophical nor linguistic in focus, we have simplified matters and in general used the words 'event' and 'occurrent' as an umbrella term for all kinds of occurrents. The only exception to our general use of the term 'event' is when another term is standard in a particular area of research. For example, computer scientists working on computational occurrences refer to these as processes, and so we use that term in that context. The point to emphasize is that we will not make any fine distinctions in this work between events and processes (but actions are distinguished as involving agents).

2. Stages in the development of spatiotemporal information systems

We begin by charting the recent history of dynamic geographic information models. This 'brief history of time' (with apologies to Steven Hawking 1988) 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*

Until now, the most common approach to spatiotemporal models 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 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 website; Marshall 2003.) The temporal sequence consists of three temporal snapshots, referenced to the years 1902, 1946, and 1955. It is clear that, over the passage of time, many changes have occurred, such as the construction of the airport and removal of one of the railway links. 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).

Stage One snapshot sequences are indexed by a temporal variable, and so the nature and structure of the underlying temporal reference domain influence 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 modelled 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 temporal queries is 'What was the state of this object at that time?' or its converse 'At what time did this object have

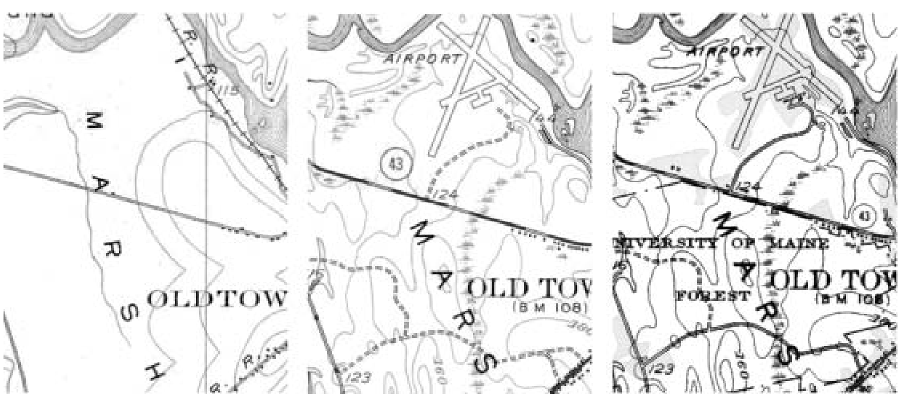

Figure 1. History of part of Old Town, Maine, recorded in snapshot at times 1902 (left), 1946 (centre), and 1955 (right).

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 (Worboys 1994, Snodgrass 1995, Abraham and Roddick 1999).

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 (2000) in a geospatial setting. 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 neighbourhood 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.

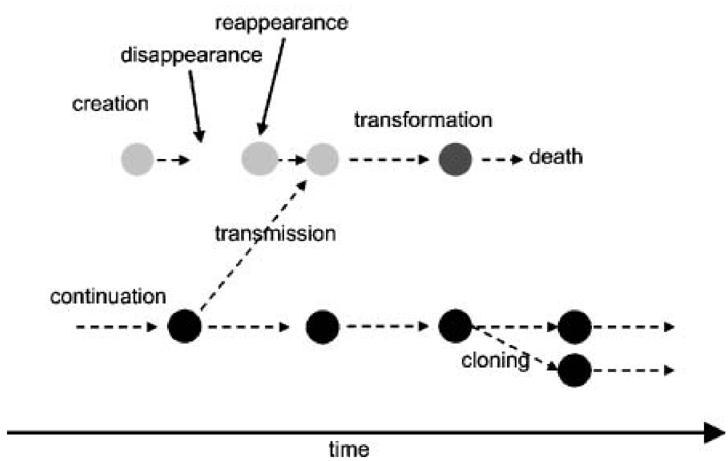

Figure 2. Object-change history.

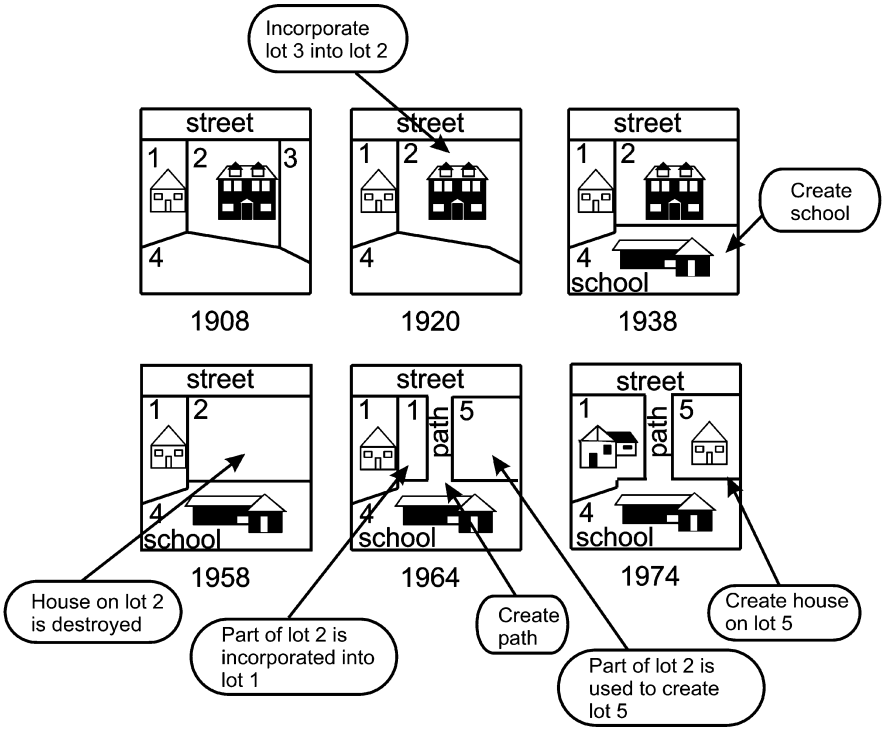

Figure 3. Neighbourhood evolution example: snapshot sequence.

- 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 toward its objective. Further problems arise from 'hybrid' examples, such as a wildfire or a spread of an infectious disease.

A natural model of the world is based on the evolution of objects through time, each of which retains its identity but changes its spatial and other attributes. However, problems arise, particularly related to continuity of identity through time. Here is an example of the kind of tangle that 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 by Heller (1990).

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 and action*

The final stage in this evolution is a full-blooded treatment of change, in terms of events and actions. Galton (in press) 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 occurrents, the ways in which objects may participate in them, and relationships between them.

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

An initial difficulty arises concerning the meaning of terms: as discussed in the introductory section, almost every account uses different definitions for event, process, and action, and we will generally use the umbrella terms 'event' and 'occurrent', unless this usage conflicts with the prevalent usage in the literature, as in computational processes.

Some broad distinctions can initially be made. Firstly, there is a distinction to be made about events 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 would not. Thirdly, 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', which name entities that can be counted, and mass nouns, such as 'water', which 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 Vendler (1967) and refined by Mourelatos (1978). In this taxonomy, occurrents are *events, accomplishments*, or *achievements*, that may be counted, or *processes* that may not.

The ontological status of events has been of interest to philosophers. Following Pianesi and Varzi (2000), 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 (1960), 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 (1969), 'Events have a unique position in the framework of causal relations between events in somewhat the same way that 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 (1976) 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. (This example shows the difficulty in precisely locating occurrents in space–time. Although we might agree that the vehicle slowing events overlap, we would not be able to say that they occupy *precisely* the same space-time regions.)

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

There are interesting modelling questions about the similarities between events and objects. Certainly, events may have instances (occurrences); attributes; belong to a subsumption hierarchy; have temporal parts; and have 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.

Table 2. Object-event similarities and differences.

Objects	Events
Object-event similarities	
Object instances	Event occurrences
Object attributes	Event attributes
Object taxonomy	Event taxonomy
Object partonomy	Event partonomy
Object relationships	Event relationships
Object-event differences	
Object endurance	Event perdurance

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 that are based upon logic. Formal theories of time go back at least as far as Hamilton (1837), 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 (1957) 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 the USA'; then $\Box p$ might be the proposition 'Necessarily, Washington is the capital of the 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 by McCarthy and Hayes (1969). An application domain is modelled as a collection of static, snapshot *situations*; hence, the situation calculus is basically a Stage Two approach. Each situation has a *state*, and *actions* change one situation to another. The changing situations are modelled using *fluents*, time-varying properties of the domain, which are expressed by propositions that evaluate to true or false, depending on the situation. The situation calculus is a *changebased* 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 Three 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 (1986) and is discussed in a more recent paper of Miller and Shanahan (1999). 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*. As with the situation calculus, the changing nature of situations is modelled by fluents. 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 before it has been initiated and after it has been terminated'. 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, 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 the USA' holds to be true during the year 2004. 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 (Shoham and Goyal 1988), among the ways that a temporal proposition P can be classified are the following:

- downward-hereditary: If P holds during an interval, then P holds during all subintervals of the interval. Example: 'The vehicle has moved less than five miles from its original station'.
- upward-hereditary: If P 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 5 mph'.
- 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'.

Allen (1983) 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 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 (1984) and Allen and Ferguson (1994) used interval temporal logic and the above calculus of temporal intervals to provide an extension of the event calculus of the preceding section. This gives quite a rich framework for representing and

Table 3. Allen's temporal interval relations.

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]

reasoning about occurrents, and is definitely a Stage Three approach. Occurrents now take place over time intervals and can have relationships dependent on their interval relationships. This allows each occurrent to have a rich internal structure. In a similar way to the event calculus, the predicate `holds(property, time)` is introduced, which 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 occurrent happened over a time interval. Allen uses this formulation to make a distinction between events and processes. Events, according to Allen, satisfy formula (5), in that their occurrence over time interval t implies their non-occurrence over any proper sub-interval of t.

$$(\texttt{occur}(e, t) \wedge \texttt{in}(t', t) \Rightarrow \neg \texttt{occur}(e, t')) \tag{5}$$

Processes and events are taken to be subtypes of the more general type, occurrence. A new predicate, `occurring(process, time)`, is introduced, and any process r must satisfy formula (6).

$$\texttt{occurring}(r, t) \Rightarrow \exists t' ((\texttt{in}(t', t) \wedge \texttt{occurring}(r, t')) \tag{6}$$

(One of the referees notes that Allen's distinction between events and processes is problematic. For example, an occurrence would satisfy formula (6) if it occurs on the intervals: [0, 1], [0.9, 1], [0.99, 1], …)

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. (Again, there are difficulties. For example, are we to suppose that the action and the resultant event occupy precisely the same temporal intervals?) A fuller axiom set is provided by Allen (1984).

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 amount of literature describing this strategy, traceable back at least to Petri's work on asynchronous flow (Petri 1963). Later work on the algebraic approach has been presented by Milner (1980), Hoare (1985), and Hennessy (1988), and Stirling (2001). Following this literature, we will use the term *process* for computational occurrents. The motivations for this work are well expressed in Milner's Turing Award lecture (Milner 1993), 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 modelled, 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.

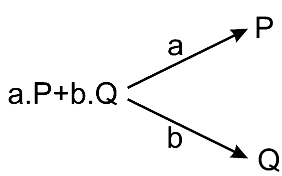

Figure 4. Transition diagram for summation.

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 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, 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 $i\{1, 2, 3, 4\}$. 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 $\overline{b}$ 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 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 provides a transition between traffic situations at the intersection. Process $X_{ij\ldots}$ can proceed when 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

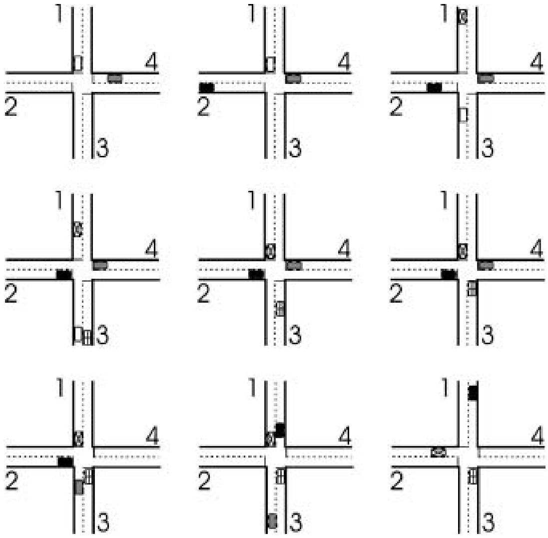

Figure 5. Four-way stop example.

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 $\overline{a}$ results 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 (1999) development of the π-calculus 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.

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 $a(v)$ 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:

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

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

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 which means 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 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, Machine(x), which takes the money x and, if it is the correct amount, executes CorrectMoney. The process Light goes green and reacts with ThroughToll(m) to a go, provided that the correct money has been inserted into the machine. The equations defining these processes are:

$$\mathsf{ThroughToll}(m) = \overline{\mathsf{Pay}(m)}.\mathsf{green}.\mathsf{Proceed}$$

$$\mathsf{Machine}(x) = \mathsf{Pay}(x).x$$

$$\mathsf{CorrectMoney} = \overline{\mathsf{go}}.\mathsf{Machine}(x)$$

$$\mathsf{Light} = \mathsf{go}.\overline{\mathsf{green}}.\mathsf{Light}$$

The situation begins with the following concurrent processes:

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

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

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

The processes CorrectMoney and Light can now react through the go channel, leading to:

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

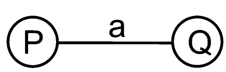

Figure 6. Processes P, Q, and their communication channel, a.

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 and events

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 (Hoare 1985; and others Hennessy 1988; Stirling 2001) 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 event-oriented model of the world: *everything is event*. To develop this idea, we now show how spatial and temporal reference frames can be modelled as processes in the process calculus. As we are now primarily concerned with the geographic world, we shall where possible move away from the term 'process' for an occurrent.

5.1 *Clocks, time, and temporal entities*

We begin by constructing a event-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 occurrent linked to the advancement of time. It is quite natural to link time to the occurrence of ticking, and we take the 'tick' occurrents as atomic and define the temporal domain (assumed linear) as a sequence of ticks. In our pure event-oriented approach, time *is* the sequence of ticking occurrents.

There is a choice as to whether to present time as a single tick occurrent that recursively 'calls' itself or as a collection of separate but sequenced tick events. We choose the latter and to this end construct a collection of events, $\mathsf{Tick}_1, \ldots, \mathsf{Tick}_n$. 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 occurrent, Tick_i, has a channel tocc_i that is capable of being occupied by temporally referenced entities. Any configuration of Tick occurrents 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 occurrent

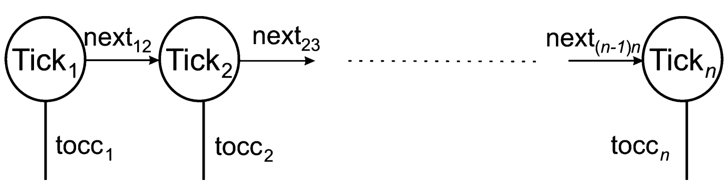

Figure 7. Events representing linear time by means of a clock.

Clock_n consists of the concurrent composition of n linked tick events, 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 occurrents. Of cowrse, we cowld 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 occurrents in the same clock. In *asynchronous* domain models, temporally referenced entities may handshake with occurrents in different clocks. Another feature to note about clocks is that there is no in-built notion of regularity in the timekeeping. Each tick of the basic clock occupies a duration that is unspecified. Clock regularity can of cowrse be built in as an extra feature or assumption.

5.2 *Locations, space, and spatial entities*

It is natural to represent time as an occurrent, or collection of occurrents, as we measure time using events such as the ticking of clocks. An event-oriented view of space requires more of a conceptual leap. However, the measurement and communication of location also require occurrents. (A collection of location-aware mobile phones provides an example of an event-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 neighbours by a directed adjacency relation. The idea is to represent each location as an occurrent that handshakes with its neighbours through its adjacency relations and has the capability of handshaking with an occupying entity, itself represented by an occurrent.

More precisely, construct a collection of location occurrents, Loc_1, …, Loc_n. Two adjacent locations, Loc_i and Loc_j, share the common channels, adj_{ij} and adj_{ji}. The two directional boundaries are modelled explicitly, thus providing more modelling flexibility (e.g. the ability to model one-way-only constraints).

In a similar way to the construction of temporal occurrents, location occurrents 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 geosensor location occurrent might communicate to neighbouring 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 an event-oriented representation of a region, Region_5, divided into locations 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 occurrent. 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 occurrents in the same region. In *asyntopic* domain models, spatially referenced entities may handshake with occurrents 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 modelled using structured collections of ST-entities.

6. Case study: Motion

To illustrate the ideas of the previous section, we show how motion can be modelled using the process calculus 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, only three times, Tick_1, Tick_2, and Tick_3, are modelled as Clock_3. The spatial region is Region_5 used in the previous section, consisting of five locations, Loc_1, …, 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 occurrents model is given below in equations (7)–(15). The complete model is the concurrent composition Motion of all the constituent occurrents, defined by:

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

$$\mathsf{Tick}_1 = \mathsf{tstart}_1.\mathsf{tocc}_1.\overline{\mathsf{next}}_{12}.\mathsf{Tick}_1 \tag{7}$$

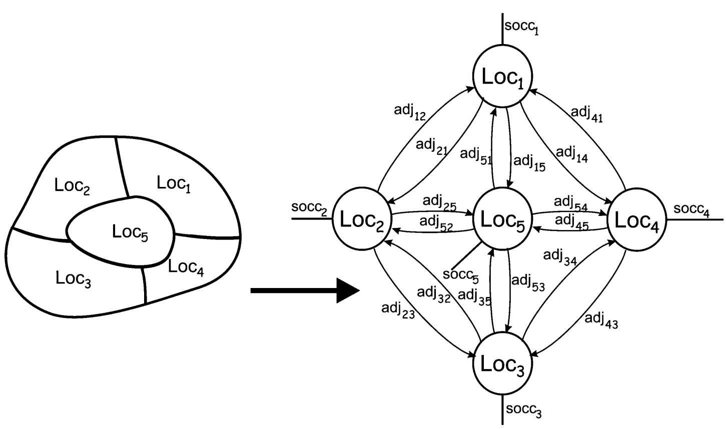

Figure 8. Occurrents representing locations in a spatial region.

Table 4. Spatiotemporal trajectory of the vehicle.

Location	Time
Loc_1	Tick_1
Loc_2	Tick_2
Loc_3	Tick_3

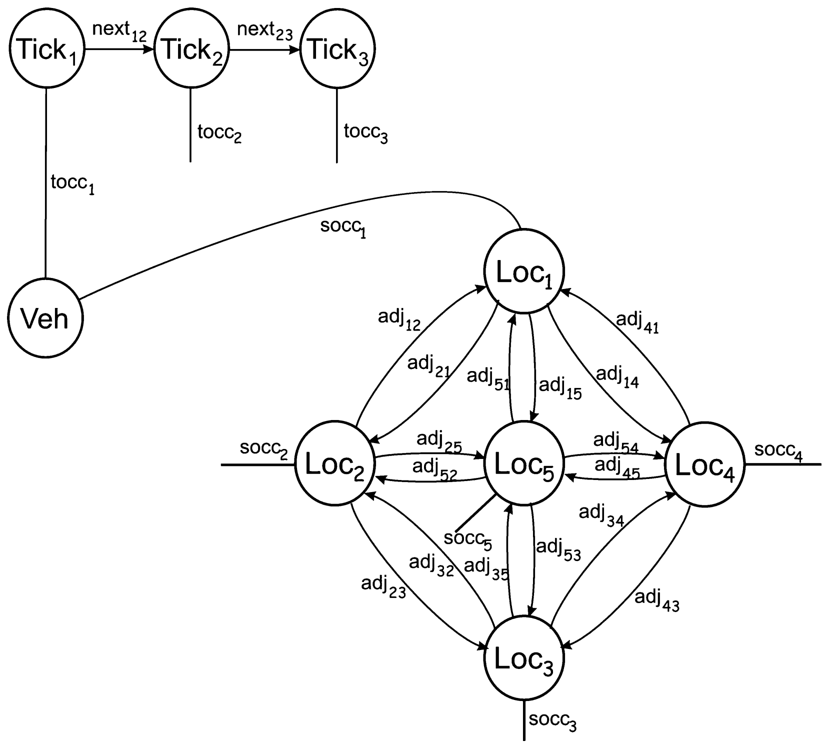

Figure 9. Initial stages of connection between occurrents for the motion example.

$$\mathsf{Tick}_2 = (\mathsf{tstart}_2 + \mathsf{next}_{12}).\mathsf{tocc}_2.\overline{\mathsf{next}}_{23}.\mathsf{Tick}_2 \tag{8}$$

$$\mathsf{Tick}_3 = (\mathsf{tstart}_3 + \mathsf{next}_{23}).\mathsf{tocc}_3.\mathsf{Tick}_3 \tag{9}$$

$$\begin{aligned}\mathsf{Loc}_1 = \\ &\left(\mathsf{sstart}_1 + \mathsf{adj}_{21} + \mathsf{adj}_{51} + \mathsf{adj}_{41}\right).\mathsf{socc}_1. \\ &\left(\overline{\mathsf{adj}}_{12} + \overline{\mathsf{adj}}_{15} + \overline{\mathsf{adj}}_{14}\right).\mathsf{Loc}_1\end{aligned} \tag{10}$$

$$\begin{aligned}\mathsf{Loc}_2 = \\ &\left(\mathsf{sstart}_2 + \mathsf{adj}_{12} + \mathsf{adj}_{52} + \mathsf{adj}_{32}\right).\mathsf{socc}_2. \\ &\left(\overline{\mathsf{adj}}_{21} + \overline{\mathsf{adj}}_{25} + \overline{\mathsf{adj}}_{23}\right).\mathsf{Loc}_2\end{aligned} \tag{11}$$

$$\begin{aligned}\mathsf{Loc}_3 = &\\ &\left(\mathsf{sstart}_3 + \mathsf{adj}_{23} + \mathsf{adj}_{53} + \mathsf{adj}_{43}\right).\mathsf{socc}_3.\\ &\left(\overline{\mathsf{adj}}_{32} + \overline{\mathsf{adj}}_{35} + \overline{\mathsf{adj}}_{34}\right).\mathsf{Loc}_3\end{aligned} \tag{12}$$

$$\begin{aligned}\mathsf{Loc}_4 = &\\ &\left(\mathsf{sstart}_4 + \mathsf{adj}_{14} + \mathsf{adj}_{34} + \mathsf{adj}_{54}\right).\mathsf{socc}_4.\\ &\left(\overline{\mathsf{adj}}_{41} + \overline{\mathsf{adj}}_{43} + \overline{\mathsf{adj}}_{45}\right).\mathsf{Loc}_4\end{aligned} \tag{13}$$

$$\begin{aligned}\mathsf{Loc}_5 = &\\ &\left(\mathsf{sstart}_5 + \mathsf{adj}_{15} + \mathsf{adj}_{25} + \mathsf{adj}_{35} + \mathsf{adj}_{45}\right).\mathsf{socc}_5.\\ &\left(\overline{\mathsf{adj}}_{51} + \overline{\mathsf{adj}}_{52} + \overline{\mathsf{adj}}_{53} + \overline{\mathsf{adj}}_{54}\right).\mathsf{Loc}_5\end{aligned} \tag{14}$$

$$\mathsf{Vehicle} = \overline{\mathsf{tstart}}_1.\overline{\mathsf{sstart}}_1.\overline{\mathsf{tocc}}_1.\overline{\mathsf{socc}}_1.\overline{\mathsf{tocc}}_2.\overline{\mathsf{socc}}_2.\overline{\mathsf{tocc}}_3.\overline{\mathsf{socc}}_3.0 \tag{15}$$

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

$$\overline{a} : P|a : Q \to b : P|c : Q$$

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

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

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

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

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

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

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

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

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

$$\overline{\mathsf{socc}}_2 : \mathsf{Vehicle}|\mathsf{socc}_2 : \mathsf{Loc}_2 \to \overline{\mathsf{tocc}}_3 : \mathsf{Vehicle}|\left(\overline{\mathsf{adj}}_{21} + \overline{\mathsf{adj}}_{25} + \overline{\mathsf{adj}}_{23}\right) : \mathsf{Loc}_2 \tag{I9}$$

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

$$\begin{aligned}&\left(\overline{\mathsf{adj}}_{21} + \overline{\mathsf{adj}}_{25} + \overline{\mathsf{adj}}_{23}\right) : \mathsf{Loc}_2|\left(\mathsf{sstart}_3 + \mathsf{adj}_{23} + \mathsf{adj}_{53} + \mathsf{adj}_{43}\right) : \mathsf{Loc}_3\\ &\rightarrow \left(\mathsf{sstart}_2 + \mathsf{adj}_{12} + \mathsf{adj}_{52} + \mathsf{adj}_{32}\right) : \mathsf{Loc}_2|\mathsf{socc}_3 : \mathsf{Loc}_3\end{aligned} \tag{I11}$$

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

The Vehicle occurrent has built-in control of the spatiotemporal trajectory, through the predetermined sequence of interactions with temporal and spatial occurrents, as given in table 4. The Motion occurrent begins with an interaction I1 between constituent occurrents 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, which 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, which establish the vehicle at a position in space.
- Temporal transitions: Interactions I5, I7, between neighbouring Tick occurrents using the next channels, which move the scenario on in time.
- Spatial transitions: Interactions I8, I11, between neighbouring Loc occurrents using the adj channels, which 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.

$$\begin{aligned}&\left(\overline{\mathsf{adj}}_{12} + \overline{\mathsf{adj}}_{15} + \overline{\mathsf{adj}}_{14}\right) : \mathsf{Loc}_1|\left(\mathsf{sstart}_4 + \mathsf{adj}_{14} + \mathsf{adj}_{34} + \mathsf{adj}_{54}\right) : \mathsf{Loc}_2\\ &\rightarrow \left(\mathsf{sstart}_1 + \mathsf{adj}_{21} + \mathsf{adj}_{51} + \mathsf{adj}_{41}\right) : \mathsf{Loc}_1|\mathsf{socc}_4 : \mathsf{Loc}_4\end{aligned} \tag{I8*}$$

As an extension, we can formalize *indeterminate* motion through the region, following only legal adjacencies in space and time. Equations (I1)–(I12) cowld 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 occurrent. 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. Event model specification, modularization, and geo-sensor example

In the introduction, we describe the development of this work as a continuation of earlier work on the foundations of the object-oriented approach to geospatial data modelling. This paper lays the foundations, developed in an algebraic framework, for an event-oriented approach that enables the modelling of dynamic geophenomena. While the focus of this paper is toward neither computational efficiency nor specific application domain models, it is reasonable for the reader to ask for some details on how we see the construction proceeding, once the foundations have been developed. This section discusses some of the construction methods and briefly considers an example based on geo-sensor technology.

Just as in the object-oriented approach everything is viewed as objects, so in our approach everything is viewed as occurrents. What brings power to object-orientation, apart from the naturalness of the model, is the richness of the abstraction techniques. As indicated in the Stage Three discussion, objects and events share many structural similarities for modelling purposes. We may distinguish event occurrences (e.g. the traffic incident in Bangor today at 3.00 p.m.) from event types (TrafficIncident). Event types can be arranged in a subsumption hierarchy; for example, event type TrafficIncident may subsume TrafficAccident. Events may have parts; for example, Serious- TrafficAccident may be composed of AccidentReport and EmergencyResponse. These, and other structural properties of the event-oriented approach, allow the possibility of modularization: large and complex events may be modelled in terms of smaller and simpler components. At the bottom of this chain are the atomic events, or literals, that correspond in object modelling to such things as line segments, polygons, numbers, and strings. The preceding section has been concerned with the development of some of these atomic event types, with the focus on geo-event modelling. We have developed simple but well-specified modules for temporal and spatial event types, and shown how such event types may be composed to model motion. In general a formal model allows a precise and unambiguous specification of a domain, and the event model developed here allows such specifications of dynamic domains.

The detailed specification of atomic event types can be developed using process algebraic formulations like those in earlier sections. This is similar to the development, using object-oriented models of space of formal specifications of point, line segment, and polygon. Just as in the object-oriented approach, such details may be hidden using encapsulation. Essentially, an event type has a hidden inner core and a set of links to other event types (these are provided by the handshaking links to other occurrents). So, applications may be built using prespecified atomic event modules, along with custom-built, domain-specific event modules. This is the mechanism by which scalability is attained.

Our example concerns the modelling, collection and analysis of data relating to the monitoring of microclimatic variations in an ecosystem. In line with near-future technology, we assume an information-gathering infrastructure consisting of large numbers of small, unattended, untethered, and collaborative sensor nodes that have non-renewable, low power supplies, and communicate via short-range radio frequency with neighbouring nodes. We assume that each sensor is equipped with a power supply, GPS component, communication module, sensing devices, and mini database system. Because of limitations on power supply, sensors need to be 'live' only when they are needed, for example when nearby sensors detect changes in conditions. The challenge of sensor networks is to aggregate sensor nodes into computational infrastructures that are able to produce globally meaningful information from raw local data obtained by individual sensor nodes. However, no global control is assumed. Instead, a hierarchical structure is imposed, where higher-level nodes control lower-level, subordinate sensors, which communicate data up the chain.

An initial task is to construct a computational model of the sensor network that captures its dynamic and spatial characteristics. For the purposes of this case study, we are concerned with a model that captures sensor coordination protocols and allows queries of an event-based nature to be made to the system. These are two distinct challenges; modelling the sensor system and modelling the dynamic aspects of microclimates. Modelling the system requires an event model of dynamic deployment protocols for the sensors, and the methods developed in the paper are well suited to this task. Atomic modules that represent the spatial region and a clock will be required. On top of these, protocols will be required for dynamic spatial configuration, hierarchical structure and communication, and responses to changes in environment and other processors.

In the case of the dynamic model of microclimatic conditions in the environment, scientific microclimate models will need to be expressed in the event-based formalism. A major issue will be discretization, because in general such models are continuous. This is the same kind of issue faced by the object model for static phenomena (e.g. soil models), and considerable work is needed in this area.

The reward for this work will be a well-specified, computational model of dynamic configurations of sensors responding to a dynamic environment. So, for example, queries to the system, naturally specified in event-based terms, do not have to be converted into artificial snapshot-based representations but can be handled by the system at the same level of representation in which they are framed.

8. Conclusions

This paper sets out some of the requirements 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 toward 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 an event-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 an event-oriented 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 occurrents, as well as spatially, temporally, and spatiotemporally referenced entities.

The research here continues the general motivation of the object-oriented approach, to bring computational representations closer to the conceptualizations of users. In the case of object-orientation, the model allows a richer view of the world, composed of compositions of complex objects. In the case of 'event-orientation', added richness is provided by aggregations of complex events. The paper has shown how such a dynamic model can be well founded, and how some of the techniques of object orientation can be carried over to allow event encapsulation, subsumption, abstraction, and aggregation. These constructions will allow the model to scale up to useful applications, and we gave hints in the previous section as to how this might take place.

8.1 *Relationship to other areas of research*

We have already dealt at length with connections between this work and logics of time, situations, and events. Also, the event-oriented model that we develop is based on the theory of computational process algebras, and these connections have been made earlier in the paper. This section briefly notes other linkages.

There is a close relationship between our work and work on cellular automata and intelligent agents. Process algebras can be thought of as a means of formally specifying the actions of configurations of interacting automata. A cellular automaton is a special kind of automaton that is situated in a cellular structure, such as a hexagon within a honeycomb grid of hexagons. There is a close relation between the spatial occurrents, as described in the motion example, and cellular automata. However, the constructions in this paper are more general, because the handshakes with spatial occurrents can be less constrained than in cellular automata. Cellular automata have been applied in several areas in GI science, including urban system modelling (O'Sullivan and Torrens 2000), transportation (Nagel *et al.* 1998), and population dynamics (Benenson 1998).

Although, for the sake of simplicity, we adopt a stance in which all entities are viewed as occurrents, in actuality many entities may be better viewed as having continuant properties, while nevertheless relating closely to events. In the traffic protocol, vehicle motion, and the geo-sensors examples, the vehicles and sensors can be viewed as agents, the behaviour of which is described by the event specification language developed here. Computer-based agents function autonomously, interact with one another and with humans by means of agent-communication languages, and respond to changes in their environment through definable protocols (Woolridge and Jennings 1995, Jennings 2000). Agents have the capacity to be mobile in their environments. Agent-based approaches are closely related to automata and have been researched in the context of GI science (Haklay *et al.* 2001).

Another area that relates to this work is that of finite-element methods (Akin 1994). Finite-element analysis is the application of finite elements to the analysis of static or dynamic physical objects and systems. Finite-element methods represent the dynamic system as a collection of linked, discrete regions—the *finite elements*. As already discussed, one of the bases of the approach discussed here is discretization. Continuous event models are discretized into a finite collection of occurrent entities. Although, on the face of it, there should be a close connection between finite-element analysis and analysis of dynamic geographic domains, the fact is that there has been little interaction between these fields of research. Finite-element analysis has seen its greatest successes in the analysis of mechanical systems. There is fertile ground for future research here.

8.2 *Limitations and further work*

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 occurrents 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 event models can be viewed and manipulated. This last point raises the issue of discreteness; the model proposed here is discrete, whereas many aspects of the world are continuous. However, many successful computational models are discrete approximations to continuous entities. For a successful discrete model, one has only to look at the impact of line segments and polygonal objects on object models of geographic space.

We envision applications for event models of dynamic geographic space–time that allow representation of coordination and control semantics, for example, in transportation, coordination of relief and rescue resources, and defence. However, before such applications can be realized, further work needs to be done. Some immediate follow-ups to work described in this paper include the following:

- Characterization of the role of objects in a mixed object–event model. Relationships to cellular automata and multi-agent system research.
- Development of richer models of movement, based on the motion example in the text. Examples of possible extension include more than one vehicle (including causal interactions and development of location occupation constraints) and more than one clock.
- Development of a theory of granularity and discretization for occurrent entities.
- Development of a framework in which event models of specific domains can be developed. Models of event abstraction, encapsulation, subsumption, and composition.
- Investigation of the role of the event-oriented approach in the development of query languages to information systems, where users can directly frame expressions in which events occur as first-class entities.
- Investigation of computational performance issues associated with event-oriented modelling.

Acknowledgements

This work was supported by the National Science Foundation under NSF grant numbers EPS-9983432 and BCS-0327615. The author is also grateful to the Ordnance Survey of Great Britain and to the United Kingdom Engineering and Physical Sciences Research cowncil for their support for spatiotemporal research. The work has greatly benefited from discussions with Matt Duckham, Max Egenhofer, Kathleen Hornsby, Lars Kulik, Barry Smith, and Rui Zhang. The author is also grateful for the insightful comments and advice provided by the anonymous referees.

References

ABRAHAM, T. and RODDICK, J., 1999, Survey of spatio-temporal databases. *GeoInformatica*, **3**, pp. 61–69.

AKIN, J.E., 1994, *Finite Elements for Analysis and Design* (San Diego, CA: Academic Press).

ALLEN, J.F., 1983, Maintaining knowledge about temporal intervals. *Communications of the ACM*, **26**, pp. 832–843.

ALLEN, J.F., 1984, Towards a general theory of action and time. *Artificial Intelligence*, **23**, pp. 123–154.

ALLEN, J.F. and FERGUSON, G., 1994, *Actions and Events in Interval Temporal Logic*, Technical Report TR521, Computer Science Department, University of Rochester.

BENENSON, I., 1998, Multi-agent simulations of residential dynamics in the city. *Computers, Environment and Urban Systems*, **22**(1), pp. 25–42.

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

DAVIDSON, D., 1969, The individuation of events. In *Essays in Honor of Carl G. Hempel*, N. Rescher (Ed.), pp. 216–234 (Dordrecht: Reidel).

GALTON, A., in press, Fields and objects in space, time, and space–time. *Spatial Cognition and Computation*.

GRENON, P. and SMITH, B., in press, SNAP and SPAN: Towards dynamic spatial ontology. *Spatial Cognition and Computation*.

HÄGERSTRAND, T., 1970, What about people in regional science? *Papers of the Regional Science Association*, **24**, pp. 7–21.

HAKLAY, M., O'SULLIVAN, D. and THURSTAIN-GOODWIN, M., 2001, So go downtown: Simulating pedestrian movement in town centres. *Environment and Planning B: Planning and Design*, **28**, pp. 343–359.

HAMILTON, W.R., 1837, Theory of conjugate functions or algebraic cowples, with a preliminary and elementary essay on algebra as the science of pure time. *Transaction of the Royal Irish Academy*, **17**(1), pp. 293–422.

HAWKING, S.W., 1988, *A Brief History of Time: From the Big Bang to Black Holes* (Toronto: Bantam Books).

HELLER, M., 1990, The Ontology of Physical Objects: Four-Dimensional Hunks of Matter. *Cambridge Studies in Philosophy* (Cambridge: Cambridge University Press).

HENNESSY, M., 1988, *An Algebraic Theory of Processes* (Boston, MA: MIT Press).

HOARE, C., 1985, *Communicating Sequential Processes* (Englewood Cliffs, NJ: Prentice-Hall).

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

JENNINGS, N.R., 2000, On agent-based software engineering. *Artificial Intelligence*, **117**, pp. 277–296.

KIM, J., 1976, Events as property exemplifications. In *Action Theory* (Dordrecht: Reidel), pp. 159–177.

KOWALSKI, R.A. and SERGOT, M.J., 1986, A logic-based calculus of events. *New Generation Computing*, **4**, pp. 67–95.

MARSHALL, C., 2003, Historic USGS topographic maps. http://historical.maptech.com/index.cfm.

MCCARTHY, J.M. and HAYES, P.J., 1969, Some philosophical problems from the standpoint of artificial intelligence. In *Reading in Artificial Intelligence* (Palo Alto, CA: Tioga), pp. 431–450.

MILLER, H.J., 2004, What about people in geographic information science? In *Re-Presenting Geographic Information Systems*, P. Fisher and D. Unwin (Eds.) (New Jersey: Wiley).

MILLER, R. and SHANAHAN, M., 1999, The event calculus in classical logic—alternative axiomisations. *Linköping Electronic Articles in Computer and Information Science*, 4 http://www.ep.liu.se/ea/cis/1999/016.

MILNER, R., 1980, A Calculus of Communicating Systems, volume 92 of *Lecture Notes in Computer Science* (Berlin: Springer).

MILNER, R., 1993, Elements of interaction (Turing Award Lecture). *Communications of the ACM*, **36**, pp. 78–89.

MILNER, R., 1999, *Communicating and Mobile Systems: The π-calculus* (Cambridge: Cambridge University Press).

MOURELATOS, A., 1978, Events, processes and states. *Linguistics and Philosophy*, **2**, pp. 415–434.

NAGEL, K., WOLF, D.E., WAGNER, P. and SIMON, P., 1998, Two-lane traffic rules for cellular automata: A systematic approach. *Physical Review E*, **58**(2), pp. 1425–1437.

O'SULLIVAN, D. and TORRENS, P.M., 2000, Cellular models of urban systems. In *Theoretical and Practical Issues of Cellular Automata*, S. Bandini and T. Worsch (Eds.), pp. 108–117 (Berlin: Springer).

PETRI, C.A., 1963, Fundamentals of a theory of asynchronous flows. In *Proceedings of IFIP (Amsterdam '62)* (Amsterdam: North Holland), pp. 386–390.

PIANESI, F. and VARZI, A., 2000, Events and event talk: An introduction. In *Speaking of Events*, J. Higginbottom, F. Pianesi and A. Varzi (Eds.), pp. 3–48 (Oxford: Oxford University Press).

PRIOR, A.N., 1957, *Time and Modality* (Oxford: Clarendon Press).

QUINE, W.V.O., 1960, *Word and Object* (Cambridge, MA: MIT Press).

SHOHAM, Y. and GOYAL, N., 1988, Representing time and action in artificial intelligence. *Frontiers of Artificial Intelligence* (Boston: MIT Press).

SNODGRASS, R, editor., 1995, *The TSQL2 Temporal Query Language* (Dordrecht: Kluwer Academic).

STIRLING, C., 2001, Modal and Temporal Properties of Processes. *Texts in Computer Science* (Berlin: Springer).

VENDLER, Z., 1967, Verbs and times. In *Linguistics in Philosophy*, chapter 4 (Ithaca: Cornell University Press), pp. 97–121.

WOOLRIDGE, M. and JENNINGS, N.R., 1995, Intelligent agents: theory and practice. *Knowledge Engineering Review*, **10**(2), pp. 115–152.

WORBOYS, M.F., 1994, Object-oriented approaches to geo-referenced information. *International Journal of Geographical Information Systems*, **8**(4), pp. 385–399.

WORBOYS, M.F., 1994b, A unified model of spatial and temporal information. *Computer Journal*, **37**(1), pp. 26–34.

WORBOYS, M.F., HEARNSHAW, H.M. and MAGUIRE, D.J., 1990, Object-oriented data modelling for spatial databases. *International Journal of Geographical Information Systems*, **4**(4), pp. 369–383.

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