A principle of formal construction and computation of spatial event information

Simon Scheider, Martin Kiesow, Anusuriya Devaraju, Krzysztof Janowicz, Henry Michels, and Werner Kuhn

Abstract—Observations produced by environmental sensors and humans signify naturally occurring events. While there is technology available for inferring events from sensor streams, knowledge about *event inference rules and event construction principles* is often hidden in application logic or needs to be informally specified by domain experts. *Semantic formalisms* support description of spatial event inference procedures and provide a way to generalize them across implementation frameworks. Furthermore, there is a lack of algorithms which are flexible enough to capture both *spatial and in-situ event* inference, where spatial events extend beyond a single in-situ sensor. In this paper, we demonstrate how spatial events can be formally specified as bounded wholes connected by a process simulator. This *formal blue print* can be used to describe different event inference methods. Unlike simple event rules, the approach accounts for recursive conditions of *spatial and temporal identity* of events, e.g., lag times, and allows inference of event completeness. We propose corresponding event inference algorithms that can be used to compute process graphs and to generate and publish events as RDF. We evaluate our approach using an officially published blizzard data set.

Index Terms—observations, event inference, spatial events, identity conditions, formalism

1 MOTIVATION

TEMPORALLY indexed data is fed into the Web by a growing number of technical and human sensors [1], [2]. Due to the size and distributed nature of the Web [3], such data is hard to explore and exploit for analysis. A key challenge is to infer higher-level knowledge about events from sensor observations, for data-intensive science [3] and for organizing knowledge on the Web. Information about events serves to detect and understand short-term as well as long-term environmental impacts [4]. However, in an open technical environment like the Web, meta-data about event information is a prerequisite to its further use [3].

Recent research efforts have been devoted to developing processing standards and technologies for complex event inference from sensor streams [5], [6]. However, event inference requires not only efficient computing methods, but also knowledge about *event inference rules* and *event construction principles*. The latter is often assumed to be supplied informally by domain experts [7]. The wealth of event literature suggests that event models can range from simple rule patterns to statistical procedures [8], [9]. Furthermore, while mainstream work on events has mainly focused on stationary time series [8], only few approaches have dealt so far with identifying complex *spatial events* [10], [11], [12]. Unfortunately, there is no *general formal blue print* that would allow a knowledge engineer to capture and specify the knowledge contained in these different approaches in order to share and regenerate diverse inference models.

In this paper, we argue that the procedure of (spatial) event abstraction needs to be better understood and *formally described*. This allows to capture the various ways of abstracting events from observations in terms of *event identity conditions*, and to design corresponding event identification algorithms. As we will show subsequently, existing approaches either leave the issue of event identity unsettled, or implicitly make rather restrictive assumptions about event identity. Our approach allows a generalization of event inference procedures, which are momentarily bound to the expressivity of particular event inference languages, such as Esper [6].

Consider, for example[1], the task of inferring weather events, such as a blizzard, from surface and weather observations[2], as depicted in Table 1. Weather agencies provide official (but informal) definitions of many meteorological events, e.g., wind storm, flash freeze, hurricane, snow squall, and tropical storm[3].

Blizzards may be defined [14] in terms of an *event ontology*. However, event ontology patterns such as [15], [16] remain silent about how events should be identified in terms of observations.

- *S. Scheider, M. Kiesow, H. Michels and W. Kuhn are with the Institut für Geoinformatik, University of Münster, Germany.*
- *A. Devaraju is with the Forschungszentrum Jülich, Germany.*
- *K. Janowicz is with the University of California, Santa Barbara, USA.*

1. Example is taken from [13].
2. http://climate.weatheroffice.gc.ca/climateData/
3. http://climate.weather.gc.ca/glossary_e.html

TABLE 1
Sample observation records from Winnipeg weather station.

time (h)	7	8	9	10	11	12	13	14
windspeed (km/h)	38	40	45	43	35	38	40	35
visibility (km)	1	0.3	0.2	0.2	0.5	0.5	0.2	0.5
condition (s=Snow; bs=Blowing snow; c=cloudy)	s	bs	bs	bs	bs	bs	bs	c

To address event identification, *complex event processing* was established as a research field [8]. Provenance information is provided by a sensor, which may be hooked into the Sensor Web and described by semantic sensor specifications [1], [17]. With these specifications, one can develop *sensor-specific inference rules* [7] which capture part of the informal definitions provided by weather agencies. For example, blizzard detection requires that there is an *anemometer* measuring windspeed of at least 40 km/h and a number of further conditions to be met, including a visibility threshold and the presence of blowing snow. Hence, one could use *rules* of the following kind (over time series moments t) in order to infer events,

$$(\exists e.Blizzard(e)) \leftarrow windspeed(t) \geq 40km/h \wedge$$
$$visibility(t) \leq 1km \ \wedge BlowingSnow(t)...$$

, either in the form of deductions (from right to left), as in [7], or as abductions (from left to right), as in [18].

However, such sensor rules still fall short of providing sufficient *identity conditions of a spatial event*:

1) How can we know that two single observations generated by a sensor, such as conditions at 8:00 and conditions at 13:00 in Table 1, which both satisfy a rule pattern such as the one above, refer to the same blizzard event and not to two consecutive ones? Note that in Table 1, the conditions specified by the rule are not satisfied between 11:00 to 12:00, however, meteorologists would nevertheless consider both intervals as parts of the same event [14]. This is the problem of *temporal identity* of an event, and it cannot be solved by a sensor rule of the kind above[4].
2) How can we identify a single event across different sensing locations? A blizzard is not point-like, but an extended spatial phenomenon, and, thus, a *geographic event* [12].
3) How can we know that an event starts or ends? Does the blizzard that seems to start at 8:00 really end at 13:00? This is the question how *temporal event boundaries* can be formalized.

As it turns out, in meteorological practice [14], [11], all these questions are given practical answers by the specific way *how event information is constructed from observations*.

In this paper, our goal is to generalize this idea, i.e., to devise a formalism that allows to specify spatio-temporal identity conditions for events and to reason with them, as well as a corresponding abstraction algorithm for generating spatial events. The contribution of the paper is threefold:

1) A formal specification of *event identity conditions* and event characteristics (e.g. boundary conditions) (Sections 4, 5 and 6).
2) A suggestion of a corresponding layered architecture (compare Section 3) which captures these conditions.
3) A suggestion of corresponding computational algorithms for event inference, as well as a prototypical implementation and an evaluation (Section 7).

Our language is first-order predicate logic (FOL) with a standard interpretation into acts of focusing attention and predicating phenomena. We show how formal event properties follow by construction in Section 5.3. We reason with the formalism on blizzard events in Section 6. In the next section, we discuss existing approaches with respect to event identity conditions.

2 EVENT INFERENCE APPROACHES

There is a wealth of literature and technology available on event inference. Recent research efforts in computer science have focused on developing processing standards and technology for complex event inference from sensor streams [5], [6], [19]. These approaches regard events as information entities triggered by rules on sensor streams, where so called "event patterns", i.e., conjunctive queries, are executed over a temporal window of a sensor stream, such that every pattern match triggers the creation of an event instance. Particular challenges addressed by these approaches are the creation of time-based data windows from streams, and the efficiency of pattern matching and event joining[5]. General event identity critera, however, are not covered by this technology. For example, spatial event inference procedures, as proposed by [20], [21], [11], [9], draw on neighborhood sensitive identification methods ranging from image processing to hierarchical clustering.

Event inference is not only a computational, but also a conceptual challenge. During the last few decades, a number of authors have provided formal ontological accounts of events [22]. On a fundamental level, one can distinguish among *occurrence types* such as events and processes [22], mirroring the dyad of

4. That is, by Horn rules without recursion.

5. Compare, e.g., the "Esper" query language [6].

things and stuff [23], and reflecting formal properties such as homogeneity, boundedness, and cumulativity. However, there are disagreements about ontological commitments to temporal entities. In philosophy, commitments range from the denial of temporal entities ("three-dimensionalism") to the claim that everything that exists is a process [24]. In practice, general-purpose event ontologies do not seem to pay off [5]. Simple *ontology patterns*, such as [15], [16], can provide useful building blocks for specific purposes, but fail to provide any event identity criteria or construction principles.

Information about events is usually considered on a higher level of abstraction than observations, which suggests that it should be treated as a *formal abstraction* of the latter. In cognitive science, there is evidence that humans abstract complex events based on applying event schemas to discrete events, which are in turn a result of spontaneous segmentation of a continuous flow of perception [25]. From a logical viewpoint, this poses the question what makes an event an event, and whether there are general *identification procedures* for events which distinguish them from other categories. The latter would be urgently needed for automatic event inference [11], [5].

There are ontological attempts to come to grips with human observers and technical sensors [17]. Related to this work, we have proposed a process-centric view on sensors in [26]. There are also attempts to infer events as abstractions from sensor observations. The approach in [7] bases events on threshold patterns over particular sensor readings. [27] propose a formal method of sensor abstraction based on abductive reasoning. [28] suggest to detect geographical processes based on formalizing homogeneous spatial change, such as deforestation. All these authors do not address complex event identity conditions such as discussed in Section 1.

We have proposed an operational constructive approach towards the semantics of spatial information in [29]. The approach suggests operations and procedures as a resource of data semantics instead of ontological claims. Events are considered results of a formal construction from observation procedures, as envisioned in [30]. This allows to precisely specify their identity criteria.

3 Logical event construction

The challenge of a constructive approach towards information is where to start with, i.e., where to *ground*, the construction of information entities. We have argued in [29] that a useful starting level are *reproducible observation procedures*, i.e., *standardized operational schemes* for acting in an environment which can be shared by observers and allow them to establish *semantic reference*[6]. We will discuss a number of these operations in the following. At the same time, we will introduce corresponding formal primitives in FOL. In the remainder, all free variables (denoted by lower-case letters) in formulae are universally quantified. Sometimes, we explicitly use *predicate wildcards* in axiom schemata (in square brackets) indicating how axioms with a more specific purpose may be generated.

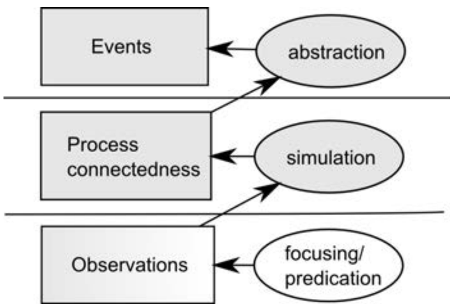


Fig. 1. Logical event abstraction schema underlying our approach. Circles denote procedures, and boxes denote resulting logic as subset of a FOL theory.

The approach taken is to introduce several logical abstraction layers in terms of the procedures that generate them, as depicted in Figure 1. The lowest level is the level of observations, which denotes results of human or sensor focusing and predication. The next layer contains process connections, which denote results of process simulators. The third layer contains events and their boundaries, which are entities logically abstracted from connectedness of processes.

4 Procedures for experiencing events and processes

Whenever one experiences a phenomenon in one's environment, one may utter a *here-now* (deictic) statement, such as "this is a tree" or "here and there continues the same precipitation event". Following [32], we call such basic kinds of acts *predications*. Predications can be understood by other observers because they have learned to perform the same predications, and because they are able to *follow the attention* of a speaker. As we will argue subsequently, we can understand data from technical sensors in an analogous way, based on regarding technical sensing as an extension of human attention.

4.1 Focusing attention and predicating the flow of time

What is the *domain of discourse* of our event reference theory? What are the entities over which we can predicate temporal phenomena? On the most fundamental

6. In [29], we argue that this is a basis for (semantic) reference systems [31], i.e., formal theories that provide reference to observable phenomena such as location, time, objects and qualities.

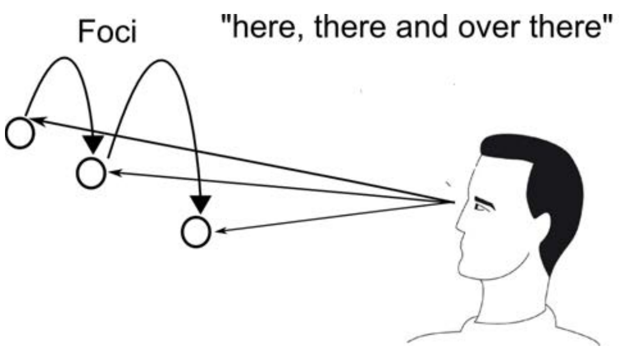


Fig. 2. Foci of attention are produced by shifting attention. They are granules in space as well as time. At the same time, they are lowest level information items on which phenomena can be predicated.

level, our domain of discourse contains a finite set of memorized acts of focusing human attention on some signal in the perceived environment [29], [33]. From the perspective of an observer, human attention is *granular*[7].

These granules are called *foci of attention* and denoted by the unary predicate F. Since foci are sharable among observers by joining attention, they can be conceived as lowest level *information items*[8]. In human attention, every "now" is also a "here" or "there", i.e., *space and time are inseparably interwoven*. Therefore, foci can be used as a basis for referring to measurable space as well as time. As [36] argues, foci of attention provide a non-circular account of time, because experiences of *succession* and *duration* are not attributed to some unobservable physical flow of time, but regarded as results of attentional activity.

Experiencing time in its simplest form means to perceive the *temporal flow or order* $\leq_T$ of foci of attention[9]. Note that we do not assume that attentional moments are indivisible for an observer. There may be many foci at a time, and so *temporal equivalence* is different from *true identity* (the former is denoted by $=_T$, the latter by $=$) among foci[10].

Axiom 1: [Temporal-order-with-identity]
$(F(x) \wedge F(y)) \leftrightarrow (x \leq_T y \vee y \leq_T x)$ totality
$x \leq_T y \wedge y \leq_T x \rightarrow x =_T y$ antisymmetry
$x \leq_T y \wedge y \leq_T z \rightarrow x \leq_T z$ transitivity
We can define the corresponding strict order, and an immediate successor relation $\prec_T$ from the strict ordering. Since the domain of foci denoted by F is assumed to be finite, every focus with a successor/predecessor has an immediate successor/predecessor.

Definition 1: [Strict-order]
$x <_T y \leftrightarrow x \leq_T y \wedge \neg y \leq_T x$

Definition 2: [Immediate-successor]
$x \prec_T y \leftrightarrow x <_T y \wedge \neg(\exists z. x <_T z \wedge z <_T y)$

The role of technical procedures is to precisely locate these foci in certain reference systems.

4.2 Predicating exact time and exact space using reference systems

Moments of attention do not provide reliable access to *exact time*, since they keep appearing and disappearing in memory. They only supply a subjective reference with respect to *now*, which is observer-relative. The absolute time underlying *temporal reference systems*, e.g., calendars and standard time, are results of *calibrating attention by periodic artificial or natural phenomena*. For example, a clock can be used to predicate temporal distances between foci of attention, based on paying attention to the number of ticks or hourly strikes in between the members of each pair.

Similarly, using spatial reference systems, such as coordinate systems, and corresponding sensors, such as GPS, it is possible to locate one's attentional focus in space. The consequence of all this is that foci as such, not only spatial objects or perceived phenomena, have mappings into regions of spatial and temporal reference systems. That is, we can assume total functions (*where when*) from foci into such regions:

Axiom 2: [Where-and-when-focusing]
$where(f) = s \rightarrow F(f) \wedge Space(s)$
$F(f) \rightarrow \exists s. where(f) = s$
$when(f) = t \rightarrow F(f) \wedge Time(t)$
$F(f) \rightarrow \exists t. when(f) = t$
Furthermore, the underlying reference systems come with algebraic operations $(+, -, \leq, \geq)$ on pairs of (temporal) regions, as well as temporal (denoted by the function *dur*) and spatial (denoted by the function *dist*) distance measures, which we directly use on pairs of foci.

4.3 Technical sensing and technical foci

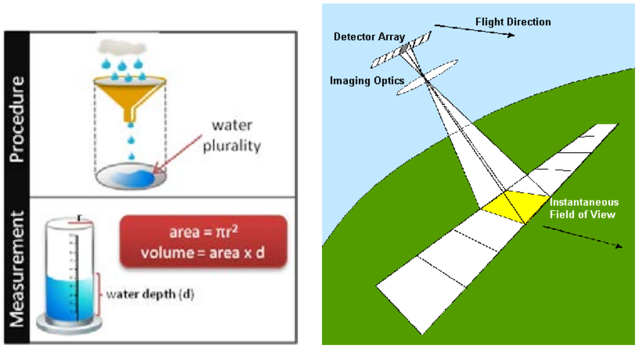
Technical sensors extend human attention by synthesizing *technical foci* and adding *new kinds of predications*. For example, a *tipping bucket rain gauge* measures rainfall intensity based on the number of times it fills up with rain and tips in an interval measured by a clock, see Figure 3 (a). It allows humans to predicate precipitation intensity on a technical focus [38]. The instrument generates its own set of granular foci, each of them having the spatial extent of the instrumental funnel opening (spatial support) and the temporal resolution of the integration interval (temporal support). Another example is remote sensing, where a focus corresponds to the instantaneous field of view, see Figure 3 (b).

7. Arguments for discreteness of human attention can be found in [34].

8. Compare [29], Chapter 3. Arguments for basing language semantics on human attention can be found in [35]. Note that we consider foci and predications as information items or "speech acts", not as elements of a theory of human cognition.

9. We leave open whether this order is based on a pulse, as argued in [33], or on comparing durations, as argued in [36].

10. Cognitive scientists assume that humans can maintain around 4 such foci simultaneously [37].



(a) (b)

Fig. 3. A *focus* is an abstraction over foci of human attention as well as spatio-temporal sensor ranges. In the case of a rain gauge (a), foci are defined by the funnel opening and the time interval of water volume measurement. In remote sensing (b), a focus corresponds to the "instantaneous field of view".

The following are predications of meteorological observations for blizzard detection with well known measurement scales for values expressed by the variable y [14]:

Axiom 3: [Met.-ground-observations]
$BlowingSnow(f) \vee windspeed(f) = y \vee prvisdist(f) = y \vee$
$windchill(f) = y \rightarrow F(f)$
BlowingSnow stands for the human observed presence of blowing (wind initiated movement) of snow at some spot in focus. *windspeed* denotes anemometer measurements predicated on the "anemometer focus". The technical focus of an anemometer, the so called "support", is the space the rotating propeller takes, which is in the order of a square foot and must be located at 10 meters (33 feet) above ground. The temporal resolution is determined by the interval during which propeller rotations are counted. The function *prvisdist* denotes a standardized procedure to determine the prevailing visibility distance around a point of view of a human observer. This point of view is the one taken into focus. Wind chill is a derived measure of how cold the weather feels to the average person, based on temperature and velocity[11].

4.4 Predicating processes

Events can be referred to by different observers based on inter-subjectively available predications. But what kinds of predications are needed for event construction? In particular, which kinds of predications can account for *identity* and *functional* aspects of events? Functional aspects include the *participatory roles* of bodies in them, in particular *causal* relations among agent and object [39]. *Individuation criteria* decide about whether the same event occurs at two opportunities, and thus account for identity of an event[12].

We follow [22] and [23] in conceiving events as bounded bits of processes. In particular, we hold that the construction of an event as an information item is based on the *predication of a process that is "going on" at some focus of attention*, similar to [24]. For example, the measure of time intervals with a sand clock is based on being able to predicate *"sand keeps running"*. While the human ability to perform such process predications seems uncontroversial, the mechanism behind it seems still unclear.

[40] and [41] have argued that the human perceptual competences (and more generally, human intelligence as such) is a result of a *situated simulation*. We similarly hold that our ability to recognise processes is dependent on our ability to make a simulation of them. Simulations enable an observer to *connect* discrete foci of attention at which the sand keeps running. Furthermore, a simulation allows to distinguish among relevant and irrevelant participants [42].

Using predicates of the following form, we express that at two foci x and y, an observer states that the *"same process is going on"*. *PType* is a wildcard standing for the underlying process simulator that enables this predication. Since the predication expresses a kind of an *equivalence*, it is assumed to be transitive and symmetric:

Axiom Schema 1: [Process-connectedness]
$PC_{[PType]}(x, y) \rightarrow F(x) \wedge F(y)$
$PC_{[PType]}(x, y) \rightarrow (z)PC_{[PType]}(y, x)$ symmetry
$PC_{[PType]}(x, y) \wedge (z)PC_{[PType]}(y, w) \rightarrow (z)PC_{[PType]}(x, w)$ transitivity

4.5 Spatio-temporal process simulators

From the viewpoint of information technology, it is desirable to mimick this human competence and thus to detect processes automatically. The human cognitive mechanisms which underlie process predications may not be known or difficult to model. However, there are many existing technical process simulations which are useful because they rely on rather simple approximations of this human competence. By formally distinguishing different kinds of process simulators, we make our theory a flexible framework for expressing and comparing different kinds of technical event detection approaches.

Technically, we will derive a relation $PC_{[PType]}$ as the *symmetric transitive closure* of a simulated preliminary relation $PC^*_{[PType]}$. Since the semantics of symmetric transitive closures cannot be expressed in FOL, we leave its definition out of the formalism, however, its computation is straightforward. We will discuss different simulators in the following and illustrate them by Figure 4.

11. Compare http://climate.weather.gc.ca/glossary_e.html.

12. Knowing the static configuration of the world in a sequence of snapshots is not enough to decide about event identity (compare [22]).

4.5.1 Process simulators based on temporal homogeneity

For example, some processes can be individuated by a *criterion of homogeneity*. This means some unary predication can be performed continuously, which accounts for identity of a process (while other properties may change). Homogeneity may be based, e.g., on a *threshold pattern*. If the water level at some river gauge continues to be above the flood stage threshold, then one can predicate that a "flooding is going on" [7]. Similarly, landslide simulation can be based on rainfall intensity thresholds [7]. Homogeneity may also be defined in terms of a certain type of *oriented change*, such as decrease, increase and expansion [28].

For example, the homogeneity condition of a *blizzard process*, i.e., the "going on" of a snowstorm, is defined by Environment Canada[13] as whenever wind speed is at least 40 km/h, windchill is at least 1600 W/m^2, and visibility is 1 km or less due to blowing snow:

Definition 3: [Blizzard-homogeneity] $H_{Blizzard}(f) \leftrightarrow windspeed(f) \geq 40km/h \wedge prvisdist(f) \leq 1km \wedge BlowingSnow(f) \wedge windchill(f) \geq 1600W/m^2$

A simple in-situ simulator can be implemented as a rule which checks whether the condition *applies continuously for temporally neighboring and spatially coinciding foci*, and then infers that the process is going on (compare Figure 4 (a)):

Definition 4: [Simple-in-situ-Blizzard-process-simulator] $PC^*_{Blizzard-insitu-simp}(x,y) \leftrightarrow x \prec_T y \wedge H_{Blizzard}(x) \wedge H_{Blizzard}(y) \wedge where(x) = where(y)$

The fact that foci coincide spatially, i.e., that the process is detected "in-situ", makes it easy to compute, since neighbors reduce to co-located temporal followers.

4.5.2 Process simulators based on spatio-temporal neighbourhood

However, many kinds of processes require a more sophisticated approach. For example, Lawson [14] adds what we call *neighbourhood conditions* to the simulation in order to detect Blizzard processes.

He proposes [14] to use a *lull period of 4 hours*, a period of ceasing Blizzard homogeneity conditions of less than four hours, during which the blizzard homogeneity condition may not be met, and which still allows to sustain the assumption that the Blizzard process is going on (compare Figure 4 (b)). The idea of this lull period is to bridge temporal variances of environmental conditions which are not due to a ceasing Blizzard. Lull periods heavily depend on the particular domain of application:

Definition 5: [In-situ-process-simulator-with-temporal-neighborhood] $PC^*_{Blizzard-insitu-temp}(x,y) \leftrightarrow \exists x'y'.x' \leq_T x \wedge x \leq_T y \wedge y \leq_T y' \wedge H_{Blizzard}(x') \wedge H_{Blizzard}(y') \wedge dur_h(x',y') < 4 \wedge where(x) = where(y)$

Note that the latter definition is much more challenging in terms of computation than Definition 4. The problem is that we cannot simply infer from the loss of homogeneity of immediate followers that a Blizzard is actually stopping. We need to take into account a temporal range of homogeneity. During that interval, immediate neighbors may not satisfy homogeneity at all but still belong to a Blizzard. In particular, we cannot infer that a focus which does not satisfy homogeneity is one where a Blizzard is *not* going on. We simply do not know, unless we have checked whether it lies inside a temporal neighborhood of homogeneous foci.

The simulators specified above are called "in situ", because they restrict the predication to stick to in-situ observations. However, our approach also allows to specify simulations of *spatially extended processes* [12]. Weather processes are good examples for the latter. In this case, one needs a simulator that connects the process over different sites, allowing to predicate that the "same process" goes on at them. Such simulators are available in practice [11] and may be based, e.g., on image recognition techniques such as the helix model [20].

Here, we propose a simple logical approach which generalizes the idea of a lag time to space in terms of a *spatio-temporal neighborhood* (see Figure 4 (c)). For this purpose, we introduce a spatio-temporal spheroid, with x located in its center and y and z lying within a maximal spatial and temporal distance from x, e.g., 25 km and 4 hours:

Definition 6: $Spheroid(x,y,z) \leftrightarrow dur_h(x,y) \leq dur_h(x,z) < 4 \wedge dist_{km}(x,y) \leq dist_{km}(x,z) < 25$

Processes can then be considered to *go on through space and time*, i.e., through foci x and y, iff x and y lie within a spatio-temporal spheroid located at focus x' and spanned by another focus y', such that y' and x' are *homogeneous*, and y' is *closer than a maximal distance* from x':

Definition 7: [Process-simulator-with-spatio-temporal-neighborhood] $PC^*_{Blizzard-spatio-temporal}(x,y) \leftrightarrow \exists x',y'.Spheroid(x',x,y') \wedge Spheroid(x',y,y') \wedge H_{Blizzard}(x') \wedge H_{Blizzard}(y')$

This process simulator moves a homogeneity spheroid through space-time and connects foci based on homogeneity neighborhoods. Simulators in Definitions 4 and 5 can be considered special cases of Definition 7, where space is kept constant. Note that from Definition 7 it directly follows that PC is locally reflexive, symmetric and transitive.

4.5.3 Process simulators with boundary conditions

In numerical simulation [42], boundary conditions play an important role in order to account for the

13. According to the definition valid before 2012, which is used in [14]. The new definition can be found here: http://www.ec.gc.ca/meteo-weather/default.asp?lang=En&n=D9553AB5-1#blizzard.

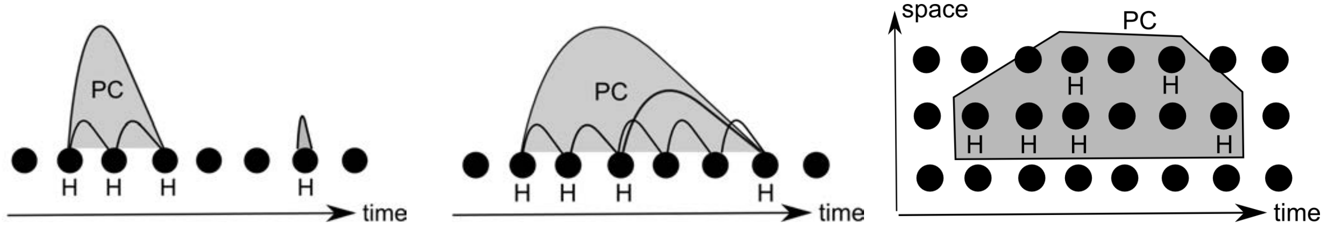


Fig. 4. Three principle variants of a process simulator: (a) is a simple in-situ simulator which connects processes based on homogeneity of temporal successors, (b) is an in-situ simulator based on lag time and (c) is a simulator based on a spatio-temporal neighborhood spheroid.

unknown. For example, every observation covers only a limited extent of the environment, an *observation window*. Beyond the spatio-temporal border of this window, it remains unknown whether a process goes on or not. A neighborhood-based simulator can be made aware of this border, which allows to make explicit that process connectedness remains unknown at the border. More precisely, once the neighborhood spheroid crosses the border of an observation window, one strictly cannot decide anymore whether a process goes on or not at the center of the spheroid since it covers unknown locations. If we constrain process simulation to the core of the observation window, we leave open whether a strip along the border of the window is connected or not[14].

For example, here is a logical schema which restricts neighborhood-based process simulation (using the definition of some process simulator $[Definiens]_{[PType]}$ as introduced above) to only those foci whose temporal neighborhood (as defined by the lag time) stays within the temporal borders defined by the start and end of the current set of observations:

Definition 8: $WithinTB(x) \leftrightarrow$
$\exists y, z.when(y) + 4 \leq when(x) \leq when(z) - 4$

Definition Schema 1: [Boundary-condition]
$WithinTB(x) \wedge WithinTB(y) \rightarrow$
$\left(PC^{*}_{[PType]}(x, y) \leftrightarrow [Definiens]_{[PType]}(x, y)\right)$

Note that by this definition, outside of the temporal interior of the observation window, it can neither be inferred that a process goes on, nor that it does not go on. In the next section, we will construct events as bounded wholes with a beginning and an end, and this beginning and end can only be inferred if it is known that a process starts or ceases to go on. If this is not the case, then, due to FOL's open world semantics, we are only able to infer the existence of *partial events*.

4.5.4 Process simulators based on event schemas

Instead of inferring process connectedness based on continuity criteria or boundary conditions, one can also rely on configurations of lower level events which trigger a higher level event. A simple example is the identification of events such as "buying a hot dog from a street vendor", which consists of two subsequent transaction events, one for transferring money, and the other for transferring a good. [9] have recently proposed a clustering approach which is based on this idea. In general, all event patterns used in conjunctive queries, such as in Esper [6], correspond to logical configurations of low level events. Since it is obvious that such configurations can be expressed in FOL, we leave away their formal specifications here.

14. This is possible because of the *open world assumption* of FOL.

5 Abstracting events from process predications

If an observer has recorded process predications, one can perform abstraction computations on them. We regard events simply as specific kinds of abstractions over foci. F is, therefore, a subdomain of the larger domain D of our FOL theory, which may also contain *reified* entities [43]. Reified entities need to be explicitly introduced by an existence claim. One way of tying reifications to foci is to individuate them as *classes of foci*[15]. We will call these *extensional reifications*.

5.1 Extensional reifications

We introduce *reified classes* as first-order entities with an element-of relation $\in$ in the domain of discourse D. Classes are extensional, so their elements give rise to an identity criterion for classes. The relation $\in$ individuates classes:

Definition 9: [Class-definition] $Class(y) \leftrightarrow \exists x.x \in y$

Axiom 4: [Extensionality]
$Class(x) \wedge Class(y) \wedge (\forall z.\ (z \in x \leftrightarrow z \in y)) \rightarrow x = y$

Axiom 5: [Distinctness-of-classes-and-elements]
$Class(x) \rightarrow \neg\,(\exists u.x \in u)$

Axiom 6: [F-as-nonempty-set-of-urelements]
$(F(x) \leftrightarrow \neg\exists z.z \in x) \wedge \exists y.F(y)$

15. We are aware of the fact that many kinds of information, e.g., predictive inferences or information from sensors that transgress the resolution of human attention, cannot be modelled as classes. There are also philosophical arguments against set-theoretic approaches towards cognitive abstraction [44]. But note that our reifications are synthesized information entities, not cognitive artifacts. Also, non-extensional reifications can be constructs with a more indirect relation to observations (see Chapter 4 in [29]).

Note that since classes are entities different from their elements due to Axioms 5 and 6, Russell's paradox cannot occur. In particular, we only allow for a singular layer of classes. It will be useful to introduce some mereological operators for these classes (where P stands for "part of" and PP for "proper part of"). Note that we do not consider these as a substitute for axiomatic mereology.

Definition 10: $P(a,b) \leftrightarrow Class(a) \wedge \forall x. x \in a \rightarrow x \in b$
$PP(a,b) \leftrightarrow P(a,b) \wedge \neg P(b,a)$

5.2 Unified wholes

There is a certain kind of class which is useful for abstracting many information items [29], in particular events. *Unified wholes* are based on *binary predications*, and they are *defined as maximally self-connected classes*. In the following axiom schemas, let R be any *binary predicate* corresponding to some predication on the domain of F.

The following axiom schema describes a predicate $Whole_R$ which applies to those reified classes that form a single whole w.r.t. R. All elements of such a whole must be mutually connected by R (this is called *unity*), and the class must be maximal in the sense that every entity which is R-related to all of the whole's elements is included:

Definition Schema 2: [Unification] $Unity_{[R]}(e) \leftrightarrow$
$Class(e) \wedge \forall x,y. \, (x \in e \wedge y \in e \rightarrow [R](x,y))$

In our case, since $R(x,y)$ means x and y "belong to the same thing" and R is symmetric and transitive, the maximality condition of a whole can be simplified to[16]:

Definition Schema 3: [Maximality]
$Whole_{[R]}(e) \leftrightarrow Unity_{[R]}(e) \wedge (\forall x, y.([R](x,y) \wedge y \in e \rightarrow x \in e))$

Wholes are our answer to question 1 and 2 in Section 1, i.e., to the problem of how to specify identity conditions for events.

5.3 Constructing events and proving facts about them

We now suggest a formal construction schema of events with boundaries (question 3 in Section 1). Many scholars have argued that events are temporal analogues of objects because they imply boundedness, while processes are analogues to matter [45], [23]. Since we hold that objects as well as events and processes are constructed based on moments of attention, we need to say how these constructions differ.

The construction of bodies naturally involves boundaries based on perceiving *surfaces*. Surfaces are indexed by a fundamental Gestalt perception mechanism that is built into our visual and tactile system [37]. However, for event individuation, we do not seem to have an equivalent mechanism. There is no perceivable "temporal surface" we can refer to. [25] argue that in human perception, discontinuities in movements may be equivalent to such surfaces. What distinguishes the construction of events from that of other bounded entities, such as bodies, is that the former bases on *transitions as temporal boundaries*. These are themselves results of a construction.

A transition is a perceivable qualitative change of a process. If a process does not go on from some moment to some immediately preceding moment (at the same location), then the pair of moments is called an *in-situ beginning*[17]. Similarly, an *in-situ ending* is when the process cannot be predicated anymore at the same location:

Definition Schema 4: [In-situ-beginning-and-ending]
$B_{[PC]}(x,y) \leftrightarrow x \prec_T y \wedge where(x) = where(y) \wedge \neg[PC](x,y) \wedge [PC](y,y)$
$E_{[PC]}(x,y) \leftrightarrow x \prec_T y \wedge where(x) = where(y) \wedge \neg[PC](x,y) \wedge [PC](x,x)$

An event is temporally bounded by transitions. But it is also a connected whole, in the sense that every pair of its moments is connected by a chain of process-connected moments. We define an *event* as a *process-connected whole* (with respect to a process predication $[PC]$) in which every focus is *temporally bounded by in-situ process transitions*:

Definition Schema 5: [Event]
$Event_{[PC]}(x) \leftrightarrow Whole_{[PC]}(x) \wedge$
$\forall f \in x. \exists b \in x, b', e \in x, e'. B_{[PC]}(b', b) \wedge E_{[PC]}(e, e') \wedge$
$where(f) = where(e) = where(b)$

Note that events may very well be *instantaneous*, in case the whole consists of a single self-connected moment with transition boundaries. However, by construction and in contrast to simple process wholes, events always imply boundary moments at which the process constituting the event ceases to occur. Thus, they imply change[18].

Commonly known *ontological properties* about events follow by construction. We proved all of the following theorems based on the resolution based theorem prover *Prover 9*[19] and the axioms so far. For example, *anti-homeomericity*, which means that an event never contains another event of the same type [24]:

Theorem 1 (Anti-homeomericity):
$Event_{[PC]}(x) \wedge PP(y,x) \rightarrow \neg Event_{[PC]}(y)$

Proof: By Definition 5, an event is a process-connected whole, which means it needs to be maximal by Definition 3. Now suppose there was a proper part x' of the event x that is also an event. By Definition 10, this means there exists some focus f that is element

16. Note that Definition 3 excludes overlapping wholes. A more general definition is in [29].

17. Other types of event-related change were proposed, e.g., by [11] and [28].

18. However, note that since we only take into account temporal bounds, we allow an event to move into or out of an observation window.

19. Sources: http://geographicknowledge.de/eventlogic/eventlogic_simple. Prover 9 can be obtained here: http://www.cs.unm.edu/~mccune/mace4/. We could also prove the consistency of the theory, see http://geographicknowledge.de/eventlogic/eventlogic_simple.model.

of x but not of x'. By Definition 5, f needs to be process-connected to all elements of x. Since x' is a proper part of x, f must also be process-connected to all elements of x', and thus must be element of x', which leads to a contradiction. □

In a similar way, one can prove that cumulating events does not form a new event of the same type, and that events of the same type cannot overlap:

Theorem 2 (Anti-cumulativity):
$PP(y,z) \wedge PP(x,z) \wedge Event_{[PC]}(y) \wedge Event_{[PC]}(x) \rightarrow \neg Event_{[PC]}(z)$

Theorem 3 (Non-overlap):
$Event_{[PC]}(z) \wedge Event_{[PC]}(y) \wedge (\exists x.x \in z \wedge x \in y) \rightarrow z = y$

Processes, in contrast, may be reified as unbounded wholes, similar to *mass nouns* [24], [46][20]. In a similar way, *partial events* can be defined as process wholes which are not events because they lack boundaries. This can either mean that the underlying process goes on forever (and thus there does not exist any corresponding event), or that the boundary lies outside the observation window, and thus remains unknown:

Definition Schema 6: $PartialEvent_{[PC]}(x) \leftrightarrow Whole_{[PC]}(x) \wedge \neg Event_{[PC]}(x)$

6 Application scenario: Inferring blizzard events

Our theory can be used, e.g., to formally infer in-situ blizzard events from ground observations, as depicted in Table 1. The times at which the homogeneity condition is satisfied in our sample of Table 1 are $\bigwedge_{i \in \{8,9,10,13\}} H_{Blizzard}(f_i)$. This is provable, as well as the contrary for all other foci.

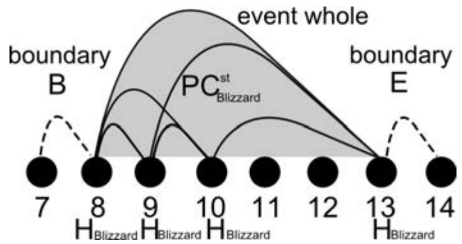


Fig. 5. A blizzard as a bounded whole of transitively closed process simulations.

It can now be inferred that the blizzard "goes on" from 10 am to 1pm, $PC_{Blizzard-insitu-temp}(f_{10}, f_{13})$, even though the homogeneity condition is not met from 11 to 12 am, and that it "ceases" from 1 to 2 pm $\neg PC_{Blizzard-insitu-temp}(f_{13}, f_{14})$. Regardless of the kind of simulator, a blizzard can be defined [14] as a corresponding event that lasts for at least four hours:

Definition 11 (Blizzard):
$Blizzard(x) \leftrightarrow Event_{PC_{Blizzard-insitu-temp}}(x) \wedge \exists z \in x, y \in x.dur_h(z,y) \geq 4$

It is now possible to infer that an appropriate class of foci corresponds to a blizzard:

Theorem 4: $Blizzard(\{f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13}\})$

All this is automatically provable based on resolution for very simple domains[21].

7 Computation of spatial event wholes

For toy domains, event construction can be computed *inside of FOL* using *resolution*, as we demonstrated above. However, resolution is too inefficient for practical purposes of database computing. Thus, FOL should be used for specifying and proving *what* should be computed, rather than for computing itself.

Following our formal architecture which distinguishes processes from events, computing event wholes has two algorithmic aspects. First, the efficiency of computing a *process simulation* on a record of attentional foci. This varies greatly with the kind of simulator, from simple Definition 5 to complex image recognition techniques for geographic processes [20] or numerical simulations [42]. Second, the efficiency of computing *wholes* on process connections. Since events are *symmetric transitive process wholes*, they correspond to *connected components* in an undirected graph, which can be even solved in linear time $O(|V|+|E|)$ [47].

7.1 Computation of spatio-temporal process simulation

The first abstract formal layer of our theory are process simulators. We have formally specified a number of different simulators in Section 4.5, and we will propose now an algorithm for computing the most general version of these, which is Definition 7. The other options discussed are special cases of this general definition. Note, however, that we do not claim that this simulator is general enough to cover all possible forms of process simulation.

In our pseudocode notation, we make use of the operators explained in Table 2, whose semantics should be obvious. For example, we used JTS[22] and Joda Time[23] Java libraries to compute spatio-temporal neighborhoods. The trick behind this algorithm is that both, the process homogeneity condition ($.satisfies()$) as well as the definition of a (spatio-temporal) neighborhood ($.neighbourhood$) can be treated as a parameter, thus allowing for different homogeneity neighborhood conditions.

The procedure is specified in Algorithm 1, and it was motivated to some extent by the idea of density-based clustering [48]. *Initial* keeps a list of yet unvisited foci which is processed until the list is empty in an outer while loop. In each iteration of this loop, the

20. However, note that we make a formal distinction between boundedness and "likepartetness" (homeomericity).

21. Compare the theorems in the Prover 9 script http://geographicknowledge.de/eventlogic/eventlogic_simple.

22. www.vividsolutions.com/jts/

23. http://joda-time.sourceforge.net/

TABLE 2
Pseudocode operations

operations	type	explanation
[ ]	*List*	the empty list
$.satisfies()$	$F \to Bool$	tests whether some homogeneity condition is satisfied
Initial	$List[F]$	list of ordered focus elements F (ordered by timestamp)
$.get()$	$List[F] \times \mathbb{N} \to F$	gets the element with the given index from a list
$.get()$	$List[F] \times F \to \mathbb{N}$	gets the index of an element in a list
$.getStart()$	$List[F] \to F$	gets the earliest element satisfying the homogeneity condition, removes all preceeding non-satisfying foci from Initial
$.remove()$	$List[F] \times F \to List$	deletes the given element from the list
$.contains()$	$List[F] \times F \to Bool$	returns true, if the given element is contained in the list
$.neighbourhood$	$List[F] \times F \to List$	creates a list of foci F in a multidimensional spheroid around some original focus, ordered by spatio-temporal distance
Neighbours	$List[F]$	keeps a list of neighboring foci
$.newEdge()$	$F \times F \to E$	creates an edge out of two foci
ProcessGraph	$List[E]$	process graph, list of edges
$.add()$	$List[E] \times E \to List[E]$	adds a new edge to a list
$.connectedSets()$	$Graph[F] \to List[Set[F]]$	generates the connected components of a graph
$.getTimeSeries()$	$F \to List[F]$	generates the temporally ordered list of foci which are colocated with a given focus
$.WithinTB()$	$F \to Bool$	checks whether a focus is within a temporal process simulation window

Algorithm 1 Process Simulation based on Homogeneity and Distance Neighborhood

Require: *Initial* contains all given foci F; $cur, newCur \in F$
$ProcessGraph \leftarrow []$, $Nbs \leftarrow []$, $continuePath \leftarrow true$
while $Initial \neq []$ **do**
  $cur \leftarrow Initial.getStart()$
  while $continuePath = true \wedge cur.satisfies()$ **do** {Continues sphere path on current focus}
    $continuePath \leftarrow false$
    $Initial.remove(cur)$ {removes visited foci}
    $newCur \leftarrow null$
    $Nbs \leftarrow Initial.neighbourhood(cur)$ {Spatio-temporal neighbours (reflexive) ordered by distance}
    for all $(i = Nbs.size() - 1; i \geq 0; i--)$ **do**
      if $continuePath = false$ **then** {if no satisfying focus has been found yet}
        if $Nbs.get(i).satisfies()$ **then** {Neighbour satisfies condition}
          $ProcessGraph.add(newEdge(cur, Nbs.get(i))$
          if $Initial.contains(Nbs.get(i))$ **then** {Neighbour not yet visited}
            $newCur \leftarrow Nbs.get(i)$ {Sets new current focus}
            $continuePath \leftarrow true$
      else {Fetches all neighbours within distance to new focus}
        $ProcessGraph.add(newEdge(cur, Nbs.get(i)))$
        $Initial.remove(Nbs.get(i))$ {Removes foci contained in the graph}
    $cur \leftarrow newCur$
  $continuePath \leftarrow true$

current focus is initialized to the earliest focus which satisfies the homogeneity condition.

Starting from there, the algorithm builds up a *homogeneity sphere path* by pushing the homogeneity sphere through the list of foci as far as possible, in any temporal or spatial direction. First, it removes the current focus (*cur*) from *Initial* and then it retrieves a neighborhood of foci around *cur* (not including *cur*) ordered by distance. In a for loop, it visits each focus in this neighborhood in the order of their distance, starting from the outermost one. If it comes across the first focus i satisfying homogeneity, then it adds an edge from *cur* to i to a *process graph*, and if i was not visited yet, then it sets i as *newCur* and continues the path. In the next for loop, continuation leads to fetching all closer neighbors to the process graph, regardless of their homogeneity, and removes them from *Initial*. This implements the (spatio-temporal) neighborhood conditions in Definition 7. After the neighborhood was completely visited, the outermost homogeneous focus (*newCur*) in this neighborhood becomes the new *cur*, i.e., the next step in the sphere path.

By the conditions in the for loop, the following is assured: if the spheroid neighborhood either does contain a homogeneous neighbor, or if all homogeneous neighbors were already visited (in case the path runs into foci which were already covered by some path), then the path will not be continued, and *cur* is set to a new starting focus in *Initial* in the outer while loop. This exhausts foci until none are left.

As in density-based clustering, the algorithm does not make any assumptions about the geometric form or the connectivity of the resulting process graph. This means that the Gestalt of the resulting events is not in any way constrained or predetermined by the algorithm.

The complexity of this algorithm is polynomial, just as density-based clustering. It depends on the index structure used to compute neighborhoods. If one can execute such a neighborhood query in $O(\log n)$, then the overall runtime complexity is $O(n \cdot \log n)$. Without an index, complexity becomes $O(n^2)$.

7.2 Computation of events

The output of Algorithm 1 is a process graph. Algorithm 2 cuts out event wholes based on this process graph. According to Definition 5, this depends on the *connectivity* of the graph as well as on the *existence of temporal boundaries* (start and end), i.e., preceding and successive foci at which process homogeneity is not met.

By Axioms 1, process connectedness is symmetric and transitive, which means that we need to consider the *symmetric transitive closure* of the process graph in order to compute event wholes. However, since wholes (see Definition 3) on symmetric transitive closures of a directed process graph correspond to *connected components* on its undirected equivalent[24], we do not need to compute the symmetric transitive closure. In the following algorithm, the method *.connectedSets*() computes event wholes based on some standard approach for finding *connected components*, such as DFS or BFS. Algorithm 2 finds such wholes and checks whether they have in-situ beginnings/endings or not, and thus whether the whole corresponds to a partial event or rather a proper event (compare Definitions 5 and 6). This is done by going through all foci of each component once and checking whether the corresponding time series (the temporally ordered list of foci which are co-located with the given focus) has a beginning and an end. The time complexity of this algorithm is therefore linear. Note that the algorithm also takes into account the boundary condition (see Definition 8 for *WithinTB*()) of process simulation as discussed in Section 4.5.3, rendering an event *partial* if it does not begin as well as end within the temporal bounds defined by the observation window. Without boundary conditions, the corresponding while-loops could be replaced by for-loops which simply go through the in-situ time series.

Algorithm 2 Event Constructor

```
Require: processGraph
events ← [], partial ← false, ts ← []
conComps ← processGraph.connectedSets() {Generates process connected components}
for all (c : conComps) do
    Set[F]cc ← c
    while f : cc do
        ts ← getTimeSeries(f) {Gets the time series of co-located foci}
        j ← ts.get(f) {Gets the index of f in ts}
        right ← false
        while ts.get(j).WithinTB() do {Constrains events to observation window}
            if ts.get(j) : cc then
                cc.remove(ts.get(j))
            else
                right ← true, break {Left event boundary detected}
            j = j + +
        j ← ts.get(f)
        left ← false
        while ts.get(j).WithinTB() do
            if ts.get(j) : cc then
                cc.remove(ts.get(j))
            else
                left ← true, break {Right event boundary detected}
            j = j − −
        partial ← left ∧ right
    events.add(newEvent(c, partial))
return events
```

Once events have been generated, it is possible to classify and to publish them. The kind of process simulator underlying event construction always contributes to classify an event. For example, Blizzard events were generated based on Blizzard processes. Furthermore, event type specific constraints may need to be included in the classification. For example, in the case of Blizzards (Definition 11), there is a minimal temporal diameter required (4 hours duration). A corresponding Blizzard classifier is straightforward and therefore not further discussed here.

7.3 Architecture and Implementation

We have implemented the event detection algorithms developed above in some prototype which is freely available here[25]. The three main class interfaces of this program are *ProcessSimulator*, *EventConstructor*, and *EventClassifier*. A process simulator is parameterized by a *ContinuityConditionChecker*, and the latter takes a criterion for a homogeneity condition, such as Definition 3, as well as a continuity condition, such as Definition 6. It has methods for assessing the spatio-temporal homogeneity neighborhood (spheroid) of a focus.

Once a process simulator is parametrized by these conditions, it can take a set of foci with appropriate attributes and generates a process graph. The event constructor, in turn, generates event instances from this graph and determines whether they are partial or not. The FOL specification proposed in this paper can be partially translated into Semantic Web standards such as *RDF*[26] and OWL[27]. A corresponing *event reference theory*[28] in OWL can then be used for event annotation, publishing, and classification. The ontology allows to express that event instances are grounded in some foci, which are located (*where*) at some spatial geometry encoded as WKT literal using the *GeoSparql ontology*[29], and (when) at some time interval encoded in terms of *OWL time*[30] and corresponding xsd:datetime literals. For specific domains, it is also possible to define classes of events in OWL and then use reasoners to do automatic event classification.

24. A symmetric, directed process graph (without loops) is equivalent to an undirected graph with the pairs of inverse arcs replaced by edges. And connected components include exactly those nodes which are transitively connected to each other in this graph.
25. https://github.com/martinkiesow/eventDetection

26. Resource Description framework, a language standard of the Semantic Web, see http://www.w3.org/RDF/.
27. http://www.w3.org/TR/owl2-overview/
28. http://www.geographicknowledge.de/vocab/EventReferenceTheory
29. http://www.opengis.net/ont/geosparql
30. http://www.w3.org/TR/owl-time/

TABLE 3
Confirmed Blizzard events[33]

Station	beginning	ending
Brandon	1964-03-18 16:00	1964-03-19 02:00
Brandon	1964-03-23 10:00	1964-03-23 19:00
Winnipeg	1964-03-23 15:00	1964-03-24 03:00

TABLE 4
In-situ event construction

Event	where	beginning	ending
e_1	Brandon	1964-03-18 16:00	1964-03-19 02:00
e_2	Brandon	1964-03-23 09:00	1964-03-23 18:00
e_3	Winnipeg	1964-03-23 12:00	1964-03-24 02:00
e_5	Gimli	1964-03-13 20:00	1964-03-14 02:00
e_4	Gimli	1964-03-23 14:00	1964-03-24 10:00
e_6	The Pas	1964-03-15 19:00	1964-03-16 06:00
e_7	Dauphin	1964-03-23 12:00	1964-03-23 20:00

7.4 Evaluation

We have applied the event detector specified above to an official dataset of hourly meteorological ground measurements in Canada[31], using data from 5 weather stations in Manitoba (Brandon, Winnipeg, Gimli, Dauphin, and The Pas) for some period in 1964. From this dataset, we first generated foci by combining the observed hours in the temporal dimension (when) and the locations of each weather station in the spatial dimension (where). Each combination corresponds to an individual focus, and each data value corresponds to some meteorological measurement, i.e., the predication of some phenomenon, including temperature, visibility, windspeed, humidity, and windchill.

First, in order to evaluate our constructor, we generated in-situ blizzard events and compared the result with blizzard warnings confirmed by Environment Canada[32], which were given as in-situ events for single sites. The confirmed events are in Table 3, and the results of our algorithm using the in-situ process simulator with lag time (Definition 5) is in Table 4.

The close correspondence between these tables demonstrates that automated blizzard detection is capable of reproducing meteorological on site assessments. The slight differences in the blizzard lengths are probably a consequence of taking slightly different homogeneity criteria into account. Furthermore, our event detector was able to find some events which were not considered blizzards by meteorologists, such as the Gimli, The Pas and Dauphin events in Table 4. As we will see below, these events may change and merge to larger ones when additionally taking spatial homogeneity conditions into account.

Is the approach also capable of identifying events in space and time? In-situ process simulation, such as used above, enforces events at different stations to be different, because the identity criterion for an event is only temporal. However, from a closer look at Table 4, one can suspect that the events e_2, e_3 e_4, e_7 actually belong to the same larger Blizzard event, which spatially extends or moves over several neighboring stations.

31. http://climate.weatheroffice.gc.ca/climateData/
32. http://pnr.hazards.ca/blizzard_website/ps_blizzard_climatology/prairie_events/event_log_web_pages/brandon.htm
33. Taken from http://pnr.hazards.ca/.

TABLE 5
Spatial event construction

Event	where	beginning	ending
e_1^s	Brandon	1964-03-23 09:00	1964-03-24 00:00
e_1^s	Winnipeg	1964-03-23 09:00	1964-03-24 10:00
e_1^s	Dauphin	1964-03-23 09:00	1964-03-23 20:00
e_1^s	Gimli	1964-03-23 19:00	1964-03-24 10:00
e_2^s	Gimli	1964-03-13 20:00	1964-03-14 02:00
e_2^s	Winnipeg	1964-03-13 20:00	1964-03-14 02:00
e_3^s	Dauphin	1964-03-15 20:00	1964-03-16 06:00
e_3^s	The Pas	1964-03-15 19:00	1964-03-16 06:00
e_4^s	Brandon	1964-03-18 16:00	1964-03-19 02:00

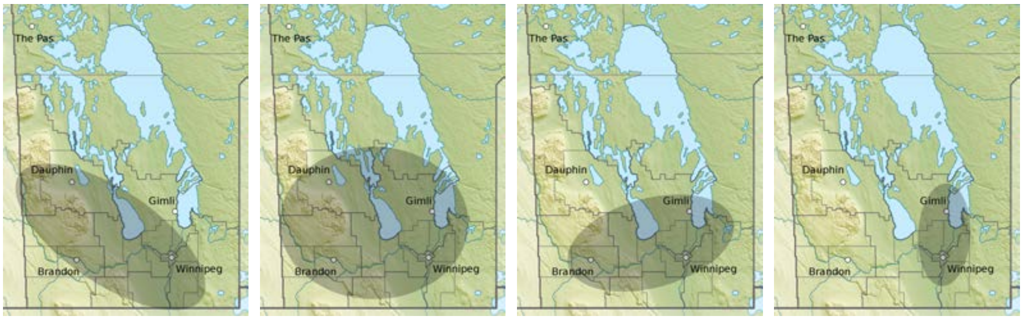
Detecting event identity over space requires a spatio-temporal process simulator, as proposed in Definition 7. The result of spatial event construction based on a process simulator with a neighborhood spheroid of 3 hours and 2.85 degree in the geographic coordinate system WGS84, and homogeneity conditions defined as before, can be seen in Table 5.

The results show that there is a single large Blizzard instance which causes subevents at 4 of the 5 stations (Brandon, Winnipeg, Dauphin, Gimli) on the 23rd of March, while the other in-situ events are actually due to different Blizzard instances over Brandon at the 18th, over Gimli and Winnipeg at the 13th, and over Dauphin and The Pas at the 15th. A spatial event constructor allows to identify events in space and time, and this enables to follow the movement of these event in space. As shown in Figure 6, the single event e_1^s begins in the southern part of Manitoba, then extends over the four stations, and then moves towards the East.

8 Conclusion

In this paper, we suggested a generalizable formal blue-print for inferring events based on *three logical levels*, namely observations, process simulation and abstraction procedures. Our approach allows to express *complex identity criteria* for events, which are usually missing in other approaches or stay often implicit. In particular, we showed how *temporal identity*, *spatial identity* and *boundary conditions* can be handled (compare the 3 questions in Section 1).

We argued that events can be formally specified as *bounded wholes of simulated process connections on foci*. This approach allows to detail for each of the logical layers separately how the relevant information may be inferred. This generates a variety of event abstraction approaches, ranging from spatio-temporal neighborhood processes (spatial identity), over stationary



(a) (b) (c) (d)

Fig. 6. Spatial blizzard detection in Manitoba. Movement of event e_1^s from 09:00 (a), over 19:00 (b), 20:00 (c) on the 1964-03-23 to 10:00 (d) on the 1964-03-24

lull periods (temporal identity), to simple threshold patterns in time series. Boundaries can be specified in terms of process beginnings and endings and can be taken into account by boundary conditions. We showed that ontological event characteristics, such as homeomericity, follows by construction, and that the existing practice, e.g., in Meteorology [14], can be captured by this blue print. Furthermore, we proposed tractable algorithms which implement it.

What is the scope of application of our work, and what are its limits? Event construction principles and identity criteria underlying event inference are often implicit, and thus not shareable via computers. One use scenario is therefore the automatic comparison of distributed event observation sources on the Web. Further work is needed to translate the formalism to a tractable subset of FOL which captures identity criteria, and which can be used to annotate and make automated comparisons of spatial event datasets on the Semantic Web and the Semantic Sensor Web [1]. Furthermore, the question remains whether all useful event construction principles can be expressed in our framework.

Acknowledgements

This work was funded by German Research Foundation (DFG KU 1368/4-2) as well as the European Commission (ICT-FP7-249120 ENVISION project).

References

[1] A. Sheth, C. Henson, and S. Sahoo, “Semantic sensor web,” *Internet Computing, IEEE*, vol. 12, no. 4, pp. 78–83, 2008.

[2] M. Goodchild, “Citizens as sensors: the world of volunteered geography,” *GeoJournal*, vol. 69, no. 4, pp. 211–221, 2007.

[3] J. Gray, D. Liu, M. Nieto-Santisteban, A. Szalay, D. DeWitt, and G. Heber, “Scientific data management in the coming decade,” *CoRR*, vol. abs/cs/0502008, 2005.

[4] P. Maué and D. Roman, “The envision environmental portal and services infrastructure,” in *Proceedings of International Symposium on Environmental Software Systems (ISESS)*, Brno, Czech Republic, June 2011, pp. 27–29.

[5] K. Teymourian and A. Paschke, “Towards semantic event processing,” in *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*. New York, USA: ACM, 2009, pp. 29:1–29:2.

[6] P. Jensen, “Complex event processing with esper,” 2008. [Online]. Available: http://jnb.ociweb.com/jnb/jnbOct2008.html

[7] A. Llaves, H. Michels, P. Maué, and M. Roth, “Semantic event processing in envision,” in *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, ser. WIMS ’12. New York, NY, USA: ACM, 2012, pp. 25:1–25:9.

[8] D. C. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001.

[9] A. Rude and K. Beard, “High-level event detection in spatially distributed time series,” in *GIScience 2012*, N. e. a. Xiao, Ed. Berlin, Heidelberg: Springer, 2012, pp. 160–172.

[10] A. Devaraju, “Representing and reasoning about geographic occurrences in the sensor web,” Ph.D. dissertation, University of Münster, 2012.

[11] K. Beard, “Modeling change in space and time: An event based approach,” in *Dynamic and Mobile GIS: Investigating Changes in Space and Time*, R. Billen, E. Joao, and D. Forrest, Eds. CRC Press, 2007, pp. 55–74.

[12] F. Reitsma, “A new geographic process data model,” Ph.D. dissertation, University of Maryland, College Park, Md., 2005.

[13] A. Devaraju and T. Kauppinen, “Sensors tell more than they sense: Modeling and reasoning about sensor observations for understanding weather events,” *International Journal of Sensors Wireless Communications and Control*, vol. 2, no. 1, pp. 14–26, 2012.

[14] B. Lawson, “Trends in blizzards at selected locations on the canadian prairies,” *Natural Hazards*, vol. 29, no. 2, pp. 123—138, 2003.

[15] W. van Hage, V. Malais, R. Segers, L. Hollink, and G. Schreiber, “Design and use of the simple event model (sem),” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 9, no. 2, 2011.

[16] A. Scherp, T. Franz, C. Saathoff, and S. Staab, “F–a model of events based on the foundational ontology dolce+dns ultralight,” in *Proceedings of the fifth international conference on Knowledge capture*, ser. K-CAP ’09. New York, NY, USA: ACM, 2009, pp. 137–144.

[17] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. L. Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor, “The ssn ontology of the w3c semantic sensor network incubator group,” *Web Semantics: Science, Services and Agents on the World Wide Web*, 2012. [Online]. Available: http://www.websemanticsjournal.org/index.php/ps/article/view/292

[18] C. Henson, A. Sheth, and K. Thirunarayan, “Semantic perception: Converting sensory observations to abstractions,” *IEEE Internet Computing*, vol. 16, pp. 26–34, 2012.

[19] D. Luckham and W. Schulte, “Event processing glossary-version 2.0,” Event Processing Technical Society, glossary, 2011.

[20] P. Agouris and A. Stefanidis, “Efficient summarization of

spatiotemporal events," *Commun. ACM*, vol. 46, no. 1, pp. 65–66, 2003.

[21] A. Stefanidis, P. Agouris, and P. Partsinevelos, "Spatio temporal helixes for event modeling," in *Proceedings of the 2002 annual national conference on Digital government research*, ser. dg.o '02. Digital Government Society of North America, 2002, pp. 1–6.

[22] A. Galton, "Experience and history: Processes and their relation to events," *J. Log. Comput.*, vol. 18, no. 3, pp. 323–340, 2008.

[23] E. Bach, "The algebra of events," *Linguistics and Philosophy*, vol. 9, no. 1, pp. 5–16, 1986.

[24] J. Seibt, "Free process theory: Towards a typology of occurrings," *Axiomathes*, vol. 14, no. 1-3, pp. 23–55, 2004.

[25] B. M. Hard, B. Tversky, and D. S. Lang, "Making sense of abstract events: Building event schemas," *Memory and Cognition*, vol. 34, no. 6, pp. 1221–1235, 2006. [Online]. Available: http://dx.doi.org/10.3758/BF03193267

[26] A. Devaraju and W. Kuhn, "A process-centric ontological approach for integrating geo-sensor data," in *Proceedings of the 2010 conference on Formal Ontology in Information Systems*. Amsterdam, The Netherlands: IOS Press, 2010, pp. 199–212.

[27] C. Henson, K. Thirunarayan, and A. Sheth, "An ontological approach to focusing attention and enhancing machine perception on the web." *Applied Ontology*, vol. 6, no. 4, pp. 345–376, 2011.

[28] C. Campelo, B. Bennett, and V. Dimitrova, "Identifying geographical processes from time-stamped data," in *GeoS'11*, 2011, pp. 70–87.

[29] S. Scheider, *Grounding geographic information in perceptual operations*, ser. Frontiers in Artificial Intelligence and Applications. Amsterdam: IOS Press, 2012, vol. 244.

[30] K. Janowicz, "Observation-driven geo-ontology engineering," *Transactions in GIS*, 2012, forthcoming.

[31] W. Kuhn, "Semantic reference systems," *Int. J. Geogr. Inf. Science*, vol. 17 (5), pp. 405–409, 2003.

[32] W. Kamlah and P. Lorenzen, *Logische Propdeutik. Vorschule des vernünftigen Redens*, 3rd ed. Stuttgart, Weimar: J.B. Metzler, 1996.

[33] E. v. Glasersfeld, *Radical Constructivism: A Way of Knowing and Learning*. London: The Falmer Press, 1995.

[34] R. VanRullen and C. Koch, "Is perception discrete or continuous?" *Trends in Cognitive Sciences*, vol. 7, no. 5, pp. 207–213, 2003.

[35] G. Marchetti, "A presentation of attentional semantics," *Cognitive Processing*, vol. 7, no. 3, pp. 163–194, 2006.

[36] ——, "Studies on time: a proposal on how to get out of circularity," *Cognitive Processsing*, vol. 10, no. 1, pp. 7–40, 2009.

[37] Z. Pylyshyn, *Things and Places. How the Mind Connects with the World*. Cambridge, Massachusetts: The MIT Press, 2007.

[38] S. Scheider, C. Keßler, J. Ortmann, J. Devaraju, A. Trame, T. Kauppinen, and W. Kuhn, "Semantic referencing of geosensor data and volunteered geographic information," in *Geospatial Semantics and the Semantic Web. Foundations, Algorithms, and Applications*, ser. Semantic Web and Beyond, N. Ashish and A. Sheth, Eds. Berlin: Springer, 2011, vol. 12, pp. 27–59.

[39] B. Smith and P. Grenon, "The cornucopia of formal ontological relations," *Dialectica*, vol. 58, pp. 279–296, 2004.

[40] J. Hawkins and S. Blakeslee, *On Intelligence: How a New Understanding of the Brain will Lead to the Creation of Truly Intelligent Machines.* Henry Holt, 2004.

[41] L. Barsalou, "Situated simulation in the human conceptual system," *Language and Cognitive Processes*, vol. 18, no. 5-6, pp. 513–562, 2003.

[42] B. Zeigler, *Theory of Modelling and Simulation*. John Wiley & Sons, 1976.

[43] S. Scheider, F. Probst, and K. Janowicz, "Constructing bodies and their qualities from observations," in *Formal Ontology in Information Systems. Proc. of the Sixth International Conference (FOIS 2010)*. Amsterdam: IOS Press, 2010, pp. 131–144.

[44] N. Goodman, *The Structure of Appearance*, 3rd ed. Dordrecht: Reidel, 1977.

[45] A. Galton and R. Mizoguchi, "The water falls but the waterfall does not fall: New perspectives on objects, processes and events," *Applied Ontology*, vol. 4 (2), pp. 71–107, 2009.

[46] K.-U. Carstensen, "Toward cognitivist ontologies," *Cognitive Processing*, vol. 12, pp. 379–393, 2011.

[47] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, pp. 146–160, 1972.

[48] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2 nd Int. Conf. on Knowledge Discovery and Data Mining(KDD ' 96)*, 1996, pp. 226–231.