UNIVERSITÄT
KOBLENZ · LANDAU
Institut für Informatik

FB 4
Informatik

A Model of Events based on a Foundational Ontology

Ansgar Scherp
Thomas Franz
Carsten Saathoff
Steffen Staab

Nr. 2/2009

Arbeitsberichte aus dem Fachbereich Informatik

Universität Koblenz-Landau • Campus Koblenz • Universitätsstraße 1 • 56070 Koblenz
Telefon +49 261 287-0 • http://www.uni-koblenz.de

Arbeitsberichte des Fachbereichs Informatik

ISSN (Print): 1864-0346
ISSN (Online): 1864-0850

Kontaktdaten der Verfasser

Ansgar Scherp, Thomas Franz, Carsten Saathoff, Steffen Staab
Institut für Informatik
Fachbereich Informatik
Universität Koblenz-Landau
Universitätsstraße 1
D-56070 Koblenz
EMail: scherp@uni-koblenz.de, franz@uni-koblenz.de, saathoff@uni-koblenz.de; staab@uni-koblenz.de

A Model of Events based on a Foundational Ontology

Ansgar Scherp, Thomas Franz, Carsten Saathoff, and Steffen Staab
University of Koblenz-Landau, Germany
{scherp,franz,saathoff,staab}@uni-koblenz.de
http://isweb.uni-koblenz.de

January 30, 2009

The lack of a formal event model hinders interoperability in distributed event-based systems. Consequently, we present in this paper a formal model of events, called F. The model bases on an upper-level ontology and provides comprehensive support for all aspects of events such as time and space, objects and persons involved, as well as the structural aspects, namely mereological, causal, and correlational relationships. The event model provides a flexible means for event composition, modeling of event causality and correlation, and allows for representing different interpretations of the same event. The foundational event model F is developed in a pattern-oriented approach, modularized in different ontologies, and can be easily extended by domain specific ontologies.

1. Introduction

The explicit modeling of events and event-based systems are increasingly gaining widespread attention by research and industry [24] due to a couple of reasons. Firstly, we find an increasing number of systems that are treating events, e.g., media delivery, recognition of vandalism, or management of emergency incidents. Secondly, a fastly growing number of intelligence-collecting devices such sensors, CCTV, upload facilities, and others lead to an ubiquity of events being recognized and communicated. Thirdly, event detection, clustering, and annotation is and will be realized in many different software components and proprietary solutions using a large variety of internal data models.

Thus, multiple systems are connected for managing events resulting in a complex, so-called distributed event-based system (also cf. Section 2.2). Such a distributed event-based system is a software system consisting of several components that are characterized by taking events as input and providing events as output. However, in contrast to related work in the field of publish/subscribe systems that work at a technical level of event management, e.g. [4, 24], the distributed event-based systems considered here deal with events on the level of the domain under consideration. These domain events may be very complex and may be linked to a variety of aspects such as time and space, objects and persons involved, as well as structural relationships like mereological, causal, and correlate relationships. Thus, the semantic interpretation of ad-hoc, ideosyncratic event models may easily become ambiguous rendering the communication between and integration of the different event-based components (and possibly different event-based systems) a challenging task. Consequently, a common model for domain events and their various aspects is needed. However, today's event models such as [23, 17, 6, 32] are developed ad-hoc and lack formal semantics. This hinders interoperability of the different event-based components and event-based systems aggravating the treatment of events in already complex, distributed infrastructures.

What is needed is a formal representation of events in a model that allows easy interchange of event information between different event-based components and systems [24]. Such an event model is presented in this paper. For our event model, F, we have analyzed existing event models and event-based systems and we have conducted a study of relevant work in foundational sciences such as philosophy and linguistics. As a result, we have decided to base the event model F on the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [10, 16] and to follow DOLCE's pattern-oriented approach for ontology development. The DOLCE approach provides native support for modularization of F and extension by domain specific ontologies. The event model F provides support for all aspects of events, namely time and space, objects and persons involved, mereological relationships of events through sub-events, causal relationships, and documentation by sensor data such as media content.

The remainder of this paper is organized as follows: The next section introduces the concept of events and motivates the metaphysical distinction between events and objects. A concrete scenario motivates the need for a formal event model F. From an analysis of related work, we have derived a set of requirements on a common event model presented in Section 3. In Section 4, we describe the development of our event model and its patterns. The use of the event model F is demonstrated in Section 5. In Section 6, we present an extensive analysis of existing event models and systems, before we conclude the paper.

2. Motivation

According to the literature, there are different characteristics of events [5]. As the definition of events is of metaphysical nature and essential for this paper, we first discuss it in Section 2.1. Subsequently, we motivate the need for a common, formal event model by a concrete scenario in Section 2.2.

2.1. Events

In philosophical literature, there are different discussions of how to discriminate events from other categories that are put as the ontological competitors of events [5]. One of them are objects. Although not undisputed, there are standard differences between events and (physical) objects [5]: Events are said to **occur** or **happen**. They are considered perduring entities (or perdurants or occurants) that unfold over time, i.e., they take up time. In contrast, material objects such as stones and chairs are said to **exist**. Such enduring entities (or endurants or continuants) unfold over space, i.e., they are in time. As said, this metaphysical distinction is not uncontroversial as some philosophers consider objects as four-dimensional entities that extend across time just as they do across space [5]. However, in our approach, we separate the two to allow for clearly distinguishing them. This seems to be a pragmatic solution as merging them later would be very easy and convenient if such a distinction is not necessary such as in mechanical engineering [2].

2.2. Scenario

An example of a distributed event-based system is the emergency response use case of the EU project WeKnowIt[1], which is depicted in Figure 1. Here, different professional entities are involved such as the emergency hotline, police department, fire department, and emergency control center. The emergency control center is in charge of coordinating the emergency response entities. It processes the event descriptions from the emergency hotline and communicates with the police department and fire department about the incident event. Floating liaison officers as part of the emergency control center are out in the field to report about the situation by taking photos and notes.

This socio-technical system for emergency response becomes active in very different concrete incidents. For example, a heavy storm with a major flooding may happen. In the course of the incident a power outage occurs. As a consequence of this, citizens are calling the emergency hotline to report about the outage. The officers at the emergency hotline record these calls and type in an event description for each call to document them in their system. These events are annotated with information about the call and its recording and are automatically transferred to the system of the emergency control center. Many other citizens have already taken photos of the flood event and report them. Here, the citizens describe the event using an application on their cell phone by attaching a photo to it and tagging it and send it to the system of the emergency control center. As they are already alarmed, the emergency control center may also be informed about event descriptions from the systems of the police department and fire department such as the pumping out of flooded cellars or about the rescuing of people trapped in their homes. Based on the evidence about the events the emergency control center collects on their system, the center may confirm the incident event and may set up an emergency response team to alleviate the situation. The officers in the emergency control center formulate hypothetical events that might have caused the power outage. To this end, the

[1] `http://www.weknowit.eu/`

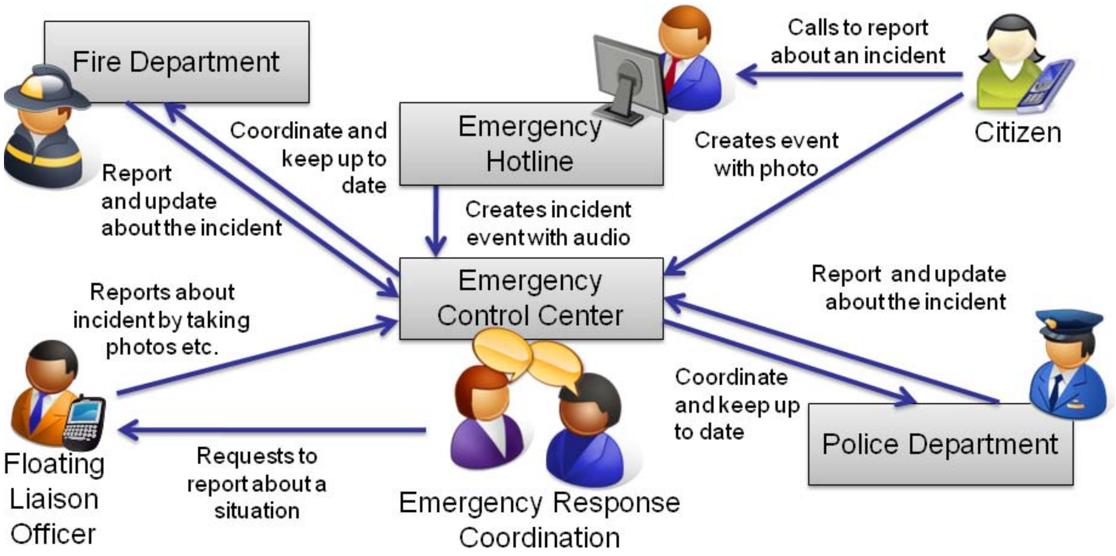

Figure 1: A distributed event-based system for emergency response

officers in the control center analyze, (semi-)automatically cluster, visualize, and put the events into relation. Possible causes they identify are a snapped power pole but also a problem with the power plant. Once the hypothetical events are modeled in the system, the officers try to verify them. For this purpose, the officers in the emergency control center may contact the personnel of the power plant. At the same time, the hypothetical event of a snapped power pole together with a task description is sent onto the PDA of a floating liaison officer. The floating liaison officer drives to the area where the control center suspects that the pole might be snapped and reports back the result using his PDA.

As depicted in Figure 1, several professional entities are involved in an emergency response using different systems. In addition, also citizens report events using an appropriate application on their cell phones. These systems and applications describe events, process them, and use the representations and descriptions of events. They need to be connected through a common understanding of events into a distributed, event-based system to enable communication between them. This can only be facilitated by a common, formal event model.

3. Requirements on a Common Event Model

From our introductory scenario and the related work, we have derived a comprehensive set of requirements that a common event model should support. These functional requirements are providing support for five aspects that can be defined for events, namely the constitutive, temporal, spatial, experiential, and structural aspects. Besides these aspects of events, a common event model shall also provide for different interpretations of events.

- **Constitutive Aspect.** The constitutive aspect describes the living and non-living objects participating in an event such as people, animals, and other material objects.
- **Temporal Aspect.** The temporal aspect covers the temporal extension of an event. It can be modeled using absolute or relative representations of time.
- **Spatial Aspect.** The spatial aspect is in charge of capturing the spatial dimension of objects participating in the event. This can be also modeled using absolute or relative positioning.
- **Experiential Aspect.** The experiential aspect comprises the annotation of events with sensor data such as media data.
- **Structural Aspect.** The structural aspect considers the arrangement of events in *mereological*, *causal*, and *correlative* relationships. Events may be and usually are made up of other events [22]. Thus, the common event model shall support the modeling of mereological relationships between events. Causality is one of the most difficult topics in philosophy [13]. It requires the modeling of causes and effects and should support the integration and use of different causal theories as discussed, e.g., in [13]. Correlation refers to two events that have a common cause (cf. [27]). While causality is very difficult to discover and, hence, often unknown correlation is typically easy to observe.
- **Event Interpretations.** Structural relations between events such as causality and correlation can be a matter of subjectivity and interpretation. For example, in a law-suit the parties involved may each claim that the other one is at fault. A common event model should be prepared to support such different interpretations of the same event.

4. Event Model F

For designing our event model F, we apply a methodology based on a foundational ontology. Such a foundational ontology provides a domain independent vocabulary that explicitly includes formal definitions of foundational concepts [1]. Based on a previous review [19], we decided to use DOLCE [10, 16] as modeling basis. Our choice is influenced by the pattern-oriented design approach that is pursued with DOLCE. The development of our event model F also bases on patterns, more precisely on specializations of the *descriptions and situations* (DnS) ontology pattern that provides a formal representation of context [11] and is part of DOLCE. We aligned the event model F with the DOLCE+ Ultra Light (DUL) ontology[2] that provides more intuitive namings

[2] `http://wiki.loa-cnr.it/index.php/LoaWiki:DOLCE-UltraLite`

and concept definitions than the older DOLCE version. DUL already defines the class `Event` next to the disjunct upper classes `Object`, `Abstract`, and `Quality`. The definition of `Event` has been specialized from the formal definition in DOLCE as an entity that exists in time (cf. discussion of events and objects in Section 2.1). Disjunctive to the class `Event` is the class `Object` that stands for entities that exist in space such as living things as well as non-living and abstract things like mental concepts. A `Quality` is a characteristic of an object or an event that has a value which is represented as a point or area in some `Abstract`. The class `Abstract` represents value spaces, e.g., the space of natural numbers, the RGB color space, or the time of a day. In our event model F, we do not prescribe specific `Abstract`s that are to be used. Thus, we refer to the generic `Abstract`s already defined in DUL such as the regions `TimeInterval`, `SpatioTemporalRegion`, and `SpaceRegion`.

We designed our event model F[3] in consideration of prior research on event representation. The formal representation of the experiential aspect (see requirements in Section 3) is already possible using existing approaches, e.g., the Core Ontology for Multimedia [1]. For the spatial and temporal aspects the DOLCE-aligned ontology modules for temporal relations and spatial relations[4], established ontologies for time and space such as OWL-Time [29] and WGS84 geo positioning [30], and others can be used. The remaining aspects and the requirement for event interpretation are represented by specialized instantiations of the DnS ontology pattern. In the following, we explain the ontology patterns of the event model F including graphical illustrations of them. Classes defined by the event model F are highlighted to show the alignment with classes of DUL that are drawn with white background.

4.1. Participation Pattern

One aspect of an event is given by the objects participating in an event such as persons. The participation pattern of the event model F enables to express this constitutive aspect of events formally. As shown in Figure 2, participation is expressed by an `EventParticipationSituation` that `satisfies` an `EventParticipationDescription`. The situation includes the `Event` being described and the objects being participants of this event. The `EventParticipationDescription` classifies the described event and its participants by the concepts `DescribedEvent` and `Participant`.

The concept `DescribedEvent` `classifies` the `Event` described by an instance of the pattern, e.g., the event of a flooded cellar. As is presented in DOLCE's DnS pattern, classifiers such as “Flooding” or “Hurricane” are represented as instances of the class `Concept`. Thus they occur at the same modeling level like concrete events such as the Hurricane Katrina. Only in this way, the instantiation relationship between “Katrina” and “Hurricane” can be reified and thus be talked about in a first-order theory. Likewise, instances of `Participant` classify objects as participants of the event, e.g., the fireman reporting about the event of a flooded cellar, the landlord of the affected building, and the affected building itself. Instances of `Participant` can be defined in a domain ontology

[3] $F = E + 1$, a homage to the event model E [32] by Westermann and Jain.

[4] `http://wiki.loa-cnr.it/index.php/LoaWiki:Ontologies`

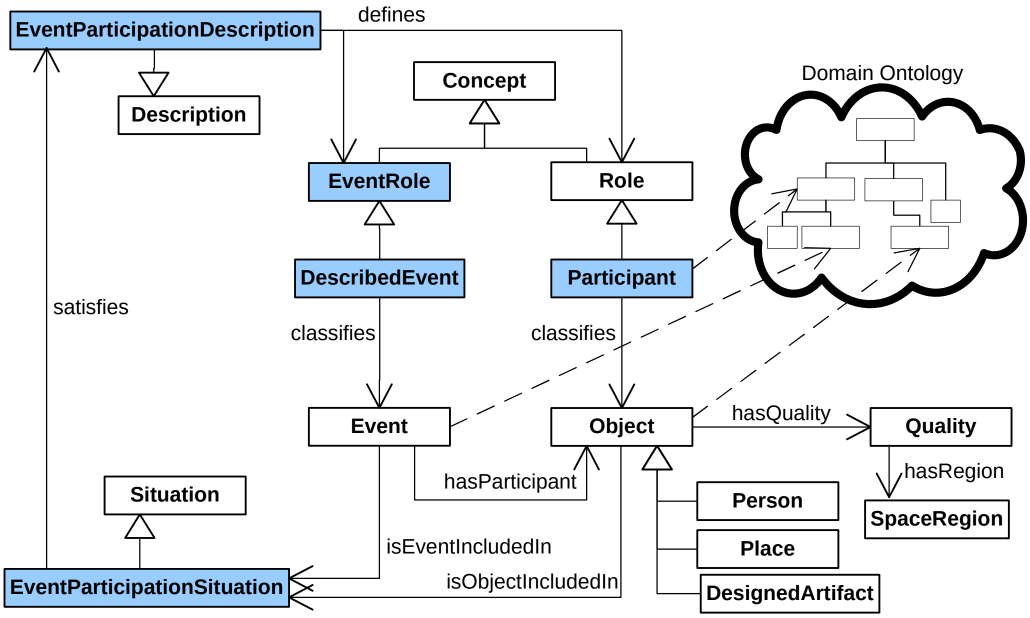

Figure 2: Participation Pattern

as indicated by Figure 2. For instance, some domain ontology modeling emergency terminology may define the role of a person being affected by some emergency case, i.e., the emergency subject, and the role describing the rescue staff such as firemen. As indicated by Figure 2, the described event, participating objects, and their roles can be defined in some domain ontology so that the participation pattern can be applied to express participation with respect to arbitrary application domains, e.g., emergency response.

4.2. (De-)Composition Pattern

As indicated by the emergency response scenario in Section 2.2, events are commonly considered at different abstraction levels depending on the view and the knowledge of a spectator. For instance, the local event of a flooded cellar may be considered as such or as part of a larger event of a regional flooding in which many such (smaller) incidents occur. The composition pattern enables to express such relations as the composition of events. Here, the composite event is the "whole" and the component events are its "parts". Formally, an `EventCompositionSituation` includes one instance of an event that has the `EventRole` of a `Composite` event and one or many events considered as a `Component`(s) of that event (cf. Figure 3). Accordingly, an `EventCompositionSituation` `satisfies` a `CompositionDescription` that `defines` the concepts `Composite` and `Component` for classifying the composite event and its component events.

Component events may be further qualified or disqualified based on temporal, spatial, and spatio-temporal qualities. For instance, component events may be required to occur within a certain time-interval, e.g., the second week of June 2009. Moreover, component events may be required to take place in a certain spatial region, e.g., the flooding of the town Sheffield should be composed exclusively by events that occur within a certain range of longitude and latitude. Furthermore, events may be qualified by a spatio-temporal quality like the progress of a flood that extents over time and space, starting with a high water level located in some area of a river and extending spatially over time into close-by areas. Any such constraints on component events are formally expressed by one or multiple instances of an `EventCompositionConstraint` that prescribes qualifying values for temporal, spatial, and spatio-temporal qualities of a component event. While events are formally defined as entities that exist in time and not in space (see Section 2.1), constraints including spatial restrictions are precisely expressed through constraints on the participating objects of a component event (cf. Figure 3).

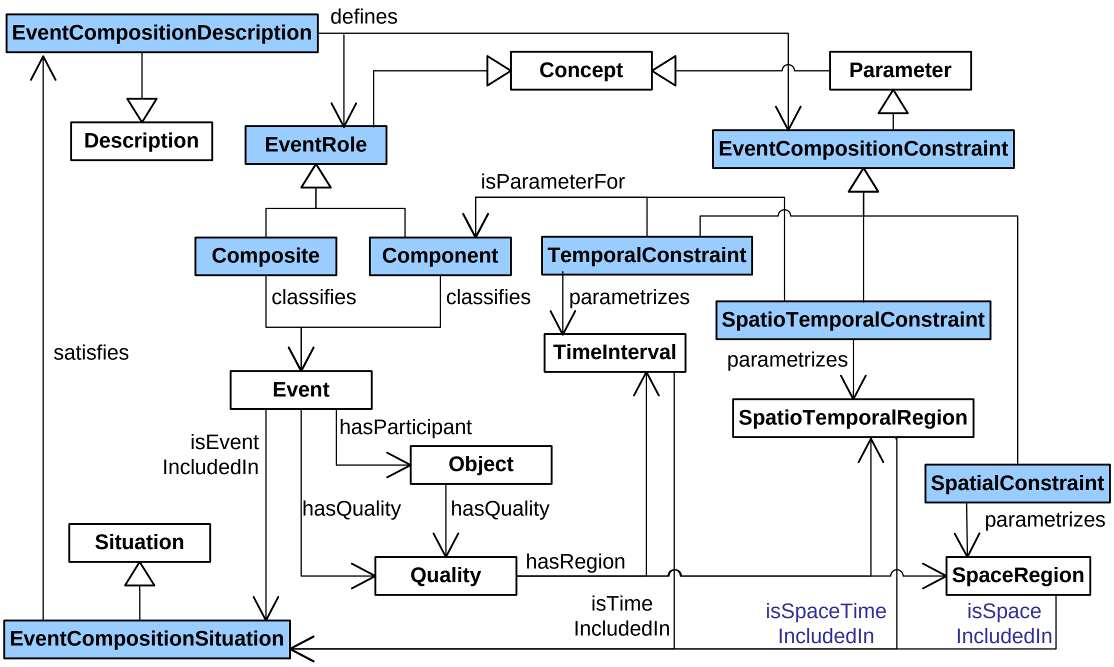

Figure 3: (De-)Composition Pattern

4.3. Causality Pattern

Causality is the traditional philosophical problem investigating the existence (or nonexistence) of any special "tie" binding causes and effects together [13]. It can be questioned either as "Why" or "How" [13]. An answer to this question is (or better said claims to be) a causal explanation. What explains, is the cause and that what is explained is the effect [13]. Events are the most natural concept to serve for defining causal relations [22]. In fact, causes and effects are events [13] and only events can be causes and effects [15]. Causation is unidirectional [13], i.e., causes are bringing about effects in an orderly succession, and causes are a necessary condition of its effect [13].

Based on the analysis of related work and discussion above, we designed a causality pattern depicted in Figure 4. The pattern defines two `EventRoles` called `Cause` and `Effect` which `classify Events`. Furthermore, a `Description` is classified by a `Justification`. Thus, the pattern expresses an explicit causal relationship between the cause and the effect under the justification of some theory, which might be an opinion, a scientific law, or even not further specified. However, the important property is that a causal relationship is always justified by some (maybe implicit) underlying causal theory. In a situation of a heavy storm, a power outage might occur and many citizens are calling the emergency hotline. This causal relationship would be justified by the social norm, which allows citizens to call the emergency control center in cases of emergency.

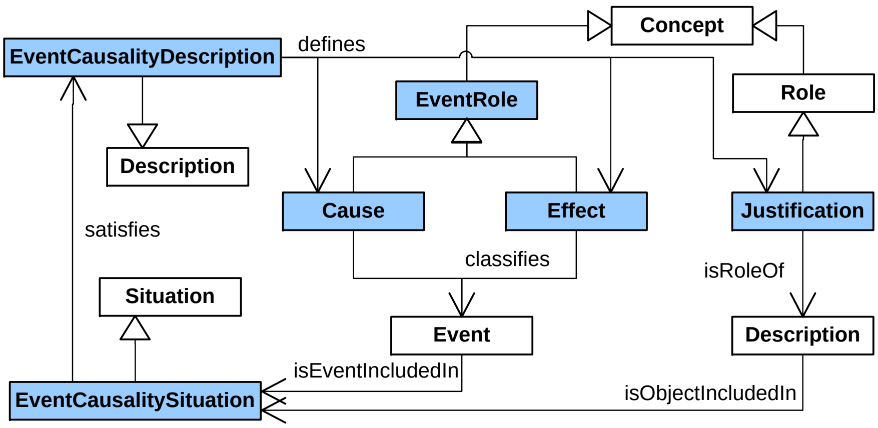

Figure 4: Causality Pattern

As defined above, causes and effects are events. Thus, we assume that objects inherently involved in the causal relationships are properly associated to the cause and effect by using the participation pattern from Section 4.1. Further, the causality pattern defines that exactly one cause is tied to one effect. If there are more than one event classified as causes or if a cause implies multiple effects, we introduce a super-cause (consisting of a set of events classified as causes) and a super-effect (the set of implied effects), respectively.

As the discussion of different causal theories shows [13], not all theories separate between objects and events as clearly as F does. The question arises, how to deal with causality if the two concepts are merged? In such a case, we assume that the causal "tie" that is modeled in the theory is applied to the event and its relation to the object is defined by the participation of the objects in the events. This allows for describing the participation of objects relevant in the context of a given causal relationship by using the causality pattern in combination with the participation pattern from Section 4.1.

Finally, according to Shafer [26] there are different kinds of causal relationships. Thus, using a single causality relationship would not be precise enough. Consequently, her logic

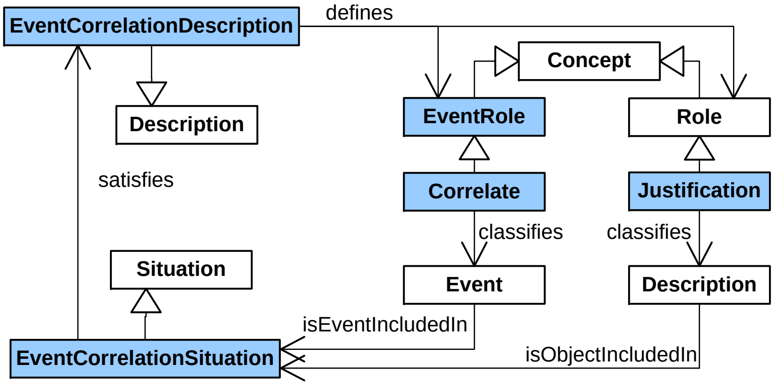

Figure 5: Correlation Pattern

for causality allows for explicitly modeling the different kinds of causal relationships found in literature and avoids labeling any one as such. This can be supported with our event model F by specializing the causality pattern to those specific kinds of causal relationships.

4.4. Correlation Pattern

We call a set of events correlated, if they occur at the same time (or share some overlap) and have a common cause. However, there exists no causal relationship between the two events (cf. [27]). The common cause may origin from a single or a chain of multiple preceding cause-effect relationships. Correlation also differs from co-occurrence where two or more events just (randomly) happen at the same time and do not have a common cause. Correlation is not of metaphysical interest as it is a property that can be derived from causality, i.e., the common cause. In our ontology we model correlation explicitly as in many cases the (correlating) effects of some common cause may be known, while the cause itself is not. The correlation pattern depicted in Figure 5 defines the role `Correlate` to `classify` the events that are correlated. The `Justification` role classifies some `Description`, which gives an explanation for the correlation in terms of a (mathematical) law or some theory that was used to determine the correlation.

4.5. Interpretation Pattern

The perception of an event depends on the context of an observer. In the emergency use case, two emergency control officers might have differing interpretations of the power outage. One might be convinced that the power outage is due to a snapped power pole, while the other might think of more serious case of a damaged power plant. Such different, context-dependent interpretations of an event can be described formally by instantiating the different event model F patterns presented so far, serving different aspects of events. One of the fundamental design decisions underlying our event model is the identification and clear separation of the different aspects of events, i.e., the separation of concerns. However, in order to move from the formal treatment of events and objects to a more natural and common sense interpretation of events, we provide the interpretation pattern depicted in Figure 6. Each instance of this pattern models a single, specific interpretation of an event by associating *decompositions*, *participations*, *correlations*, and *causal relationships* relevant in the context of this specific *interpretation*.

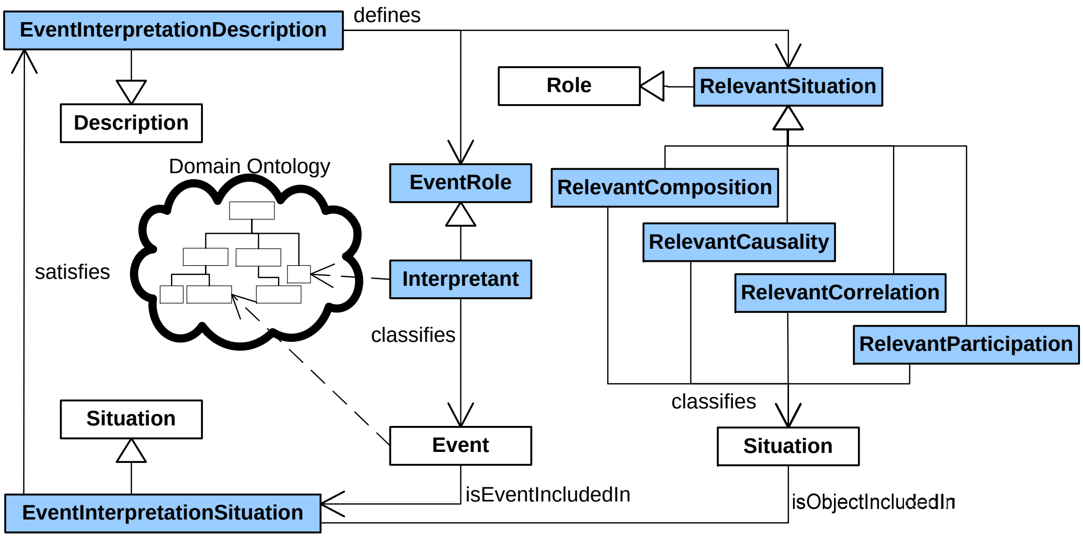

Figure 6: Interpretation Pattern

We define the `Interpretant` as an `EventRole` that classifies the `Event` we interpret. The `Interpretant` might be some domain ontology individual that specifies how an event is interpreted, e.g., as an emergency incident in case of the emergency control center, or as a news event covered by a news paper. Within each interpretation, we classify the `RelevantSituations`, namely the situations satisfying the decomposition, participation, correlation, or causality descriptions, respectively. These are defined as sub-classes of `RelevantSituation` and are called `RelevantComposition`, `RelevantParticipation`, `RelevantCorrelation`, and `RelevantCausality`.

4.6. Summary

We designed a formal event model F based on a foundational ontology. The experiential, temporal, and spatial aspects of events can be well covered by existing ontologies. We introduced new ontology patterns of small size to cover the remaining aspects of events, namely event constitutives (object participation) and structural relations. Finally, we introduced a pattern for event interpretation that allows to represent different "point of views" onto the same event.

5. Use of Event Model F

In this section, we demonstrate the use of the event model F and provide examples for two of our patterns. The scenario is the emergency situation described in Section 2.2. In this case many events occur and are documented within the different systems. For instance, for each call to the hotline an event description is typed into the system, the fire and police departments document events during their work, and the emergency control center relates all of them with some super-event. In the following, we focus on a phone call event and show how the participants of the phone call can be represented and how the phone call event can be related to the overall flooding event using our patterns. We further describe, how other events can be modeled using our patterns, without providing examples at the instance level.

Figure 7 depicts a phone call event `call-1` that is described in two patterns. The event was documented when a citizen called the emergency hotline. The lower part shows the participation pattern for this event. The phone call involves two persons, the caller and the officer who answers the call. These two persons are represented by the individuals `person-1` and `person-2`. Please note that these instances are of type `Person`. The role that they play within this event, i.e., the role caller or officer, respectively, is specified by individuals of type `Participant`. In this case, we use the individuals `caller-1` and `officer-1`, which are of type `Caller` and `Officer`, respectively. These concepts are defined within a domain ontology and are reused. Since reusing these concepts as roles automatically infers a subclass relationship between the domain concepts and `Participant`, we assume that either the domain ontology is not aligned with DOLCE or that it is aligned correctly, so that the inference should not lead to any inconsistencies. The upper part of Figure 7 depicts the composition pattern, which describes the composition of a super-event `flooding-1` with the two sub-events `call-1` and `call-2`. We classify the super-event with an individual `composite-ev-1` of type `EventType`. Each component of the super-event is classified by individuals of type `Component`. Every further incoming call and all events related to the flooding that are, e.g., documented in and collected from other systems, are added to this composition.

All further patterns are used in a similar fashion. If we assume that one of the officers in the emergency control center has formulated an event `power-outage-1` to represent the power outage that is currently happening, he might link this to the incoming calls reporting the outage using the causality pattern. In the pattern, the event `power-outage-1` would be classified as the `Cause`, while a phone call `call-1` would be classified as the `Effect`. As this example shows, we reuse the individuals representing events in the patterns, but every pattern only describes a certain aspect of the event. In the composition pattern the event `call-1` is a component, while in the causality pattern it is an effect. We can even collect patterns within an interpretation and thus formulate a certain view on the events. For instance, we could have an event representing the snapped power pole, e.g., `snapped-pp-1`, and one event representing a problem with the power plant, e.g., `problem-pp-1`. For one interpretation, we would create a causal relationship between `snapped-pp-1` and `power-outage-1` and for the other a causal relationship between `problem-pp-1` and `power-outage-1`. The officers in the

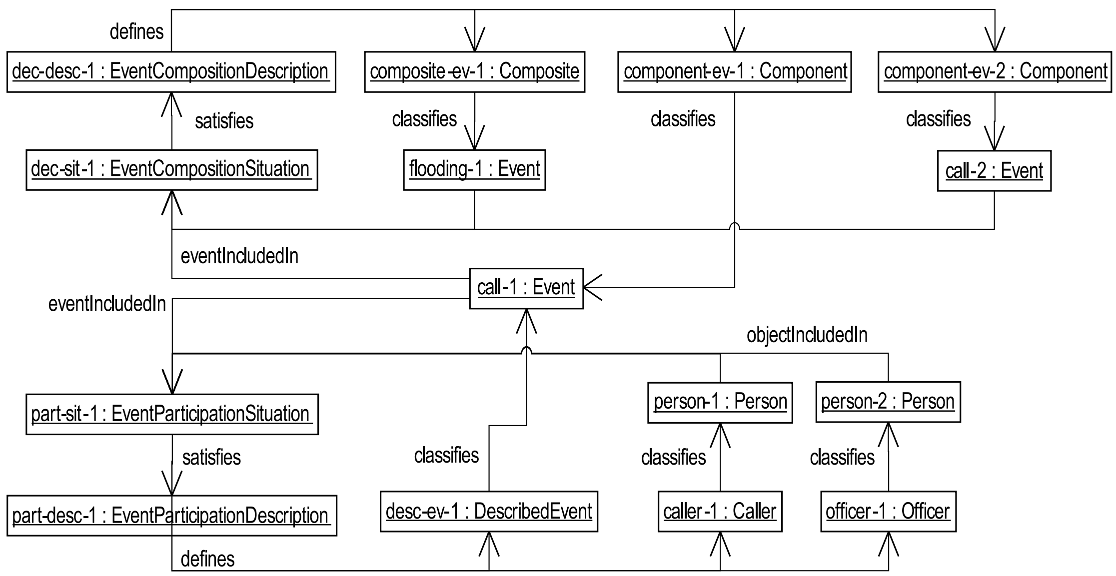

Figure 7: Application of participation pattern and composition pattern for a "call" event

emergency control center might relate these causal relationships with the other event descriptions using the interpretation pattern. All situations (representing the descriptions of the events) that are required to fully describe the interpretation are classified as `RelevantSituation`s within the interpretation pattern. Such an interpretation could further be analyzed within the emergency control system, e.g., in order to validate some of the interpretations or disprove them and to support the officers in making the right decisions.

6. Existing Event Models

For designing our event model F, we analyzed existing event-based systems and event models. These models are motivated from different domains and are the Eventory [31] system for journalism, the Event Ontology [23] as part of a music ontology framework, the Semantic-syntactic Video Model (SsVM) [7] and Video Event Representation Language (VERL) [8, 17] for video data, the ISO-standard of the International Committee for Documentation on a Conceptual Reference Model (CIDOC CRM) [6, 28] for cultural heritage, and the event model E [14, 32] for event-based multimedia applications. An overview of the analysis results and comparison to the features of our event model F along the requirements in Section 3 is shown in Figure 8. Our analysis shows that the existing event models almost fully support the constitutive, temporal, spatial, and experiential aspects. Support for some aspects remains unclear, i.e., absolute and relative spatial positioning in SsVM and VERL, respectively, and the linkage of CIDOC CRM with the experiential aspect.

	Consti-tutive	Temporal		Spatial		Experi-ential	Structural			Inter-pretation
		Relative	Absolute	Relative	Absolute		Mereologic	Causal	Correlate	
Eventory	Yes	Yes	Yes	Yes	Yes	Yes	Limited	Limited	No	No
Event Ontology	Yes	Yes	Yes	No	Yes	No	Limited	Limited	No	No
SsVM	Yes	Yes	Yes	Yes	Unclear	Yes	Yes	Limited	No	No
VERL	Yes	Yes	Yes	Unclear	Yes	Yes	Limited	Limited	No	No
CIDOC CRM	Yes	Yes	Yes	Yes	Yes	Unclear	Limited	Limited	No	No
Event Model E	Yes	Yes	Yes	Yes	Yes	Yes	Limited	Limited	No	Limited
Event Model F	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Figure 8: Event models compared to the requirements

However, existing event models substantially lack in supporting the structural aspect, i.e., mereological, causal, and correlation relationships, and representation of different interpretations of the same event. Here, we find different variations of limitations or even no support by the existing event models. With respect to mereological relationships, the existing event models typically provide support for simple part-of relationships such as with the `sub_event` property of the Event Ontology. However, no further axiomatizations are provided for refining the mereological relationship by different criteria such as temporal and spatial constraints. Only SsVM allows for describing more complex mereological relationships. Similar, also the causal relationship is only limited supported by the existing event models. It is typically defined as a cause-effect relationship such as with the `factor` and `product` properties of the Event Ontology or the "resulted in" property in CIDOC CRM. Here, also no further axiomatization of causality is provided by the existing models such as that causes and effects are events and only events, cardinality between causes and effects, and support for modeling a justification of a causal relationship (see Section 4.3). The correlate relationship is not considered by any of the existing event models. With respect to representing different interpretations of the same event (cf. requirements in Section 3), we find an interesting concept of "constellations" introduced in E. However, constellations are only planned for E and thus not applicable. Finally, we can state that F supports all the requirements to serve as a common model of events.

The notion of events is different in the existing event models and not always explicitly stated. Eventory, the Event Ontology, and E consider events as four-dimensional entities. VERL and CIDOC CRM explicitly follow the separation of events and objects as introduced in Section 2.1 and indicate that the authors studied foundational literature. However, the literature studied is not revealed and no consequences are drawn for their modeling approach. In general, one can say that the existing event models are developed "ad hoc". Consequently, they are often ambiguous and lack a formal basis. Some ambiguities in E [32] with respect to naming conventions and ontological relationships of key concepts have recently been removed [25]. However, the principal problem of being an informal, ad-hoc created event model remains. Such a formal basis can be provided by using an upper-level ontology such as DOLCE [10, 16] or SUMO [21]. The purpose of such ontologies is not to provide a fully fledged event model as our goal is in this work. However, these ontologies come with formal definitions of fundamental concepts such as events and objects and provide axioms that can be used and extended. Thus, they provide a solid modeling basis for reference ontologies such as our event model F and ensure good design. None of the existing event models leverage an upper-level ontology for event modeling. They also do not follow a pattern-oriented approach. This is very unfortunate as, e.g., DOLCE has already proven to provide a good design basis and modeling approach for different ontologies such as [1, 9, 19, 18]. Finally, we should mention that only the Event Ontology reuses existing ontologies (for modeling its constitutive, temporal, spatial aspects). A feature that is also supported by our event model F.

7. Conclusions

We have applied a pattern-oriented approach to develop a formal event model F. Our model provides support for the different aspects of events, namely constitutive, spatial, temporal, experiential, and structural relationships. The latter comprises the mereological relationships, causal relationships (by possibly using different underlying causal theories), and event correlation. Finally, the event model supports representing different interpretations of the same event. The full support for the structural aspect as well as different event interpretations distinguishes F very much from existing event models. Separating the different aspects of events into more easily manageable patterns allows to master the complexity of events, even for the difficult mereological and causal relationships and for event interpretations. The event model F has been created in OWL and is available online.[5] An example of an event description modeled in F and further information is available at the event model F website.[6] The axiomatization of our event model F in Description Logics [3] and a general discussion about how to axiomatize a pattern-oriented foundational ontology like F is provided in the appendix.

Acknowledgment. We thank our student Chantal Neuhaus for her support in implementing the F ontology examples. This research has been co-funded by the EU in FP6 in the NoE K-Space (027026) and X-Media project (026978) and FP7 in the WeKnowIt project (215453).

Appendix

A. Event Model F Axioms

In the following, we describe the axioms of our event model F in Description Logics [3] along the different patterns for participation, composition, causality, correlation, and interpretation. We discuss shortly in Appendix B what future extensions of Semantic Web languages would be useful for models such as F, for instance in order two specify *constraints* on the patterns.

[5] `http://events.semantic-multimedia.org/ontology/2008/12/15/model.owl`

[6] `http://www.uni-koblenz-landau.de/koblenz/fb4/institute/IFI/AGStaab/Research/events`

For better understanding, we have renamed (specialized) the DOLCE concept `EventType` to `EventRole`, and formulate that formally as

$$EventType \equiv EventRole$$

A.1. Participation Pattern

The participation pattern is discussed in Section 4.1. It describes the set of objects that participate in an event and are relevant in a given context. The participation pattern defines that for one event there has to be at least one participant. We formalize this with the following set of axioms:

$$\begin{aligned}
EventParticipationDescription &\sqsubseteq Description \\
EventParticipationDescription &\sqsubseteq \forall defines.(Participant \sqcup DescribedEvent) \\
EventParticipationDescription &\sqsubseteq \geq 1(defines.Participant) \\
EventParticipationDescription &\sqsubseteq = 1(defines.DescribedEvent) \\
EventParticipationDescription &\sqsubseteq = 1(satisfiedBy.EventParticipationSituation) \\
EventParticipationSituation &\sqsubseteq Situation \\
EventParticipationSituation &\sqsubseteq \forall includesEvent.(\\
&\qquad \exists isClassifiedBy.DescribedEvent) \\
EventParticipationSituation &\sqsubseteq \forall includesObject.(\\
&\qquad \exists isClassifiedBy.Participant) \\
EventParticipationSituation &\sqsubseteq = 1(satisfies.EventParticipationDescription) \\
DescribedEvent &\sqsubseteq EventRole \\
DescribedEvent &\sqsubseteq \forall classifies.(\exists isEventIncludedIn. \\
&\qquad EventParticipationSituation) \\
DescribedEvent &\sqsubseteq = 1(isDefinedIn.EventParticipationDescription) \\
Participant &\sqsubseteq Role \\
Participant &\sqsubseteq \forall classifies.(\exists isObjectIncludedIn. \\
&\qquad EventParticipationSituation) \\
Participant &\sqsubseteq = 1(isDefinedIn.EventParticipationDescription)
\end{aligned}$$

A.2. Composition Pattern

The composition pattern defines how events are composed, i.e., it basically describes a part-whole relationship between events that is valid in a certain context and is possibly subject to a set of constraints (cf. Section 4.2). We require exactly one composite event, i.e., the whole, and at least one component, i.e., the part. The specification of constraints is optional.

$$
\begin{aligned}
EventCompositionDescription &\sqsubseteq Description\\
EventCompositionDescription &\sqsubseteq \forall defines.(Composite \sqcup Component\\
&\qquad \sqcup EventCompositionConstraint)\\
EventCompositionDescription &\sqsubseteq = 1(defines.Composite)\\
EventCompositionDescription &\sqsubseteq \geq 1(defines.Component)\\
EventCompositionDescription &\sqsubseteq = 1(satisfiedBy.EventCompositionSituation)
\end{aligned}
$$

$$
\begin{aligned}
EventCompositionSituation &\sqsubseteq Situation\\
EventCompositionSituation &\sqsubseteq \forall includesEvent.(\exists isClassifiedBy.\\
&\quad (Composite \sqcup Component))\\
EventCompositionSituation &\sqsubseteq \forall includesSpace.(\exists isParametrizedBy.SpatialConstraint)\\
EventCompositionSituation &\sqsubseteq \forall includesTime.(\exists isParametrizedBy.TemporalConstraint)\\
EventCompositionSituation &\sqsubseteq \forall includesSpaceTime.\\
&\quad (\exists isParametrizedBy.SpatioTemporalConstraint)\\
EventCompositionSituation &\sqsubseteq = 1(satisfies.EventCompositionDescription)\\
Composite &\sqsubseteq EventRole\\
Composite &\sqsubseteq \forall classifies.(\exists isEventIncludedIn.\\
&\quad EventCompositionSituation)\\
Composite &\sqsubseteq = 1(isDefinedIn.EventCompositionDescription)\\
Component &\sqsubseteq EventRole\\
Component &\sqsubseteq \forall classifies.(\exists isEventIncludedIn.\\
&\quad EventCompositionSituation)\\
Component &\sqsubseteq = 1(isDefinedIn.EventCompositionDescription)\\
EventCompositionConstraint &\sqsubseteq Parameter\\
EventCompositionConstraint &\sqsubseteq = 1(isDefinedIn.EventComposition)\\
EventCompositionConstraint &\sqsubseteq \forall parametrizes.(\exists hasSetting.\\
&\quad EventCompositionSituation)\\
SpatialConstraint &\sqsubseteq EventCompositionConstraint\\
SpatialConstraint &\sqsubseteq \forall parametrizes.SpaceRegion\\
TemporalConstraint &\sqsubseteq EventCompositionConstraint\\
TemporalConstraint &\sqsubseteq \forall parametrizes.TimeRegion\\
SpatioTemporalConstraint &\sqsubseteq EventCompositionConstraint\\
SpatioTemporalConstraint &\sqsubseteq \forall parametrizes.SpatioTemporalRegion
\end{aligned}
$$

A.3. Causality Pattern

The causality pattern (cf. Section 4.3) defines a causal relationship by exactly one cause, exactly one effect, and exactly one justification, which classifies a description.

$$
\begin{aligned}
EventCausalityDescription &\sqsubseteq Description\\
EventCausalityDescription &\sqsubseteq \forall defines.(Cause \sqcup Effect \sqcup Justification)\\
EventCausalityDescription &\sqsubseteq = 1(defines.Cause)
\end{aligned}
$$

$$
\begin{aligned}
EventCausalityDescription &\sqsubseteq = 1(defines.Effect)\\
EventCausalityDescription &\sqsubseteq = 1(defines.Justification)\\
EventCausalityDescription &\sqsubseteq = 1(satisfiedBy.EventCausalitySituation)\\
EventCausalitySituation &\sqsubseteq Situation\\
EventCausalitySituation &\sqsubseteq = 1(includesEvent.(\exists isClassifiedBy.Cause))\\
EventCausalitySituation &\sqsubseteq = 1(includesEvent.(\exists isClassifiedBy.Effect))\\
EventCausalitySituation &\sqsubseteq = 1(includesObject.(\exists isClassifiedBy.Justification))\\
EventCausalitySituation &\sqsubseteq = 1(satisfies.EventCausalityDescription)\\
Cause &\sqsubseteq EventRole\\
Cause &\sqsubseteq \forall classifies.(\exists isEventIncludedIn.\\
&\qquad EventCausalitySituation)\\
Cause &\sqsubseteq = 1(isDefinedIn.EventCausalityDescription)\\
Effect &\sqsubseteq EventRole\\
Effect &\sqsubseteq \forall classifies.(\exists isEventIncludedIn.\\
&\qquad EventCausalitySituation)\\
Effect &\sqsubseteq = 1(isDefinedIn.EventCausalityDescription)\\
Justification &\sqsubseteq Role\\
Justification &\sqsubseteq \forall classifies.(Description \sqcap \exists isObjectIncludedIn.\\
&\qquad (EventCausalitySituation \sqcup\\
&\qquad EventCorrelationSituation)\\
Justification &\sqsubseteq = 1(isDefinedIn.(EventCausalityDescription \sqcup\\
&\quad EventCorrelationDescription))
\end{aligned}
$$

A.4. Correlation Pattern

The correlation pattern describes the correlation of a set of events, as discussed in Section 4.4. It only makes sense to specify a correlation between two or more events. Further, the correlation descriptions also refer to the justification defined for the causality pattern.

$$
\begin{aligned}
EventCorrelationDescription &\sqsubseteq Description\\
EventCorrelationDescription &\sqsubseteq \forall defines.(Correlate \sqcup Justification)\\
EventCorrelationDescription &\sqsubseteq \geq 2(defines.Correlate)\\
EventCorrelationDescription &\sqsubseteq = 1(defines.Justification)\\
EventCorrelationDescription &\sqsubseteq = 1(satisfiedBy.EventCorrelationSituation)
\end{aligned}
$$

$$
\begin{aligned}
EventCorrelationSituation &\sqsubseteq Situation\\
EventCorrelationSituation &\sqsubseteq \geq 2(includesEvent.(\exists isClassifiedBy.Correlate))\\
EventCorrelationSituation &\sqsubseteq = 1(includesObject.(\exists isClassifiedBy.Justification))\\
EventCorrelationSituation &\sqsubseteq = 1(satisfies.EventCausalityDescription)\\
Correlate &\sqsubseteq EventRole\\
Correlate &\sqsubseteq \forall classifies.(\exists isEventIncludedIn.\\
&\qquad EventCorrelationSituation)\\
Correlate &\sqsubseteq = 1(isDefinedIn.EventCorrelationDescription)
\end{aligned}
$$

A.5. Interpretation Pattern

The interpretation pattern defines an interpretation of exactly one event. Therefore, it provides the means to specify all those patterns for an event that are relevant for the interpretation. We have discussed the pattern in Section 4.5, and give the formal axiomatization in the following.

$$
\begin{aligned}
EventInterpretationDescription &\sqsubseteq Description\\
EventInterpretationDescription &\sqsubseteq \forall defines.(Interpretant \sqcup RelevantSituation)\\
EventInterpretationDescription &\sqsubseteq = 1(defines.Interpretant)\\
EventInterpretationDescription &\sqsubseteq \geq 1(defines.RelevantSituation)\\
EventInterpretationDescription &\sqsubseteq = 1(satisfiedBy.EventInterpretationSituation)\\
EventInterpretationSituation &\sqsubseteq Situation\\
EventInterpretationSituation &\sqsubseteq = 1(includesEvent.(\exists isClassifiedBy.Interpretant))\\
EventInterpretationSituation &\sqsubseteq \geq 1(includesObject.(Situation \sqcap \exists\\
&\qquad isClassifiedBy.RelevantSituation))\\
EventInterpretationSituation &\sqsubseteq = 1(satisfies.EventInterpretationDescription)\\
Interpretant &\sqsubseteq EventRole\\
Interpretant &\sqsubseteq \forall classifies.(\exists isEventIncludedIn.\\
&\qquad EventInterpretationSituation)\\
Interpretant &\sqsubseteq = 1(isDefinedIn.EventInterpretationDescription)\\
RelevantSituation &\sqsubseteq Role\\
RelevantSituation &\sqsubseteq \forall classifies.(Situation \sqcap\\
&\quad \exists isObjectIncludedIn.EventInterpretationSituation)\\
RelevantSituation &\sqsubseteq = 1(isDefinedIn.EventInterpretationDescription)
\end{aligned}
$$

B. Discussion of our Axiomatization

Our axiomatization is based on Description Logics, since the ontology itself is formulated in OWL [20]. A human reader will instantly understand the meaning of most of the axioms, e.g., that in the case of the causality pattern there *must* be a cause and an effect and some justification for this causal relationship. However, an OWL reasoner will not detect an inconsistency if, e.g., a causality pattern does not specify the cause. This is due to the open world assumption that underlies Description Logics and more generally the Semantic Web. Instead, the reasoner would infer that there must be some cause that is currently unknown.

In general, this assumption is advantageous in the Semantic Web since we have to deal with incomplete knowledge. Nevertheless, in certain cases, and a formal model like F is such a case, a closed world is desired. The information that there is some cause without specifying it is not really useful for processing events. Even in the case of hypothetical events that *could* be the cause for some other event, we want to require that the hypothetical event is explicitly represented with the aim of being able to talk about it. This is of pragmatic importance for an information system as otherwise the existence of the cause is not justified.

For such cases and for ontologies such as F, COMM [1], or X-COSIM [9], it would be beneficial if we could "close" certain parts of the world and thus state that the knowledge in this small worlds has to be complete. An example of such an approach is presented in [12], where the **K**-operator is introduced for Description Logics. This operator can be interpreted as *known to be*. If we extend the axiom of the causality pattern with this operator, we would get the following axiom

$$EventCausalityDescription \sqsubseteq = 1(\mathbf{K}defines.Cause),$$

which is interpreted as: An `EventCausalityDescription` must define exactly one *known* cause. The difference is that in this case we must know the individual, and if we don't find one, the reasoner will detect an inconsistency.

Such extensions are useful to close the patterns and provide some automatic means to detect inconsistencies in the *data*. However, they do not solve all problems. For instance, currently we are also not able to specify that a pattern has to be closed. We cannot say that the role defined by a description has to classify an object that is included in *exactly that* situation that satisfies the description. We can only specify that it has to classify *some* object that is included in a compatible situation. Theoretically, this could be a situation that does not satisfy the correct description. In other words, it is possible to generate inconsistent data.

To summarize, while we can formally define the patterns of in Description Logics, we currently do not have the means to fully check instances of them for consistency, e.g., if there is an effect than there also has to be a cause. This has to be and can only be guaranteed by the code implementing a concrete system. However, especially in distributed systems, we have no means to check whether the generated data is valid. Although the open-world assumption is very important for the Semantic Web in general, we argue that extensions to Description Logics like the **K**-operator are very useful in certain cases such as with our event model F and thus should be considered in the future.

References

[1] Richard Arndt, Raphael Troncy, Steffen Staab, Lynda Hardman, and Miroslav Vacura. COMM: Designing a well-founded multimedia ontology for the web. In *ISWC*. Springer, 2007.

[2] V. Arnold. *Mathematical Methods of Classical Mechanics*. Springer, 1997.

[3] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[4] Rolando Blanco, Jun Wang, and Paulo Alencar. A metamodel for distributed event based systems. In *Distributed event-based systems*. ACM, 2008.

[5] Roberto Casati and Achille Varzi. Events. Stanford Encyclopedia of Philosophy, 2006. http://plato.stanford.edu/entries/events/.

[6] Martin Doerr, Christian-Emil Ore, and Stephen Stead. The CIDOC conceptual reference model. In *Int. Conf. on Conceptual modeling*. Australian Computer Society, Inc., 2007.

[7] Ahmet Ekin, A. Murat Tekalp, and Rajiv Mehrotra. Integrated semantic-syntactic video modeling for search and browsing. *IEEE Transactions on Multimedia*, 6(6), 2004.

[8] Alexandre R. J. Francois, Ram Nevatia, Jerry Hobbs, and Robert C. Bolles. VERL: An ontology framework for representing and annotating video events. *IEEE MultiMedia*, 12(4), 2005.

[9] Thomas Franz, Steffen Staab, and Richard Arndt. The X-COSIM integration framework for a seamless semantic desktop. In *Int. Conf. on Knowledge capture*. ACM, 2007.

[10] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening Ontologies with DOLCE. In *EKAW*. Springer, 2002.

[11] Aldo Gangemi and Peter Mika. Understanding the semantic web through descriptions and situations. In *CoopIS/DOA/ODBASE*, pages 689–706, 2003.

[12] Stephan Grimm, Boris Motik, and Chris Preist. Matching Semantic Service Descriptions with Local Closed-World Reasoning. In *ESWC*. Springer, 2006.

[13] Esa Itkonen. *Causality in Linguistic Theory*. Indiana Univ. Press, 1983.

[14] Ramesh Jain. Eventweb: Developing a human-centered computing system. *Computer*, 41(2), 2008.

[15] Lawrence Lombard. *Events: A metaphysical study*. Routledge & Kegan Paul, 1986.

[16] Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, and Alessandro Oltramari. WonderWeb deliverable D18 ontology library (final). Technical report, IST Project 2001-33052 WonderWeb: Ontology Infrastructure for the Semantic Web, 2003.

[17] Ram Nevatia, Jerry Hobbs, and Bob Bolles. An ontology for video event representation. In *Computer Vision and Pattern Recognition*. IEEE, 2004.

[18] Daniel Oberle, Anupriya Ankolekar, Pascal Hitzler, and et al. DOLCE ergo SUMO: On foundational & domain models in the SmartWeb Integrated Ontology. *Web Semant.*, 5(3), 2007.

[19] Daniel Oberle, Steffen Lamparter, S. Grimm, D. Vrandečić, S. Staab, and A. Gangemi. Towards ontologies for formalizing modularization and communication in large software systems. *Appl. Ontol.*, 1(2):163–202, 2006.

[20] Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks. OWL web ontology language semantics and abstract syntax. Technical report, W3C, 2004.

[21] Adam Pease, Ian Niles, and John Li. The suggested upper merged ontology. In *Workshop on Ontologies and the Semantic Web*, 2002.

[22] Anthony Quinton. Objects and events. *Mind*, 88(350), 1979.

[23] Yves Raimond and Samer Abdallah. The event ontology, 2007. http://motools.sf.net/event.

[24] Szabolcs Rozsnyai, Josef Schiefer, and Alexander Schatten. Concepts and models for typing events for event-based systems. In *Distributed Event-Based Systems*. ACM, 2007.

[25] A. Scherp, S. Agaram, and R. Jain. Event-centric media management. In *Multimedia Content Access: Algorithms and Systems II*, volume 6820 of *SPIE*, 2008.

[26] Glenn Shafer. Causal logic. In *ECAI*. Wiley, 1998.

[27] Bill Shipley. *Cause and Correlation in Biology*. Cambridge Univ. Press, 2002.

[28] P. Sinclair, M. Addis, F. Choi, and et al. The use of CRM core in multimedia annotation. In *Workshop on Semantic Web Annotations for Multimedia*, 2006.

[29] W3C. Time ontology in OWL, 2006. http://www.w3.org/TR/owl-time/.

[30] W3C. WGS84 geo positioning, 2006. http://www.w3.org/2003/01/geo/wgs84_pos.

[31] Xiang Wang, Swathi Mamadgi, Atit Thekdi, Aisling Kelliher, and Hari Sundaram. Eventory – an event based media repository. In *Int. Conf. on Semantic Computing*. IEEE, 2007.

[32] Utz Westermann and Ramesh Jain. Toward a common event model for multimedia applications. *IEEE MultiMedia*, 14(1), 2007.

Bisher erschienen

Arbeitsberichte aus dem Fachbereich Informatik
(http://www.uni-koblenz.de/fb4/publikationen/arbeitsberichte)

Ansgar Scherp, Thomas Franz, Carsten Saathoff, Steffen Staab, F-A Model of Events based on a Foundational Ontology, Arbeitsberichte aus dem Fachbereich Informatik 2/2009

Frank Bohdanovicz, Harald Dickel, Christoph Steigner, Avoidance of Routing Loops, Arbeitsberichte aus dem Fachbereich Informatik 1/2009

Stefan Ameling, Stephan Wirth, Dietrich Paulus, Methods for Polyp Detection in Colonoscopy Videos: A Review, Arbeitsberichte aus dem Fachbereich Informatik 14/2008

Tassilo Horn, Jürgen Ebert, Ein Referenzschema für die Sprachen der IEC 61131-3, Arbeitsberichte aus dem Fachbereich Informatik 13/2008

Thomas Franz, Ansgar Scherp, Steffen Staab, Does a Semantic Web Facilitate Your Daily Tasks?, Arbeitsberichte aus dem Fachbereich Informatik 12/2008

Norbert Frick, Künftige Anfordeungen an ERP-Systeme: Deutsche Anbieter im Fokus, Arbeitsberichte aus dem Fachbereicht Informatik 11/2008

Jürgen Ebert, Rüdiger Grimm, Alexander Hug, Lehramtsbezogene Bachelor- und Masterstudiengänge im Fach Informatik an der Universität Koblenz-Landau, Campus Koblenz, Arbeitsberichte aus dem Fachbereich Informatik 10/2008

Mario Schaarschmidt, Harald von Kortzfleisch, Social Networking Platforms as Creativity Fostering Systems: Research Model and Exploratory Study, Arbeitsberichte aus dem Fachbereich Informatik 9/2008

Bernhard Schueler, Sergej Sizov, Steffen Staab, Querying for Meta Knowledge, Arbeitsberichte aus dem Fachbereich Informatik 8/2008

Stefan Stein, Entwicklung einer Architektur für komplexe kontextbezogene Dienste im mobilen Umfeld, Arbeitsberichte aus dem Fachbereich Informatik 7/2008

Matthias Bohnen, Lina Brühl, Sebastian Bzdak, RoboCup 2008 Mixed Reality League Team Description, Arbeitsberichte aus dem Fachbereich Informatik 6/2008

Bernhard Beckert, Reiner Hähnle, Tests and Proofs: Papers Presented at the Second International Conference, TAP 2008, Prato, Italy, April 2008, Arbeitsberichte aus dem Fachbereich Informatik 5/2008

Klaas Dellschaft, Steffen Staab, Unterstützung und Dokumentation kollaborativer Entwurfs- und Entscheidungsprozesse, Arbeitsberichte aus dem Fachbereich Informatik 4/2008

Rüdiger Grimm: IT-Sicherheitsmodelle, Arbeitsberichte aus dem Fachbereich Informatik 3/2008

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik 2/2008

Markus Maron, Kevin Read, Michael Schulze: CAMPUS NEWS – Artificial Intelligence Methods Combined for an Intelligent Information Network, Arbeitsberichte aus dem Fachbereich Informatik 1/2008

Lutz Priese,Frank Schmitt, Patrick Sturm, Haojun Wang: BMBF-Verbundprojekt 3D-RETISEG Abschlussbericht des Labors Bilderkennen der Universität Koblenz-Landau, Arbeitsberichte aus dem Fachbereich Informatik 26/2007

Stephan Philippi, Alexander Pinl: Proceedings 14. Workshop 20.-21. September 2007 Algorithmen und Werkzeuge für Petrinetze, Arbeitsberichte aus dem Fachbereich Informatik 25/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS – an Intelligent Bluetooth-based Mobile Information Network, Arbeitsberichte aus dem Fachbereich Informatik 24/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS - an Information Network for Pervasive Universities, Arbeitsberichte aus dem Fachbereich Informatik 23/2007

Lutz Priese: Finite Automata on Unranked and Unordered DAGs Extented Version, Arbeitsberichte aus dem Fachbereich Informatik 22/2007

Mario Schaarschmidt, Harald F.O. von Kortzfleisch: Modularität als alternative Technologie- und Innovationsstrategie, Arbeitsberichte aus dem Fachbereich Informatik 21/2007

Kurt Lautenbach, Alexander Pinl: Probability Propagation Nets, Arbeitsberichte aus dem Fachbereich Informatik 20/2007

Rüdiger Grimm, Farid Mehr, Anastasia Meletiadou, Daniel Pähler, Ilka Uerz: SOA-Security, Arbeitsberichte aus dem Fachbereich Informatik 19/2007

Christoph Wernhard: Tableaux Between Proving, Projection and Compilation, Arbeitsberichte aus dem Fachbereich Informatik 18/2007

Ulrich Furbach, Claudia Obermaier: Knowledge Compilation for Description Logics, Arbeitsberichte aus dem Fachbereich Informatik 17/2007

Fernando Silva Parreiras, Steffen Staab, Andreas Winter: TwoUse: Integrating UML Models and OWL Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 16/2007

Rüdiger Grimm, Anastasia Meletiadou: Rollenbasierte Zugriffskontrolle (RBAC) im Gesundheitswesen, Arbeitsberichte aud dem Fachbereich Informatik 15/2007

Ulrich Furbach, Jan Murray, Falk Schmidsberger, Frieder Stolzenburg: Hybrid Multiagent Systems with Timed Synchronization-Specification and Model Checking, Arbeitsberichte aus dem Fachbereich Informatik 14/2007

Björn Pelzer, Christoph Wernhard: System Description:“E-KRHyper“, Arbeitsberichte aus dem Fachbereich Informatik, 13/2007

Ulrich Furbach, Peter Baumgartner, Björn Pelzer: Hyper Tableaux with Equality, Arbeitsberichte aus dem Fachbereich Informatik, 12/2007

Ulrich Furbach, Markus Maron, Kevin Read: Location based Informationsystems, Arbeitsberichte aus dem Fachbereich Informatik, 11/2007

Philipp Schaer, Marco Thum: State-of-the-Art: Interaktion in erweiterten Realitäten, Arbeitsberichte aus dem Fachbereich Informatik, 10/2007

Ulrich Furbach, Claudia Obermaier: Applications of Automated Reasoning, Arbeitsberichte aus dem Fachbereich Informatik, 9/2007

Jürgen Ebert, Kerstin Falkowski: A First Proposal for an Overall Structure of an Enhanced Reality Framework, Arbeitsberichte aus dem Fachbereich Informatik, 8/2007

Lutz Priese, Frank Schmitt, Paul Lemke: Automatische See-Through Kalibrierung, Arbeitsberichte aus dem Fachbereich Informatik, 7/2007

Rüdiger Grimm, Robert Krimmer, Nils Meißner, Kai Reinhard, Melanie Volkamer, Marcel Weinand, Jörg Helbach: Security Requirements for Non-political Internet Voting, Arbeitsberichte aus dem Fachbereich Informatik, 6/2007

Daniel Bildhauer, Volker Riediger, Hannes Schwarz, Sascha Strauß, „grUML – Eine UML-basierte Modellierungssprache für T-Graphen“, Arbeitsberichte aus dem Fachbereich Informatik, 5/2007

Richard Arndt, Steffen Staab, Raphaël Troncy, Lynda Hardman: Adding Formal Semantics to MPEG-7: Designing a Well Founded Multimedia Ontology for the Web, Arbeitsberichte aus dem Fachbereich Informatik, 4/2007

Simon Schenk, Steffen Staab: Networked RDF Graphs, Arbeitsberichte aus dem Fachbereich Informatik, 3/2007

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik, 2/2007

Anastasia Meletiadou, J. Felix Hampe: Begriffsbestimmung und erwartete Trends im IT-Risk-Management, Arbeitsberichte aus dem Fachbereich Informatik, 1/2007

„Gelbe Reihe“
(http://www.uni-koblenz.de/fb4/publikationen/gelbereihe)

Lutz Priese: Some Examples of Semi-rational and Non-semi-rational DAG Languages. Extended Version, Fachberichte Informatik 3-2006

Kurt Lautenbach, Stephan Philippi, and Alexander Pinl: Bayesian Networks and Petri Nets, Fachberichte Informatik 2-2006

Rainer Gimnich and Andreas Winter: Workshop Software-Reengineering und Services, Fachberichte Informatik 1-2006

Kurt Lautenbach and Alexander Pinl: Probability Propagation in Petri Nets, Fachberichte Informatik 16-2005

Rainer Gimnich, Uwe Kaiser, and Andreas Winter: 2. Workshop "Reengineering Prozesse" – Software Migration, Fachberichte Informatik 15-2005

Jan Murray, Frieder Stolzenburg, and Toshiaki Arai: Hybrid State Machines with Timed Synchronization for Multi-Robot System Specification, Fachberichte Informatik 14-2005

Reinhold Letz: FTP 2005 – Fifth International Workshop on First-Order Theorem Proving, Fachberichte Informatik 13-2005

Bernhard Beckert: TABLEAUX 2005 – Position Papers and Tutorial Descriptions, Fachberichte Informatik 12-2005

Dietrich Paulus and Detlev Droege: Mixed-reality as a challenge to image understanding and artificial intelligence, Fachberichte Informatik 11-2005

Jürgen Sauer: 19. Workshop Planen, Scheduling und Konfigurieren / Entwerfen, Fachberichte Informatik 10-2005

Pascal Hitzler, Carsten Lutz, and Gerd Stumme: Foundational Aspects of Ontologies, Fachberichte Informatik 9-2005

Joachim Baumeister and Dietmar Seipel: Knowledge Engineering and Software Engineering, Fachberichte Informatik 8-2005

Benno Stein and Sven Meier zu Eißen: Proceedings of the Second International Workshop on Text-Based Information Retrieval, Fachberichte Informatik 7-2005

Andreas Winter and Jürgen Ebert: Metamodel-driven Service Interoperability, Fachberichte Informatik 6-2005

Joschka Boedecker, Norbert Michael Mayer, Masaki Ogino, Rodrigo da Silva Guerra, Masaaki Kikuchi, and Minoru Asada: Getting closer: How Simulation and Humanoid League can benefit from each other, Fachberichte Informatik 5-2005

Torsten Gipp and Jürgen Ebert: Web Engineering does profit from a Functional Approach, Fachberichte Informatik 4-2005

Oliver Obst, Anita Maas, and Joschka Boedecker: HTN Planning for Flexible Coordination Of Multiagent Team Behavior, Fachberichte Informatik 3-2005

Andreas von Hessling, Thomas Kleemann, and Alex Sinner: Semantic User Profiles and their Applications in a Mobile Environment, Fachberichte Informatik 2-2005

Heni Ben Amor and Achim Rettinger: Intelligent Exploration for Genetic Algorithms – Using Self-Organizing Maps in Evolutionary Computation, Fachberichte Informatik 1-2005