Ontologies for Complex Event Processing

Irina Astrova, Arne Koschel, Jan Lukanowski, Jose Luis Munoz Martinez, Valerij Procenko, Marc Schaaf

***Abstract*—**In this paper, five ontologies are described, which include the event concepts. The paper provides an overview and comparison of existing event models. The main criteria for comparison are that there should be possibilities to model events with stretch in the time and location and participation of objects; however, there are other factors that should be taken into account as well. The paper also shows an example of using ontologies in complex event processing.

***Keywords*—**Ontologies, events, complex event processing (CEP).

I. Introduction

THE investigation of events are of increasing importance because one would like to generate new knowledge from existing knowledge. But individual events oftentimes have little or no meaning. Only when one looks at the events together, it results in a sense that we can interpret. There is currently a lot of work on systems that can analyze the events, e.g., complex event processing (CEP). The ontologies are an important component in this development [33], [34], [31], since ontologies are strategies to describe semantic models. Furthermore, in ontologies it can be provided common comprehension of (possibly multiple domain) knowledge, common recognized terminologies can be determined on certain domains, properties can be implemented and restrictions and axioms can be formulated at different ways and levels [36].

The rest of the paper is organized as follows. At the beginning of the paper, there is a small introduction to the topic ontology. It will briefly present the main concepts (Section II). Here the term of ontology is also introduced. Additionally, the term event is defined. Event will be understood in different ways and differently defined. Therefore, one must firstly define a definition of the event. The selection of ontologies has been dependent of this event - definition. At the end of Section II there is an overview of ontologies, which include events. In Section III some ontologies are presented. These ontologies can be practically used in some projects. One of such projects is the EU project WeKnowIt [8]. In Section IV it is presented the use of ontologies in a particular project that is already being used. At the end of the paper, Section V compares the ontologies, by summarizing their commonalities and differences.

Irina Astrova is with the Institute of Cybernetics, Tallinn University of Technology, Estonia (e-mail: irina@cs.ioc.ee).

Arne Koschel, Jan Lukanowski, Jose Luis Munoz Martinez, and Valerij Procenko are with the Faculty IV, Department for Computer Science, University of Applied Sciences and Arts Hannover, Germany (e-mail: akoschel@acm.org).

Marc Schaaf is with the Institute for Information Systems, University of Applied Sciences Northwestern Switzerland, Olten, Switzerland (e-mail: marc.schaaf@fhnw.ch).

A. Complex Events

The systems based on events are correlated with message-oriented middleware (MOM) platforms. Within those systems, it is permitted the processing of events making that possible by different ways. One of them is a notification that is done to maintain informed participants of event monitoring. Notifications facilitate mechanisms to make decisions in function of event's context. In fact, there are no problems by generating, capturing or notification of such events, but the processing of many sort of events produced in real time and correlation between them generating automated responses based on event semantics.

B. Knowledge in Real Time

As we know, in the reality there are many cases with retardation or latency in several seconds. This means that there might be severe losses in organizations regarding the economic. CEP permits the obtaining of event knowledge at real time. It is able to register and process in real time events that were generated by agents or producers.

C. Lack of Background Knowledge

CEP is able to process streams of events generated by producers, store and classify events in repositories taking into account the temporal dimension (i.e., when events were produced). CEP has mechanisms to detect complex patterns and generate automatically responses to consumer patterns that were detected.

II. Basics

In this section, we'll briefly introduced definitions, which will be used throughout the whole paper.

A. Events

An event is anything that has occurred in a certain time and environment where some actors could take part and show some action features [9]. An event has the following views: state-change view (the event is a change in the state of anything), happening view (the event is anything that happens or is contemplated as happening) and detectable-condition view (the event is a detectable condition that can trigger a notification [13].

The following elements are present in an event: an action that has occurred, an object that is involved in the event, the time that event lasting, the environment of the event, the assertions about pre-condition, post-condition, and intermediate them, and the language expressions [9].

B. Ontologies

One definition of ontology, which can be often found in the literature stems from Thomas Gruber. The definition is as follows: *An ontology is an explicit specification of a conceptualization, where a conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose* [23].

One difference to some other knowledge representation forms (e.g., thesaurus, taxonomy, etc.) is that new knowledge can be gained from existing content. That occurs with aim of defined rules (relations), from which logical conclusions could be deduced [27]. With ontologies the new knowledge can be automatically structured in the corresponding model of knowledge representation.

There are different types of ontologies. Ontologies can be divided into following categories: upper ontologies and domain ontologies [28]. An upper ontology is domain-independent ontology, from which more domain-specific ontologies may be derived. One example of upper ontology is OpenCyc. A domain ontology specifies concepts, which are belonging to a specific domain of interest.

There are a lot of ontologies, which include events or play event roles [35]. For example:

- **CIDOC CRM:** museums and libraries.
- **ABC Ontology:** digital libraries.
- **Event Ontology:** digital music.
- **DOLCE+DnS Ultralite:** event aspects in social reality.
- **Event-Model-F:** event-based systems.
- **VUevent Model:** An extension of DOLCE and other event conceptualizations.
- **IPTC. EventML:** structured event information.
- **GEM:** geospatial events.
- **Event MultiMedia:** multimedia.
- **LODE:** events as Linked Data.
- **CultureSampo:** Publication System of Cultural Heritage.
- **OpenCyC Ontology:** human consensus reality, upper ontology with lots of terms and assertions.
- **Super BPEL:** ontology for the business process execution language.
- **Semantic Sensor Net Ontology:** ontology for sensor networks.

These ontologies were created for different purposes. Some of those ontologies are upper ontologies (e.g., DOLCE + Ultralite DnS and OpenCyc). Some of the ontologies are well-documented. But there is very little documentation on others. The events in some of the ontologies play a central role. In other ontologies, they are only a small part. If one wants to use an ontology that describes events, then one must know which existing ontologies there are already being used. So, one needs a quick overview of available ontologies without looking at many details. Some of the ontologies are also very large and one must invest time to understand them. Therefore, this paper attempts to provide such an overview.

III. Ontologies for Complex Event Processing

In this section, we'll describe several ontologies, which include events or play event roles in more detail.

A. OpenCyc

This section describes the representation of events in the OpenCyc ontology. OpenCyc was chosen for analysis because it is a very large ontology, which offers the possibility of event modeling. This ontology offers the possibility for events to stretch over the time and space. Furthermore, events can involve participation of living or non-living actors, which extends representation possibilities of it. Additionally, it is an upper ontology, which is application-independent, and thus more domain-specific ontologies can be derived from it.

What is OpenCyc: It is the open source version of the Cyc Technology [4]. It is an upper ontology, which is used for the representation of human knowledge about the objects and events of everyday life [1]. Today the Cyc knowledge base contains nearly five hundred thousand terms, including about fifteen thousand types of relations, and about five million facts (assertions) relating these terms.

The contents of knowledge base ise described in CycL. CycL is a formal language whose syntax derives from the language of formal logic and from Lisp [14]. An example of an assertion in CycL is:

```
$#isa $#ValerijProcenko $#Student
```

where `$#isa` is an predicate (predicates start with lower case), `$#ValerijProcenko` is an individual and `$#Student` is a collection (collections and individuals start with upper case). This assertion means that an individual `Valerij Procenko` is an instance of a `Students` collection.

Events in OpenCyc: OpenCyc uses around 37,000 different event types to describe what happens in the real world [19].

The representation of events can be seen in Fig. 1. This figure shows a small overview of `$#genls1` hierarchy surrounding `$#Event`. Both `$#Event` and `$#StaticSituation` are specializations of `$#Situation-Temporal`. Therefore, both specializations are objects that extend over time. The difference between these two collections is that `$#StaticSituation` collects situations that are extended in time but do not change, whereas `$#Event` collects situations that are extended in time and do change [20]. There are ontologies that do not distinguish between these types of classification (e.g., the Event ontology). The advantage of this classification is that events can be modeled very precisely [32]. Furthermore, for both collections, the instances are represented as temporal objects and not as predicates. This is needed because events may involve things (e.g., place of event or performers) that are not possible if the action was represented by a predicate [28].

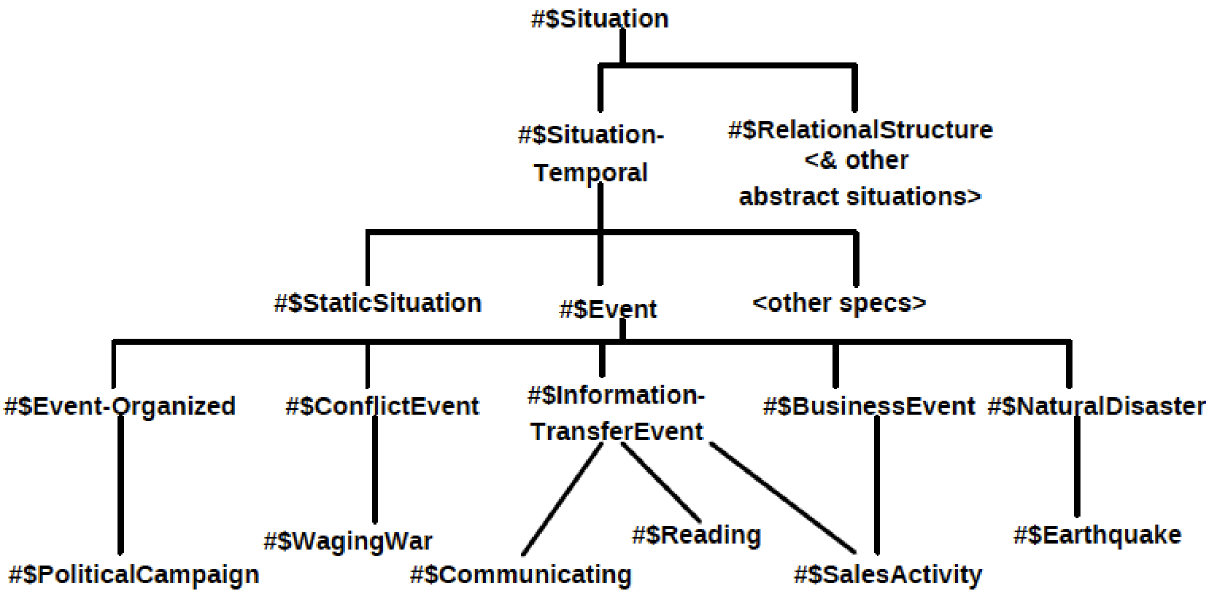

Fig. 1 Partial hierarchy of $#Event [20]

Fig. 2 shows an example of event "Battle of the Nile" and components involved in this event. One can see here that involving other things into the event offers the possibility to describe more precisely events.

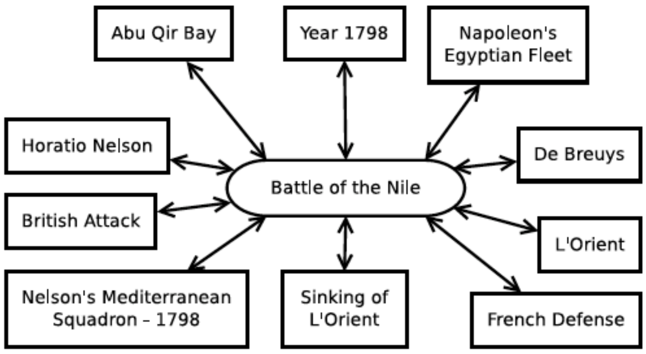

Fig. 2 Example of use of event roles [20]

Events in OpenCyc have temporal extent [38]. OpenCyc offers many ways of representation of time that belongs to the event. One can model temporal duration or specific point in time [15]. There are also many predicates for describing relations between temporal things and interrelatedness of time intervals. OpenSyc offers possibility to model absolute and relative representation of points in time. Events are also dynamic [38]. That means events have a temporal state over time.

Events have stretches of space [38]. For this purpose, OpenCyc offers many ways of describing spatial properties and relations, e.g., one can model the time point as a relative or absolute position of events. Additionally, the spatial points can be modeled with mereological relations.

In CycL can be used special predicates called roles to reason in general terms about kinds of situations and kinds of things that are involved in them [21]. These predicates are instances of the collection $#Role. Moreover, `$#Role` has two important sub-types: `$#ActorSlot` and `$#Sub-ProcessSlot`. `$#ActorSlot` defines the roles, which involve an actor in the event. Therefore, all the instances of `$#ActorSlot` are predicates that have `$#actors` as their `$#genlPreds` [16]. An example of representation in CycL can look like that:

`&#performedBy &#Writing &#ValerijProcenko`

The first argument of every instance of `$#ActorSlot` is constrained to be an instance of some specialization of `$#Event` and the second argument is constrained to be an instance of some specialization of `$#SomethingExisting` [18]. Fig. 3 shows a partial `$#ActorSlot` hierarchy.

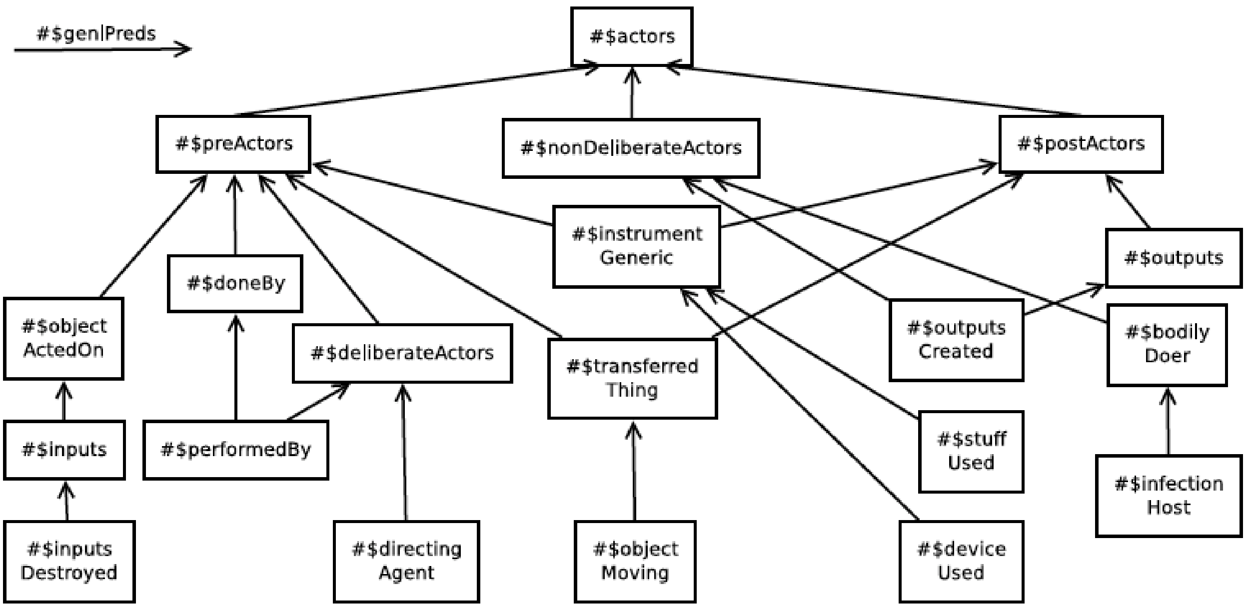

Fig. 3 Partial hierarchy of `$#ActorSlot` [16]

Another important sub-collection of `$#Role` is `$#SubProcessSlot`. This specialization allows events to have sub-events. So it is easier to model complex events, which consist of simple ones. Sub-events are events that occur within the temporal scope of a large event [17]. There are two different predicates:

- `$#subEvents` is used to model sub-events, which are involved in main-event. Sub-events are part of the main event, so one can model mereological relations.
- `$#temporallySubsumes` is a strictly temporal relationship that occurs between start and end of the main event. But events, which happening in the world at the same time point as main events are not parts of them. So it is weaker than the relation `$#subEvents` [1].

The sub-events can be related to each other. They can occur at different time points (e.g., simultaneously, in sequence or they can temporally overlap) [17].

Summary: The event modeling possibility of OpenCyc was examined. The events can be modeled over time and location. Since the ontology is very large one, it is possible to model events in many different ways. Furthermore, if one has a specific domain, then one can derive a domain-specific ontology from this upper ontology. With `$#ActorSlot` and sub-events, there are ways to model more accurately the events as well as relations between them.

B. Event

This section describes the representation of events in the Event ontology that was chosen for analysis because it is centered on the notion of event.

What is Event: The Event ontology was developed in the Center for Digital Music in Queen Mary, University of London. The aim of this development was an ontology, which can be used in conjunction with other music-related ontologies [32]. But in this ontology, there is nothing specific to the music domain so it can be used in other domains as well.

Events in Event: The top-level class in the Event ontology is oe:Event. The ontology defines event as “an arbitrary classification of a space/time region, by a cognitive agent. An event may have actively participating agents, passive factors, products, and a location in space/time” [29]. This ontology enables modeling anything with spatial extends. The fact that one must consider is that one can model spatial relation only as an absolute point and not as a relative one [10].

The Event ontology differs from some other event ontologies because it does not make any attempt to distinguish fundamental types of events. For instance, OpenCyc distinguishes between `$#StaticSituation` and `$#Event`. On the one hand, a problem with building such classification is that it is not always easy to decide what type of events has happened. On the other hand, this is desirable for precise modeling in specific domains that share descriptive paradigm [32].

The representation of events can be seen in Fig. 4. To involve things into events, the following properties are employed: `event:product`, `event:factor` and `event:agent` [29]. The `event:product` relates an event to something produced during the event. The result of the event is represented as an object `event:Product`. The `event:factor` relates an event to a passive factor. This means that it is used to provide participation things in an event and things influence upon an event. Whereby, the Event ontology “does not distinguish between a thing’s participation in an event and a thing’s influence upon an event” [32]. The `event:agent` relates event to an active agent. The spatial regions are represented as objects as well. A property `event:place` relates an event to a spatial object `geo:SpatialThing`. For spatial objects, the WGS84 Geo

Positioning Ontology [11] is used.

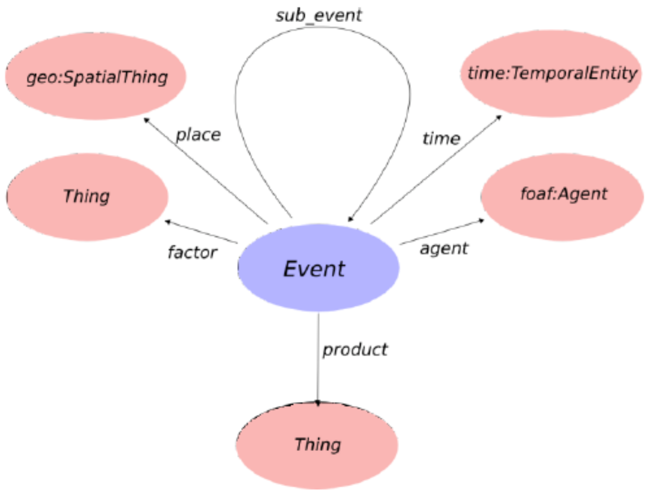

Fig. 4 Event ontology [29]

The relationship between events and timespans is represented as a property `event:time`, which relates an event to a time object [29]. For the time object, the OWL-Time ontology [24] and the Timeline ontology [30] are used. The OWL-Time ontology is also used to represent a class `time:TemporalEntity`. By using the Timeline ontology, one can also define a number of intervals and instants on the timeline. Thereto one can use the `tl:onTimeLine` property [30]. One example of the representation of timeline can be seen in Fig. 5.

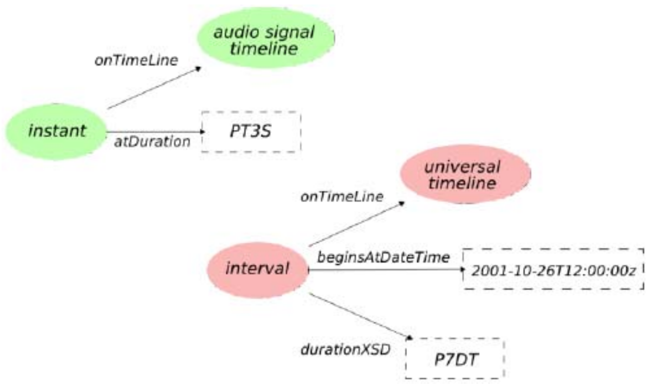

Fig. 5 Example of representation of timeline in the Timeline ontology [30]

Oftentimes one needs to model an event as part of another event because one "must distinguish between mere temporal containment and mereological relationships between sub-events and some greater event" [32]. The temporal containment is represented with temporal spans. This means that for example an event B can occur in timespan of an event A, but that does not make the event B part of the event A. For mereological relationships, the Event ontology defines event:sub event. This property splits a complex event into simpler ones [29]. However, the Event ontology provides support only for simple part-of relationships; so one cannot model more complex mereological relationships [10].

Summary: The event modeling possibility of the Event ontology was analyzed. The events can be modeled over time and location but the spatial possibility constraint only over an absolute position. The relative position cannot be modeled with this ontology. The ontology also provides properties for modeling participation in events, thus it offers many possibilities to model an event as part of some other events. However, the Event ontology does not offer possibilities to model complex mereological relationships.

C. Event-Model-F

The Event-Model-F ontology was specially designed for events. This was the main criterion for its selection.

What is Event-Model-F: The Event-Model-F ontology was developed at the University of Koblenz-Landau (Germany). This ontology is mainly focused on the processing of events [7] (see Fig. 6). The ontology is to be used in event-based systems [5, 10]. Its model is "based on the foundational ontology DOLCE+DnS Ultralight (DUL) and provides comprehensive support to represent time and space, objects and persons as well as mereological, causal, and correlative relationships between events.

In addition, the ontology provides a flexible means for event composition, modeling event causality and event correlation, and representing different interpretations of the same event" [7].

The Event-Model-F ontology permits the easy exchange of information, which is undertaken between different event-based components and systems [10].

Events in Event-Model-F: The events in this ontology involve different information. The ontology contains details about the objects involved. These can be people or other not living objects. Time point on which event occurred can also be stored. The time can be relative or absolute. Spatial position of the affected object is also entered in the event. The spatial position can be given, just like time, and might be absolute or relative time. This ontology also includes relationships between events. These can be of different types: mereological, causal and correlation relationships. The ontology provides a possibility to add documentary. So the documentary can be included in both events and objects. Furthermore, the different interpretations of an event are supported by the Event-Model-F ontology as well [10].

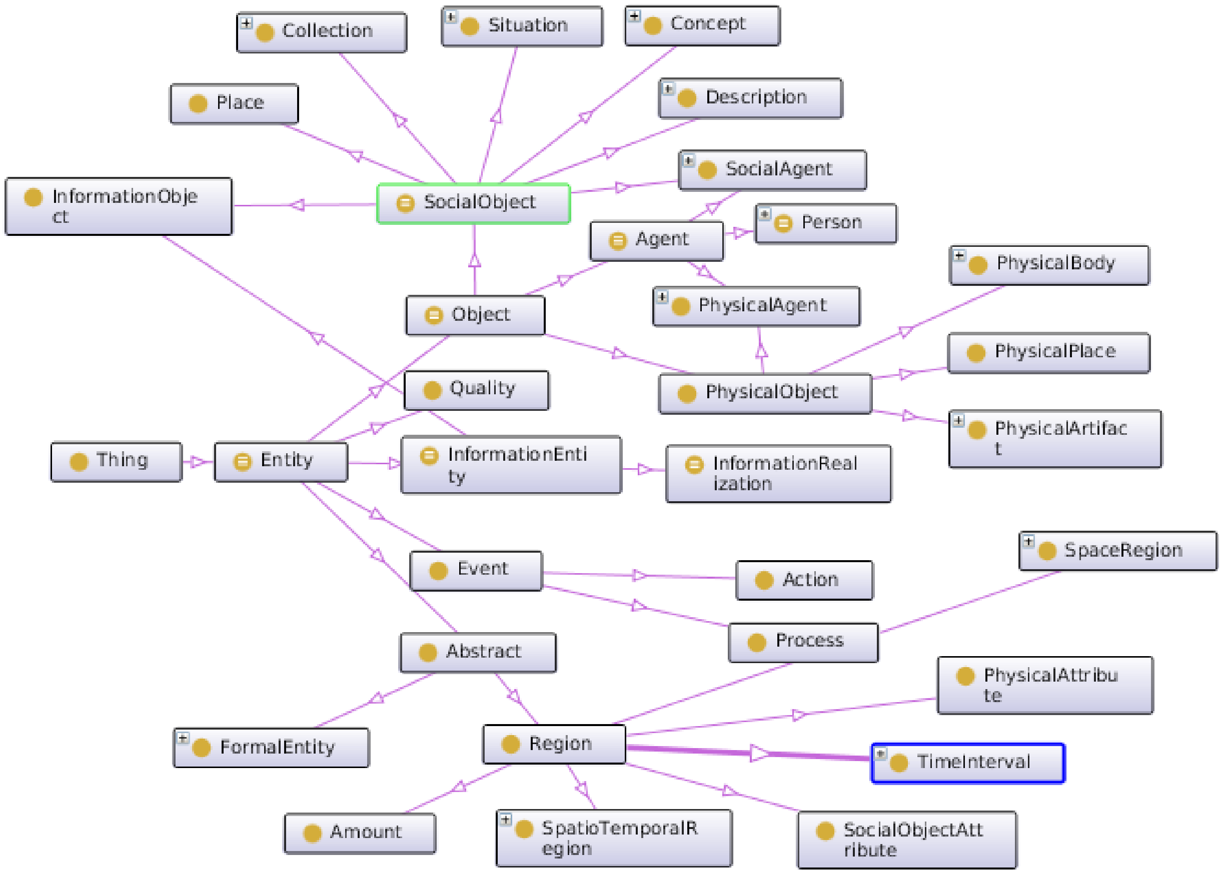

Fig. 6 Event-Model-F [6]

The representation of events can be seen in Fig. 7. Some objects were taken from the upper ontology DUL (DOLCE + DnS Ultralight). One of these classes is `DUL:Event`. An `DUL:Event` exists in a certain time. Another class is `DUL:Object`. This class represents an entity. An entity exists in a certain space. A `DUL:Object` can be a living (person) or non-living (board) entity. Yet another class is `DUL:Quality`. It is a characteristic of an entity or event [10].

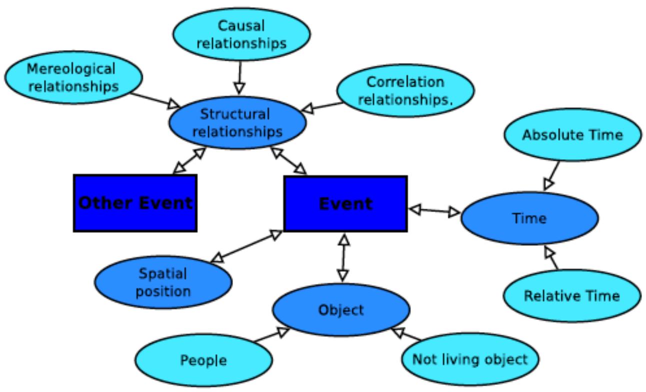

Fig. 7 Hierarchy of events in Event-Model-F

For the realization of relationships in DUL, the following patterns are used; participation, documentation and interpretation patterns [10]. Objects in events are realized with the help of participation pattern. So one or more of those objects are involved in an event. The developers of the Event-Model-F ontology wanted to give users a way to document the events. For this purpose, a documentation pattern is used. Yet another pattern is an interpretation pattern. This pattern is used to interpret events differently. This may be necessary in some cases, e.g., if one wants to look at an event from different viewpoints. Through various points of view, observers are thinking of different event sources

The relationships between events are shown with the help of three other patterns. These are mereology, causality and correlation patterns. It is often the case that several events are part of a big event. A mereology pattern is used just to represent such a relationship. A causality pattern is used to map relationships between the cause and the effect. The ontology Event-Model-F has two major event types. These are `F:Cause` and `F:Effect`. An example of causality pattern: one presses a pedal (`F:Cause`) and the car drives (`F:Effect`). There are two events here: “pedal pressed” and “car drives.” A correlation pattern describes the relationship between statistical characteristics. Correlation occurs only when the events themselves did not contain influences of each other.

Summary: The Event-Model-F ontology is an easily extensible upper ontology. The ontology provides a formal definition of events. Events could have time, space and objects involved. In the ontology, different relations between events are possible. These relationships are depicted with the help of special patterns. The Event-Model-F ontology can easily be extended to a domain-specific ontology.

D.LODE

This section describes the representation of events in the LODE (Linking Open Description of Events) ontology. The linking is achieved, by publishing and connecting RDF data sets on the web using URIs for the identification of web documents, relations between them, real-world objects, and other types of information that is sought in RDF data sets [32]. The LODE ontology summarizes a minimal model to condense the most important properties related to events such as `atPlace`, `atTime`, `circa`, `illustrate`, `inSpace`, `involve`, and `involvedAgent` of a particular event.

What is LODE: "The LODE ontology is a minimal model that encapsulates the most useful properties for describing events" [32]. This ontology was the result of the design of an Interlingua model that resolves an interoperability problem. This is undertaken by providing a set of axioms expressing mappings between the existing event ontologies. The target of the LODE ontology is to permit interoperability when modeling the factual aspects of events. Those aspects are characterized in the four W's, which are: **W**hat happened, **W**here did it happen, **W**hen did it happen, and **W**ho was involved. These "factual" relations within and among events are constructed to generate representations of inter-subjective "consensus of reality" and thus are not necessarily associated with a particular perspective or interpretation [22]. The model then permits to express features about which a stable consensus has been attained [32].

Events in LODE: In the LODE ontology, it is defined the following vocabulary, which is formed by one principal class (`Event`) and seven properties that refer to the happening of the event [3]:

- `Event` is described as a class that is defined as "something that happened". An event comprises of boundaries that are temporal and spatial imposed on the flow of imagination that is treated as an entity for the purposes of making statements about it. An event can be particularly related to make statements correlated to people, places or things. The event definition does not stipulate that some particular event involves state change. Additionally, events are not differentiated from processes or states.
- `atPlace` is described as a property and answers the question: "Where did the event happen?" This property refers an event to one particular or relative place. It might have a name (e.g., "USA") or might be defined as a relative name to other entity or entities (e.g., "new unincorporated regions between Mexico and USA"). An event can be correlated to more than one or more places.
- `atTime` is described as a property and answers the question: "When did the event happen?" by giving of an abstract instant or a time interval. This property refers an event to imposed temporal boundaries (i.e., a time span). Thus, an event can be related to only one time span.
- `circa` describes a temporal relation expressing approximation in time. A time interval that cannot be precisely located in calendar dates and clock times (i.e., chronological series to another time span that can be precisely located in chronological series).
- `illustrate` depicts that an event can be illustrated by something. So this property refers anything to an event, so in such a way that the event then illustrates, documents or comments upon.
- `inSpace` is referred as an abstract region of space and answers to the question: "Where an event happened?" in spatial terms. It refers an event to a region of space (i.e., it only asserts the event occurred somewhere within certain region but does not assert that the event has occurred everywhere within the region.
- `involved` is just referred to any physical, social, or mental object or substance involved in an event.
- `involvedAgent` is referred one event to anything with an agency such as a person, a group, an organization, a computational agent, etc.

Fig. 8 illustrates the metadata attached to the event with ID 1380633 on last.fm according to the LODE ontology. More precisely, it indicates that an event of type `Concert` has been given on January 24, 2010 at 20:00 PM in the Henry Fonda Theater featuring the Radiohead rock band.

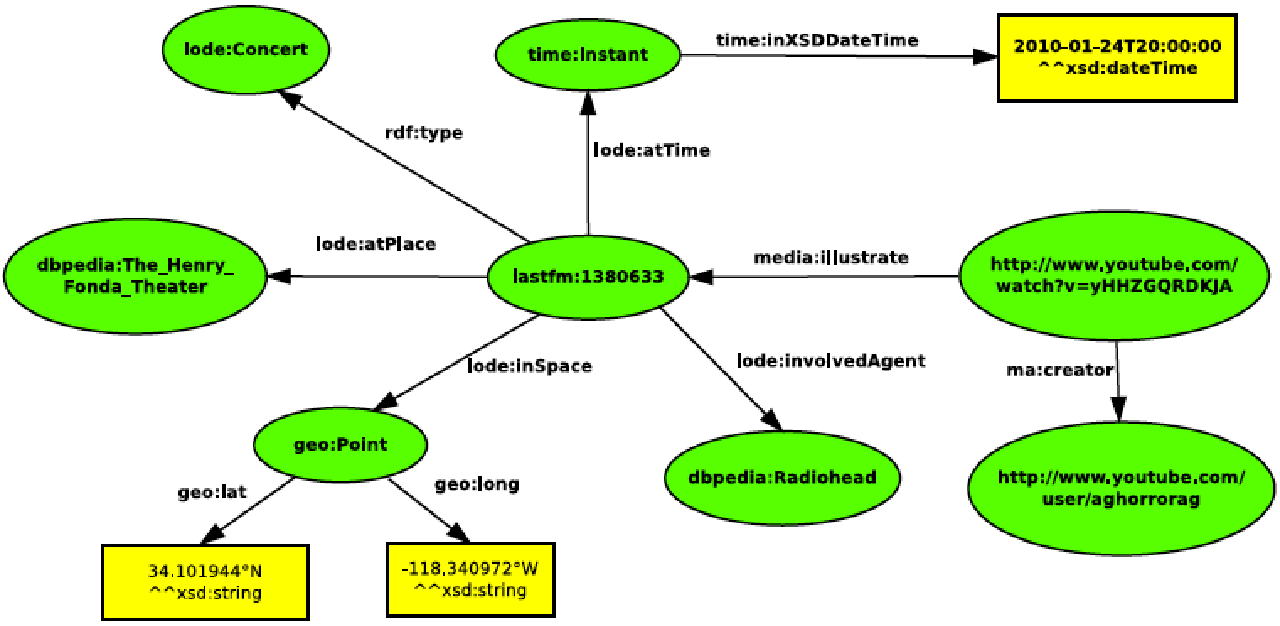

Fig. 8 Radiohead Haiti Relief Concert described with LODE [22]

Summary: The LODE ontology facilitates interoperability when factual aspects of events are modeling. Those aspects can be characterized in the four W's, which are: **W**hat happened, **W**here did it happen, **W**hen did it happen, and **W**ho was involved, all of them are described by the punctual properties defined in the LODE ontology with reference to the occurred event (i.e., LODE's vocabulary). Some representations o consensus of reality can be effectuated by the factual relations (within and among) events and thus those representations can be associated to many interpretations.

E. ABC

This section describes the ABC ontology, which is a metadata model that is a result of the DSTC (Australia), JISC (UK), and NSF (US) funded Harmony Project [2]. This ontology does not pretend to consolidate a metadata vocabulary per se, but intended as a basic model and an ontology that can facilitate the notional basis to develop a domain, role, or community specific ontologies [26].

The ABC ontology firstly had the following goals:

- To provide conductance to communities that are beginning to examine and develop some descriptive ontologies.
- To develop a basis for mappings automated amongst metadata ontologies.
- To supply a basis to understand and analyze existing metadata ontologies and instances.

The ABC ontology then assembles a number of basic entities and relationships done by common across metadata ontologies, including object and time modification, concepts, places, agency, and tangible objects. The main purpose of the ontology has been constructed by recognition of many existing metadata efforts that are often proceed with insufficient attention to underlying modeling principles.

One of the ABC's formal principles has been constantly conducted to attempts to express complex resource descriptions without a clear exposition of entities and relationships that are necessary for such descriptions [12]. The ABC ontology is intentionally designed as a primitive ontology, in that way this permits communities can be competent to build on the top of the mentioning ontology. A set of base classes is provided to perform as either attachment points for domain-specific properties or super classes. Those can be sub-classed to create domain-specific classes.

What is ABC: "It is a basic Ontology, which provides a basic model for domain-related or community-related development" [37]. It is a model that has been deliberately designed for the modeling of physical, digital and analogue objects that are held in libraries, archives, and museums and on the Internet [25]. Those objects can be media types (i.e., multimedia such as image, video, audio) and other types (e.g., text, web pages) [26]. "It can also be used to model abstract concepts such as intellectual content and temporal entities such as performances or lifecycle events that happen to an object. In addition, the ontology can be used to describe other fundamental entities that occur across many domains such as agents (e.g., people, organizations, instruments), places and times." [25]

Because there have been developed a number of metadata models, which identify importance of events describing resources, lifecycles of them and in supplying semantic interoperability between certain domains of metadata, in this paper it is just analyzed those classes of ABC ontology that are involved with events.

Fig. 9 illustrates the top-level class hierarchy of the ABC ontology. An entity is the primitive category at the core of the ontology. Three categories lie at the next level of the ontology: temporality, actuality, and abstraction (apart of time and place).

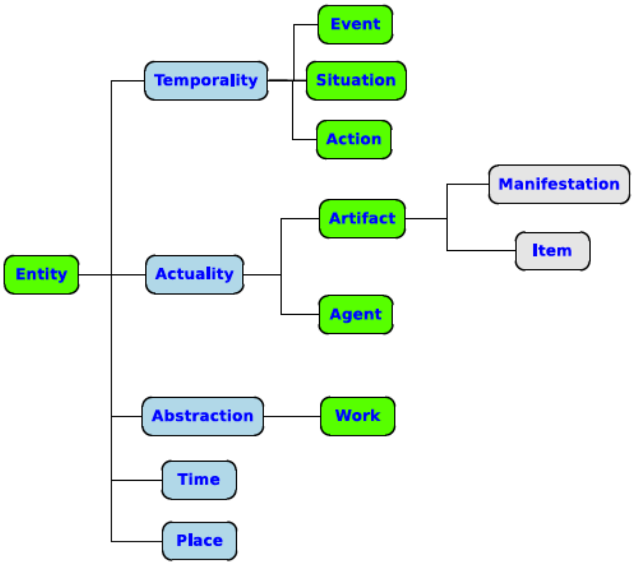

Fig. 9 Hierarchy of events in ABC ontology [25]

Events in ABC: A differentiated aspect of the ABC ontology is the way that it models the time, the way in which properties of objects are transformed over time. The ontology expresses unambiguous situations (i.e., where there are object properties), the demarking transitions of those situations, the actions and agency participated in those transitions.

Time is modeled in the ABC ontology by situation, event and action. A situation provides the context for framing time-dependent properties of (possibly multiple) entities, whose can be a person or a document, may have properties that exist only in the context of a situation and other properties that are constant across the context of a description. An event demarks a transition from one situation to another. Events always have time properties. The situation has time duration but in an implicit way. For instance, the model might express Guernica de Pablo Ruiz Picasso to the Metropolitan Museum for a fixed period (e.g., from June 20, 2011 to July 20, 2011) as follows: an existential facet of Guernica de Pablo Ruiz Picasso located at the Metropolitan Museum could be associated with a situation that is related via precedes and follows properties with two events, which one of them gives the time of Guernica de Pablo Ruiz Picasso, whereas the other does the time of the return [26]. An action supplies mechanisms to model increased the knowledge regarding the involvement and responsibility of agents in events.

An actuality comprehends entities that are sensible (i.e., they can be heard, seen, smelled, or touched). This can be contrasted with an abstraction, which comprehends concepts instead of entities. An abstraction expresses concepts or ideas. Entities have two demarkable characteristics: the entities are never in the context of a situation and the ideas cannot exist in isolation in the model.

The ABC's vocabulary is as follows:

- `Entity` is the primitive category/class.
- `Temporality` is for sub-classing categories of entities, which provide time existential contexts.
- `Actuality` is for sub-classing categories of entities. They have a tangible existence in some world's view.
- `Abstraction` is for sub-classing categories of entities, which contemplate pure information or concepts.
- `Event` marks a transition between situations. That transition must be associated with the event through properties `precedes` and `follows`. The time of a transition is variable (e.g., events are truly in a point time). However, an event may have a coarser granularity such as a span of time during which some situation change was undertaken (e.g., the painting of the Sistine Chapel Ceiling). The granularity of the snapshot is associated with the event via a property `atTime`.
- `Situation` is a context to make time-dependent or existential assertions with respect to actualities. Every situation can act as a context for existential facets of multiple actualities.
- `Action` is an activity developed by some agent in the context of an event.
- `Agent` is an actuality presented during an event or it can be the part of some action. In agents, it may involve persons, instruments, organizations, etc.
- `Time` is an entity represented as either a time span or a point in time, which can be used to confine the temporal extent of temporalities (i.e., situations or events). This entity provides the range constraint for the property `atTime`.
- `Place` is an entity represented in spatial location. It is used to specify the location of either temporalities or actualities. This entity supplies the range constraint for the property `inPlace`.
- `precedes`: This property serves to ligate actualities and situations within its context as existing before an event.
- `follows`: This property serves to ligate actualities and situations within its context as existing after an event.
- `contains`: This property establishes a "contains" relationship between entities.
- `isPartOf`: This property is the inverse of `contains` and makes an establishment of an "is-part-of" relationship between entities.
- `isSubEventOf`: This property establishes an "is-part-of" relationship between events (e.g., the relationship between D-Day and World War).
- `involves`: This property takes part in the action and event domain, and expresses actuality involvement in the event or action performance.
- `hasPatient`: This property forms part of the action and event domain and enforces the notion of `involves` to the classic patient sense stating that the actuality that is the value of this property is transformed by the action or event.
- `usesTool`: This property is a specialization of `involves`.

- `hasResult`: Using this property, it is expressed an actuality result. In that result, there always must be in an existential facet in the performance of the action (i.e., in the context of an event).
- `destroys`: This property is a specialization of `hasPatient`. It points out the value with which the actuality ceases to exist in situations that follow an event.
- `creates`: This property specializes `asResult` to signify the coming into existence of the actuality.
- `hasAction`: This property serves to indicate that an event can have one or more actions. Those actions are verbs performed by agents in the context of an event.
- `hasPresence`: This property makes an association of an agent being present in the context of an event.
- `hasParticipant`: This property refines `hasPresence` to associate an agent as an active participant in an event or action.

Summary: The ABC ontology was specifically developed and designed to model the creation, evolution and transition of objects over time, where the ontology supplies a simple model for domain-related or community related development. The ABC ontology may be extended and may provide one schema to model, validate, storage, navigate, and search for the different types of metadata. In the ABC ontology, it is described that there are event-related concepts such as an event, a situation, an action, an agent, and their relationships. An event signals when situations are transformed. Situations are contexts, which are predicates to aspects of an actuality. An action can be performed by agents (i.e., in the event context). Agents may be people, organizations, instruments, etc. In the ABC ontology, a class Event has two main concepts, which are: `subEvent` and `Action`. A property `isSubEvent` is used to express integral-part relationship between events. A property `atTime` is not considered in relationships between events. The remarked difference between events and actions of an event is that an event and its sub-events mark the transformation points between their starting point and ending states. The verbs of relevant events can be represented by actions (i.e., events and sub-events have temporal and spatial restrictions but actions do not).

IV. Example of Practical Use of Ontology in Complex Event Processing

The investigated ontologies are already being used in several projects. One of these projects is the EU project WeKnowIt [8]. A case study of the project is an emergency system, which uses the Event-Model-F ontology.

Fig. 10 gives an overview of the emergency system. The aim of the emergency system is to recognize disasters in real time. There are many things happening. These events are collected in the central system. Each event can be documented. It invites all parties of listed objects and persons. As an example, consider a woman who calls the fire department hotline and reports an overflowing basement. The fire department drives to the emergency position and announces a new event. The power failure is reported and police drives to that place and registers the event. All the events are registered to a central location. This enables to analyze the overall situation and permits quickly and timely to recognize the disaster and the affected place.

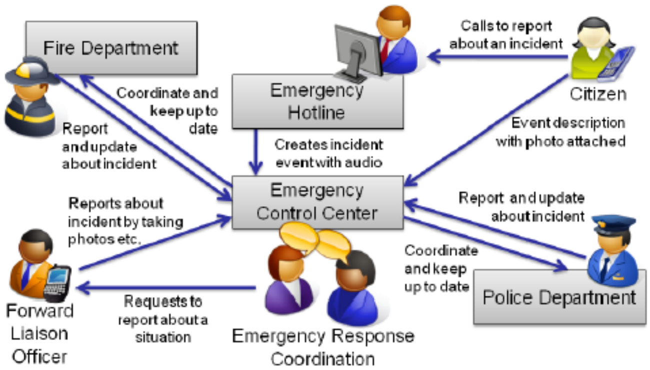

Fig. 10 Emergency system from WeKnowIt [8]

Due to the different patterns, the Event-Model-F enables to recognize the connections between the events. In the ontology, it is also possible to document events. At individual events, there can be more involved entities (e.g., people or other objects).

V. Conclusion

The fact is that there have been developed many ontologies for representing events, where an event is a concept individually treated in the description of the underlying ontology because the event can be composed of some temporal and spatial boundaries.

In this paper it was investigated five ontologies, which involve event-related concepts. The ontologies include many similar concepts. This section provides a comparison of the ontologies against different properties such as time, space, participation, causality, etc.

There are possibilities in the investigated ontologies to modeling events with stretch in time. There are two approaches to describing the time by events: as a description or as an individual object. All of the investigated ontologies take the second approach. Another distinguishing feature is a representation of point in time. The ontologies can represent both absolute and relative time. Yet another important factor in event modeling is that spatial regions and places can be involved in events. The Event and ABC ontologies support relations only to spatial regions. Whereby, the LODE, Event-Model-F and OpenCyc ontologies support relations to the place as well. To distinguish between the place and the spatial region one needs for spatial region a geospatial coordinate systems whereas for place does not [32]. The properties for involving living and non-living objects in events are important factors as well. All of the investigated ontologies use their own concepts to involve objects in events.

To model an event as part of another event, the investigated ontologies offer different possibilities: the OpenCyc ontology has for this purpose a spatial type called `$#SubProcessSlot`, using which one can model complex events that consist of simple ones. The Event ontology offers a property sub-event for modeling sub-events. The LODE ontology does not consider sub-events at all. The ABC ontology has a property `isSubEventOf` for modeling sub-events. For modeling more complex relationships between events, the Event-Model-F ontology offers the largest possibility for it. For this purpose, the ontology offers mereology, causality and correlation patterns.

To sum up, the investigated ontologies offer different possibilities to model events. OpenCyc is a large ontology that contains not only events but also other concepts. This ontology can be used in different domains. The Event ontology was specifically developed for music-related ontologies, but in this ontology there is nothing specific to music so it can be used for modeling events in the CEP domain. The Event-Model-F ontology is developed for the processing of events only; there is no specific domain on it. Therefore, the ontology offers many ways to model more complex events than it is possible with the other ontologies. LODE is an ontology, which facilitates interoperability when modeling the factual aspects of events. That ontology extracts some concepts from the other ontologies to make a vocabulary that describes four the factual aspects of events characterized by: what happened, where did it happen, when did it happen and who was involved? The ABC Ontology is designed to model the process of the object's life cycle over time, where it is supplied a simple model for the domain-related development. In the ABC ontology, one can note that the events signal when situations are transformed.

ACKNOWLEDGMENT

Irina Astrova's work was supported by the Estonian Centre of Excellence in Computer Science (EXCS) funded mainly by the European Regional Development Fund (ERDF). Irina Astrova's work was also supported by the Estonian Ministry of Education and Research target-financed research theme no. 0140007s12.

REFERENCES

[1] Cycorp, Inc. http://cyc.com. [Online; 15.04.2011].
[2] Harmony project. http://metadata.net/harmony/. [Online; 15.04.2011].
[3] LODE ontology. http://linkedevents.org/ontology/. [Online; 15.04.2011].
[4] OpenCyc.org. http://opencyc.org/. [Online; 15.04.2011].
[5] DOLCE Laboratory for Applied Ontology. http://dolce.semanticweb.org/, May 2011.
[6] Protege Homepage. http://protege.stanford.edu/, June 2011.
[7] University of Koblenz-Landau. http://www.uni-koblenzlandau.de/koblenz/fb4/AGStaab/Research/ontologies/events, May 2011.
[8] Weknowit Projekt Homepage. http://www.weknowit.eu/, June 2011.
[9] 5th International Conference. Lecture Notes in Computer Science / Theoretical Computer Sci. Hualien, Taiwan, May 2010.
[10] S. Ansgar, F. Thomas, S. Carsten, and S. Steffen. FA Model of Events based on the Foundational Ontology DOLCE+DnS Ultralite. September 2009.
[11] D. Brickley. W3C Semantic Web Interest Group: Basic Geo (WGS84 lat/long) Vocabulary. http://www.w3.org/2003/01/geo/. [Online; 24.06.2011].
[12] D. Brickley, J. Hunter, and C. Lagoze. Abc: A logical model for metadata interoperability, harmony project. 1999.
[13] K. Chandly and W. Schulte. Event Processing: Designing IT Systems for Agile Companies. 2009.
[14] M. Croitoru, S. Ferr´e, and D. Lukose(Eds.). Conceptual Structures: From Information to Intelligence. Springer, Berlin Heidelberg New York, 2010.
[15] Cycorp,Inc. Actor Slot. http://www.cyc.com/doc/tut/DnLoad/TimeAndDates.pdf. [Online; 15.04.2011].
[16] Cycorp,Inc. Actor Slot. http://www.cyc.com/doc/tut/DnLoad/ActorSlots.pdf. [Online; 15.04.2011].
[17] Cycorp,Inc. Actor Slot. http://www.cyc.com/doc/tut/DnLoad/SubEvents.pdf. [Online; 15.04.2011].
[18] Cycorp,Inc. Actor Vocabulary. http://www.cyc.com/cycdoc/vocab/actor-vocab.html. [Online; 15.04.2011].
[19] Cycorp,Inc. Event Types. http://www.cyc.com/doc/tut/DnLoad/EventTypes.pdf. [Online; 15.04.2011].
[20] Cycorp,Inc. Events in Cyc. http://www.cyc.com/doc/tut/DnLoad/EventsInCyc.pdf. [Online; 15.04.2011].
[21] Cycorp,Inc. Roles and Event Predicates. http://www.cyc.com/doc/tut/DnLoad/RolesAndEventPredicates.pdf. [Online; 15.04.2011].
[22] A. Fialho, R. Troncy, L. Hardman, C. Saatho, and A. Scherp. What's on this evening? designing user support for event-based annotation and exploration of media. Athens, Greece, May 2010.
[23] T. R. Gruber. A translation approach to portable ontology specification. Knowledge Acquisition, pages 199–220, April 1993.
[24] J. R. Hobbs and F. Pan. Time OWL Ontology. http://www.w3.org/TR/owl-time/. [Online; 24.06.2011].
[25] J. Hunter. Enhancing the semantic interoperability of multimedia through a core ontology. IEEE Transactions on Circuits and Systems for Video Technology, 13:49–58, 2003.
[26] C. Lagoze and J. Hunter. The abc ontology and model. pages 160–176, 2001.
[27] C. Maa and G. Gernot. Suchmaschinen und Informationsqualit ¨at: Status quo, Problemfelder, Entwicklungstendenzen. In K. Hildebrand, M. Mielke, M. Gebauer, and H. Hinrichts, editors, Daten- und Informationsqualitt: Auf dem Weg zur Information Excellence, pages 157–171. Vieweg+Teubner Verlag, 2008.
[28] R. Poli, M. Healy, and A. Kameas. Theory and Applications of Ontology: Computer Applications. Springer, Dordrecht Heidelberg London New York, 2010.
[29] Y. Raimond and S. Abdallah. Event Ontology. http://motools.sf.net/event/event.122.html. [Online; 24.06.2011].
[30] Y. Raimond and S. Abdallah. Timeline Ontology. http://motools.sourceforge.net/timeline/timeline.html. [Online; 24.06.2011].
[31] H. Sack. Ontologien. January 2006.
[32] R. Shaw, R. Troncy, and L. Hardman. Lode: Linking open descriptions of events. In A. G´omez-P´erez, Y. Yu, and Y. Ding, editors, The Semantic Web, volume 5926 of Lecture Notes in Computer Science, pages 153–167. Springer Berlin / Heidelberg, 2009.
[33] A. Tacke. Taxonomien & Ontologien. 2008.
[34] K. Teymourian. Semantic Complex Event Processing. September 2010.
[35] Vincent, P. Paul, B. Adrian, and Harold. Semantic Complex Event Processing. July 2010.
[36] W. Wang, W. Guo, X. W. Yingwei Luo, and Z. Xu. Ontological model of event for integration of inter-organization applications. pages 301–310, 2005.
[37] W.Wang,W. Guo, X.W. Yingwei Luo, and Z. Xu. The study and application of crime emergency ontology event model. pages 807–808, 2005.
[38] J. Zhuk. Integration-Ready Architecture and Design: Software Engineering with XML, Java, .NET, Wireless, Speech, and Knowledge Technologies. University Press, Cambridge, 2004.