Applied Ontology 3 (2009) 1–3
IOS Press 

# Designing Core Ontologies

Ansgar Scherp [a], Carsten Saathoff [a], Thomas Franz [a], Steffen Staab [a]
[a] *WeST, University of Koblenz-Landau, Germany*
*E-mail: {scherp|saathoff|franz|staab}@uni-koblenz.de*

**Abstract.** One of the key factors that hinders integration of distributed, heterogeneous information systems is the lack of a formal basis for modeling the complex, structured knowledge that is to be exchanged. To alleviate this situation, we present an approach based on core ontologies. Core ontologies are characterized by a high degree of axiomatization and formal precision. This is achieved by basing on a foundational ontology. In addition, core ontologies should follow a pattern-oriented design approach. By this, they are modular and extensible. Core ontologies allow for reusing the structured knowledge they define as well as integrating existing domain knowledge. The structured knowledge of the core ontologies is clearly separated from the domain-specific knowledge. Such core ontologies allow for both formally conceptualize their particular fields and to be flexibly combined to cover the needs of concrete, complex application domains. Over the last years, we have developed three independent core ontologies for events and objects, multimedia annotations, and personal information management. In this paper, we present the simultaneous use and integration of our core ontologies at the example of a complex, distributed socio-technical system of emergency response. We describe our design approach for core ontologies and discuss the lessons learned in designing them. Finally, we elaborate on the beauty aspects of our core ontologies.

Keywords: Core Ontologies, Ontology Design Patterns, Ontology Interplay, Ontology Design Approach

## 1. Introduction

Domains that require the exchange of a high amount of complex, structured knowledge such as medical systems, distributed media management, and emergency response have a high pressure for systems integration in order to facilitate efficient communication and information exchange. For example, in emergency response different entities such as an emergency hotline, police department, fire department, and emergency control center are involved. These entities need to exchange among others knowledge about what happened during an incident, tasks communicated between the entities, and media information that documents the incident. Due to the lack of appropriately integrated systems at the different emergency response entities, the complex, structured knowledge is currently exchanged via natural language on the phone. This is very error-prone and inefficient. Rather, the different, heterogeneous systems used by the emergency response entities should be integrated to provide a more efficient and effective exchange of the knowledge. One of the key factors that hinders integration of these systems is the lack of a formal basis for modeling the complex, structured knowledge that is to be exchanged. So far, this problem has not been solved due to the lack of networked ontologies that provide a flexible means to model the complex structure of the knowledge exchanged and at the same time provide a formal semantics to that structure.

In this paper, we propose an approach based on core ontologies to alleviate this situation. An ontology allows for formally representing the relevant concepts and relations of a considered domain in a machine readable format (Oberle et al., 2009b; Oberle, 2006). Core ontologies provide a precise definition of structural knowledge in a specific field that spans across different application domains, e.g., software services, personal information management, knowledge organization, multimedia annotations, and others (Oberle, 2006). They combine a number of specific properties that have been derived from reported experiences in designing core ontologies (Oberle et al., 2007, 2006; Oberle, 2006) and the development of our own core ontologies (Arndt et al., 2009; Scherp et al., 2009a; Staab et al., 2008; Arndt et al., 2007; Franz et al., 2007). These properties are axiomatization and formal precision, modularity, extensibility, reuseability, and separation of concerns.

**Axiomatization and Formal Precision.** A high degree of axiomatization and formal precision is provided by core ontologies. By this, a common understanding in a particular field is established in order to ensure interoperability through machine accessible semantics. Systems can reason about the represented knowledge and carry out semantic checks on its validity. The axiomatization and formal precision is achieved by basing on a foundational ontology.

**Modularity.** Core ontologies should follow a pattern-oriented design approach. By this, they are modular within the field for which they are designed. Ontology design patterns[1] provide a modeling solution to a recurrent ontology design problem (Gangemi and Presutti, 2009; Gangemi, 2007, 2005). Thus, as it is also argued by (d'Aquin and Gangemi, 2011), ontology design patterns are similar to design patterns in software engineering (Gamma, 2007). The core ontology is a composition of such ontology design patterns with appropriate dependencies between the patterns (Gangemi and Presutti, 2009). This enables a pathway for extensibility and reuseability.

**Extensibility.** Being modular, a core ontology allows for adding new and updating or removing modules, i.e., ontology design patterns it defines. By this, the core ontology is able to reflect system evolution (cf. adaptability in (Vrandečić, 2009)). It is extensible towards new developments and functional requirements that arise.

**Reuseability.** Different systems are built for different purposes and users in different domains. Being modular, a core ontology supports reuse of its modules, i.e., the ontology design patterns despite of the different foci and domains the concrete systems have. At the same time a core ontology still guarantees formal precision of the overall knowledge it represents. In addition to the reuse of the domain-independent, structured knowledge defined by the core ontologies themselves, also existing domain knowledge can be reused. Core ontologies are able to incorporate existing domain ontologies and make use of that domain knowledge rather than requiring to remodel it.

**Separation of Concerns.** The structural knowledge defined in a core ontology is clearly separated from the domain-specific knowledge. This allows core ontologies to be applied in arbitrary application domains. Domain-specific knowledge such as a domain ontology on emergency response or sports can be integrated and reused without affecting the core ontology itself.

In this paper, we will show that combing these properties in a core ontology can lead to elegant solutions and interoperability in complex application domains. Due to the characteristics of their design such core ontologies can be flexibly combined to cover the needs of concrete, complex application domains. Thus, from our perspective they are to be considered *beautiful* ontologies.

Over the last years, we have developed three of those beautiful core ontologies. These core ontologies have been used over a long time, are very stable with respect to their design, and thus provide a sustainable solution for ensuring interoperability in complex socio-technical systems such as emergency response. These core ontologies are the Event-Model-F, COMM, and X-COSIMO. The core ontology Event-Model-F is designed for modeling events and objects (Scherp et al., 2010, 2009a). It allows for representing human experience and participation in real world occurrences and provides comprehensive support for modeling time and space, objects and persons, as well as mereological, causal, and correlative event relationships and event interpretations. The Core Ontology on Multimedia (COMM) (Arndt et al., 2009; Staab et al., 2008; Arndt et al., 2007) is designed for describing arbitrary digital media data. It allows for (semantic) annotations of media data and their decompositions. Finally, the Cross-Context Semantic Information Management Ontology (X-COSIMO) is designed for semantic information management and communication (Franz et al., 2007). It supports modeling the communication taking place between different persons and systems and the information associated with it such as task descriptions.

Our three core ontologies, i.e., the Event-Model-F, COMM, and X-COSIMO are based on the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) (Gangemi et al., 2002; Masolo et al., 2003). As foundational ontology, DOLCE aims at modeling the very basic and general concepts and relations (Borgo and Masolo, 2009; Oberle, 2006) that make up our world, e.g., objects, events, participation, and parthood. Foundational ontologies are generic across many fields (Oberle, 2006). They have a large scope and are highly reusable in different modeling scenarios (Borgo and Masolo, 2009). By their nature,

[1] `http://ontologydesignpatterns.org/`

foundational ontologies are much broader than core ontologies such as our Event-Model-F, COMM, and X-COSIMO. Core ontologies provide a refinement to foundational ontologies by adding detailed concepts and relations in their specific field. DOLCE already has proved to be a good modeling basis for core ontologies such as (Scherp et al., 2009a; Arndt et al., 2007; Franz et al., 2007; Oberle et al., 2006, 2007).

The Event-Model-F, COMM, and X-COSIMO have been carefully aligned with the foundational ontology DOLCE+DnS Ultralight[2] (DUL), a lightweight version of DOLCE. By this alignment, our core ontologies can be flexibly combined to cover the needs of complex application domains. Our core ontologies follow a pattern-oriented ontology design approach, i.e., they define a set of ontology design patterns targeted for the specific field they model. These patterns are based on the very generic patterns DUL provides such as the Descriptions and Situations (DnS) pattern and the Information Object (IO) pattern (Borgo and Masolo, 2009). By using a foundational ontology and following a pattern-oriented design approach, the core ontologies possess a solid, semantically precise basis. At the same time these core ontologies become modular and extensible with respect to their use in concrete applications and to changes in functional requirements. By applying the DnS pattern, our core ontologies allow for a clear separation of the structured knowledge captured by the core ontology and the domain knowledge provided by a domain ontology. Thus, they allow for integrating and reusing existing domain ontologies.

The remainder of the paper is organized as follows: In the next section, we motivate the need for core ontologies to model complex, structured knowledge by presenting a scenario of a complex, socio-technical system in the domain of emergency response. In Section 3, we demonstrate the simultaneous use and smooth interplay of our three core ontologies Event-Model-F, COMM, and X-COSIMO in the emergency response scenario. It demonstrates the use of the three core ontologies to model the complex, structured knowledge that needs to be exchanged between the different systems involved. The properties of core ontologies and our design approach for developing such core ontologies are presented in detail in Section 4. The concrete design of our three core ontologies the Event-Model-F, COMM, and X-COSIMO is presented in Section 5. In Section 6, we discuss the lessons learned when designing and using our core ontologies. We argue for the beauty of our core ontologies in Section 7, which lies in their ability to both formally conceptualize their particular fields and to be flexibly combined to cover the needs of concrete scenarios, before we conclude the paper.

## 2. Modeling and Sharing Complex, Structured Knowledge in Emergency Response

In the emergency response scenario of the EU project WeKnowIt[3] depicted in Figure 1 different emergency response entities are involved using different, heterogeneous systems. These systems need to exchange complex, structured knowledge that needs to be shared among the emergency response entities. Examples of emergency response entities are the emergency hotline, police department, fire department, emergency control center, and forward liaison officers. The emergency control center is in charge of coordinating the emergency response entities. It receives event descriptions from the emergency hotline, processes them, and communicates event descriptions with the police department and fire department. In addition, the emergency control center forwards event descriptions together with task descriptions to their forward liaison officers. Forward liaison officers are out in the field to report about a situation, verifying it, and documenting it, e.g., by taking photos and notes. As the following scenario shows, this socio-technical system for emergency response becomes very active in the case of an incident. Many different pieces of structured knowledge such as events, tasks, and multimedia data with metadata are created, combined, and communicated between the different emergency response entities involved. Subsequently, we discuss how our core ontologies are involved in modeling this complex, structured knowledge.

[2] http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS_Ultralite

[3] http://www.weknowit.eu/

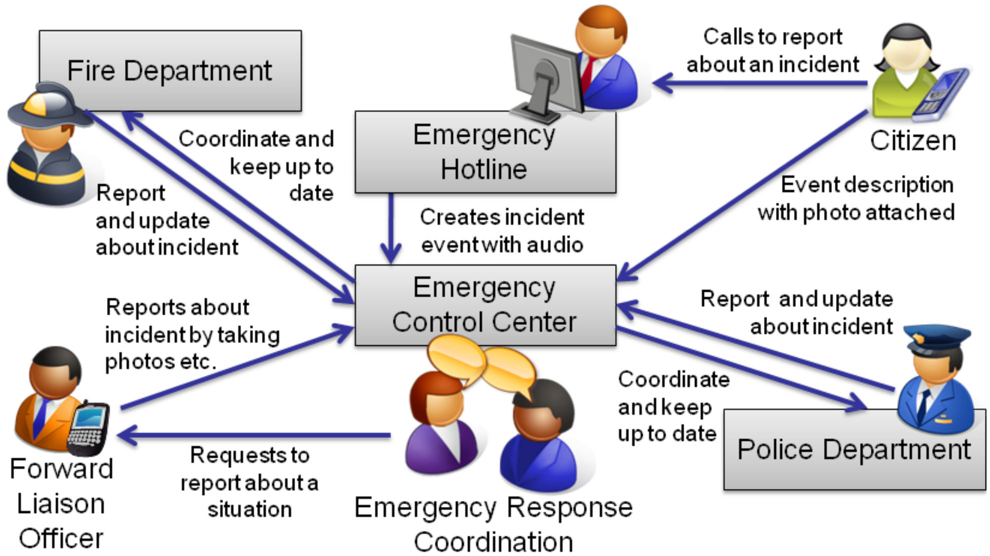


Fig. 1. A distributed system for emergency response

## 2.1. *Scenario*

In an incident of a heavy storm a major flooding may happen. During the flooding a power outage occurs. Some citizens are lacking power supply and are calling the emergency hotline to report about the power outage event. The officers at the emergency hotline record these calls. In addition, they type in a textual description for each call to document the reported event in their system. The recorded calls are automatically processed by some speech recognition techniques, which creates a transcript of the calls. The algorithm also automatically distinguishes the different voices of the participants in the phone call and can automatically associate and differentiate utterances made by the citizen and the officer at the emergency hotline. Subsequently, the event description together with the processed call recording and its transcript are automatically transferred to the system of the emergency control center. The emergency control center also receives event descriptions from the systems of the police department and fire department that happen during the incident. Based on the evidence of the event descriptions, the officers in the emergency control center use their system to formulate hypothetical events that might have caused the power outage. To this end, event descriptions are analyzed, (semi-)automatically clustered, visualized, and put into relation. The officers conclude that there are two possible interpretations that might have caused the power outage, namely a snapped power pole close to the river that was reported by a citizen calling the emergency hotline or a serious problem with the power plant nearby. The correct assessment of the situation is very crucial in order to most effectively deploy the available emergency response resources. Thus, the different event interpretations modeled in the system need to be verified by the officers as soon as possible in order to confirm or reject the hypothesis. For this purpose, the officers in the emergency control center may contact the personnel of the power plant. At the same time, the description of the hypothetical event of a snapped power pole together with a task description is sent to the mobile device of a forward liaison officer. The forward liaison officer receives the task description. She drives to the location of the events to verify it and to document it by taking photos and notes. The photos can also be tagged or otherwise annotated. The event description extended by annotated photos are send back to the system of the emergency control center.

As the scenario shows, the different entities involved in an emergency response have to share structured knowledge among each other through the different systems they use. The structured knowledge is a combination of event descriptions that happen during an incident, e.g., provided by the citizens calling the emergency hotline or annotated by forward liaison officers when verifying a situation. It also comprises task descriptions that are communicated and shared, e.g., the forward liaison officers receive task descriptions (together with the event description) to clarify a specific situation at a certain place and to document about it by taking pictures and notes. Finally, these pictures and notes taken and the metadata that is attached to the media is another kind of structured knowledge that is communicated within the distributed socio-technical system for emergency response shown in Figure 1. As the discussion shows, the structured knowledge that is to be communicated is quite complex. In addition, the entities involved with this structured knowledge, i.e., the events, media data, and tasks appear and are relevant in different contextual settings that need to be modeled. For example, an event reported by a citizen to the emergency hotline and represented appropriately may become an attachment of a message with a task description that is sent by the emergency control center to one of their forward liaison officers.

### 2.2. *Involved Ontologies*

To model the complex, structured knowledge in the emergency response scenario, i.e., the events, media data, and tasks we use and combine our three core ontologies Event-Model-F, COMM, and X-COSIMO. For representing events and the multiple relationships between them, we use the Event-Model-F (Scherp et al., 2009a) that has been developed in the WeKnowIt project. The Event-Model-F provides a formal representation of the different aspects of events in which humans participate such as time and space, composition, correlation, and documentation. Compared to existing models, the Event-Model-F differs in providing sophisticated support for modeling causality, correlation, and interpretation of events.

The Core Ontology on Multimedia (COMM) (Arndt et al., 2009; Staab et al., 2008; Arndt et al., 2007) allows to represent arbitrary digital media data such as images, videos, and audio. It supports the different kinds of annotations of media data and their decomposition into segments. COMM is highly influenced by and specifically designed to support the low-level descriptors of MPEG-7 (MPEG-7, 2001). Its roots go back to the EU project aceMedia[4], where a first attempt to model a MPEG-7 ontology has been undertaken. This ontology is not based on a direct translation of MPEG-7 but on an analysis of the MPEG-7 standard. It follows a formal approach for modeling the multimedia annotation domain based on DOLCE (Bloehdorn et al., 2005). In the EU Network of Excellence K-Space[5], this idea of analyzing MPEG-7 in order to model a formal core ontology for multimedia has been taken up again. In contrast to the aceMedia approach, the COMM developed in K-Space is further axiomatized and based on ontology design patterns in order to acquire an easier to use and formally more sound model.

Finally, the Cross-Context Semantic Information Management Ontology (X-COSIMO) supports for modeling semantic information management and communication (Franz et al., 2007). X-COSIMO allows to represent the communication taking place between different persons and systems and the information associated with this communication like a task description. The core ontology has been developed in the X-Media project[6], which is dedicated to research on large scale and cross-media knowledge management solutions. In the X-Media project also COMM has been used and extended. Both core ontologies COMM and X-COSIMO play a key role in the shared representation of automatically extracted and newly created information in the X-Media project. Among others, the shared representation is exploited in user interfaces that enable users to deal with the diversity of the knowledge represented.

### 2.3. *Summary*

We have shown the requirement of sharing complex, structured knowledge at the example of a socio-technical system for emergency response. The structured knowledge to be modeled and exchanged within this scenario are the representation of events and objects, multimedia data and its annotations, and personal information management such as communication and tasks. In the next section, we present how such structured knowledge can be modeled with the core ontologies Event-Model-F, COMM, and X-COSIMO we have developed. We demonstrate the use of our core ontologies at the example of the emergency response scenario of the WeKnowIt project.

---

[4] `http://www.acemedia.org/`
[5] `http://kspace.qmul.net/`
[6] `http://www.x-media-project.org/`

## 3. Modeling the Emergency Response Scenario

Referring to the scenario of the distributed socio-technical system for emergency response presented in Section 2, we exemplify in this section how the structured knowledge that is exchanged in this system can be modeled. We pick out a small part of the scenario and fully represent it by applying the core ontologies we have developed, namely the Event-Model-F for modeling events and objects (Scherp et al., 2009a), COMM for representing multimedia annotations (Arndt et al., 2007), and X-COSIMO for personal information management and communication (Franz et al., 2007). With modeling the scenario, we show the interplay of these core ontologies, before we discuss the properties of core ontologies in Section 4 and their concrete design in Section 5.

In the following, we consider a power outage that has happened in the course of a major flooding. Many citizens are calling the emergency response hotline such as Paul. He calls the emergency hotline to report about an observation he made, a power pole in his street has just snapped. Shortly after, the power outage happens. Thus, Paul reports to the hotline that he thinks that the snapped power pole has caused the power outage. The officer Rita at the emergency hotline answers Paul's call. She types into her system what Paul reports, while also an automatic recording of the conversation is taken. In our ontologies, the citizen Paul is represented by the individual `paul-1` and the power pole is represented by the individual `power-pole-1`. We model the event when the power pole snapped as the individual `snapped-pp-1`, the event of the power outage as `power-outage-1`, and the event in which Paul calls the hotline as `call-1`. The officer Rita working at the emergency hotline is represented by the individual `rita-1`. She is answering the `call-1`. The overall flooding event is referenced as `flooding-1`.

Using our Event-Model-F, we model the participation of the person `paul-1` in the event of a `snapped-pp-1` as shown in Figure 2 by applying the core ontology's participation pattern. The pattern is based on the generic ontology design pattern Descriptions and Situations (DnS) (Gangemi, 2008; Gangemi and Mika, 2003). The ontology design pattern DnS provides an ontological formalization of context (Gangemi and Mika, 2003; Oberle, 2006). It allows for a formally precise representation of different, contextualized views by defining roles (Welty et al., 2006). Roles classify entities in a specific contextual situation. The role an entity plays may be different in other situations. Thus, roles are typically a non-rigid property (Guarino and Welty, 2002b, 2000a,b). Besides representing the participation of a person in an event, the participation pattern shown in Figure 2 also defines that Paul plays the role of a citizen in this participation, indicated with `paul-citizen-1` which is of concept `CitizenRole`. In another situation, Paul might have a different role, e.g., if he is besides being a citizen also a professional firefighter. Thus, Paul can play the role of a `FiremanRole` in other events. It is important to note that both the `CitizenRole` and the `FiremanRole` are not defined within the participation pattern of the Event-Model-F. However, they are provided from some external, domain-specific ontologies. Thus, the participation pattern and the Event-Model-F in general allows to reuse existing domain knowledge. The use of the DnS pattern in an ontology such as the Event-Model-F can be easily recognized. It always defines a situation that satisfies a description. The situation includes the events and objects of a concrete contextual situation, i.e., the real-world entities that can be observed in a concrete situation. The description defines the roles of these events and objects in the observed situation.

In the concrete example, the situation `part-sit-snapped-pp-1` is an `EventParticipation-Situation` that `satisfies` the description `part-desc-snapped-pp-1`, which is an `Event-ParticipationDescription`. The individual `desc-ev-snapped-pp-1` classifies the real-world event of the snapped power pole `snapped-pp-1`, which is of interest, i.e., described in the considered situation. In addition, we can model the time of the event and location of Paul when participating in the event. This is not shown in the figure, but available online as OWL ontology from our ontologies website: `http://west.uni-koblenz.de/Research/ontologies/`.

The phone call between Paul and Rita is recorded to document the event. The information that the call has been recorded is represented by the individual `audio-rec-1`. This documentary evidence that the phone call event `call-1` actually happened can be modeled using the documentation pattern of the Event-Model-F as shown in Figure 3. It represents the documentation of the event `call-1` by a recording

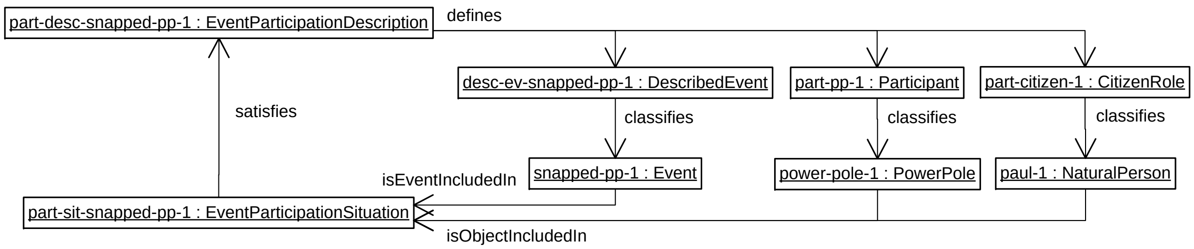


Fig. 2. Citizen Paul participating in the event of a snapped power pole

of the call `audio-rec-1`. The individual `audio-rec-1` is of type `AudioData`, which is a concept taken from the COMM core ontology. The `AudioData` represents the information that is realized, i.e., contained in the audio recording. However, it is not a representation of the audio artifact such as a digital media stream that actually has been captured during the call.

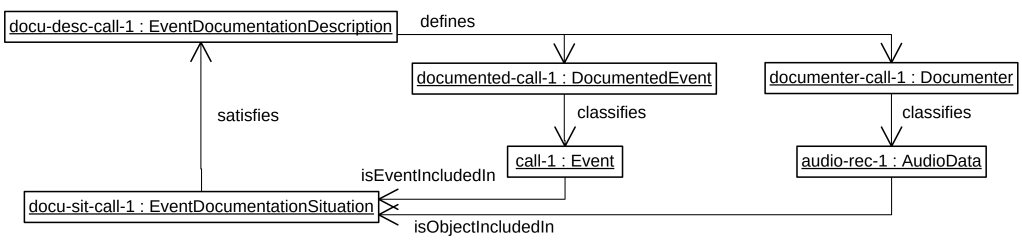


Fig. 3. Documentation of Paul's call to the emergency hotline with an audio recording

In addition to the audio recording, also a description of the `snapped-pp-1` event is typed into the computer by the operators at the emergency hotline. This event description captures detailed information about the incident. For example, in the case of a flooded cellar the event description may contain detailed information for the emergency response entities like the fire department and provide specific instructions about, e.g., how to best reach the cellar, size, water level, and others. In our example, the event description is captured by the individual `text-description-1`, which is of concept `TextData`. It represents a textual description of where the snapped power pole is located and how it can be best reached. This textual description documents the `snapped-pp-1` event as depicted in Figure 4 using the Event-Model-F's documentation pattern. The `TextData` is a specialization of the concept `DigitalData` that is provided by COMM. It is a specialization of COMM towards representing textual data and is defined in the Ontology for Knowledge Acquisition (OAK) (Iria, 2009). The specialization of COMM by OAK is discussed in detail in Section 6.3.

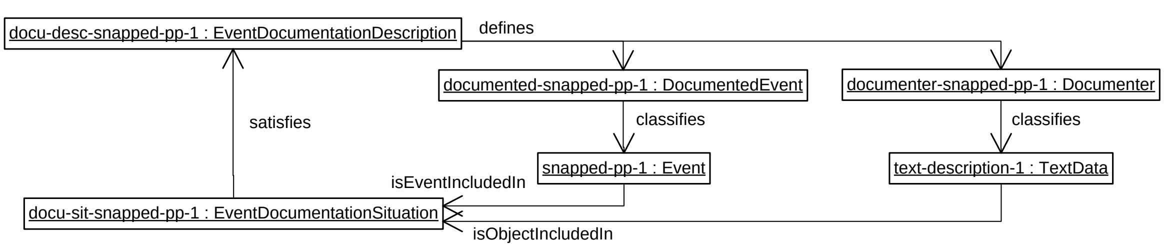


Fig. 4. Documentation of the snapped power pole event with textual instructions for the emergency entities

For the course of this modeling example, it does not make a difference whether one considers `AudioData` or `TextData` to understand the role of media data for documenting events in emergency response. In the following we concentrate on the audio recording, i.e., the `audio-rec-1` indicating that there was a recording taken during the call. Besides its mere documentation purpose, the recording of the phone call `audio-rec-1` between Paul and Rita can be replayed by the emergency response personnel in order to listen again to the conversation. In order to provide a more efficient access to the information communicated in the phone call, the audio recording `audio-rec-1` is processed by automatic classification algorithms such as a segmentation into smaller, distinct parts in which either Paul or Rita are speaking. Each part is automatically annotated with the speaker's name using automatic classification methods. Both the segmentation and the annotation with the speaker's name are modeled using the core ontology COMM. In order to conduct such a sophisticated annotation of an audio recording, we first need to provide a basic representation of the audio recording and the digital artifact created for it as shown in Figure 5.

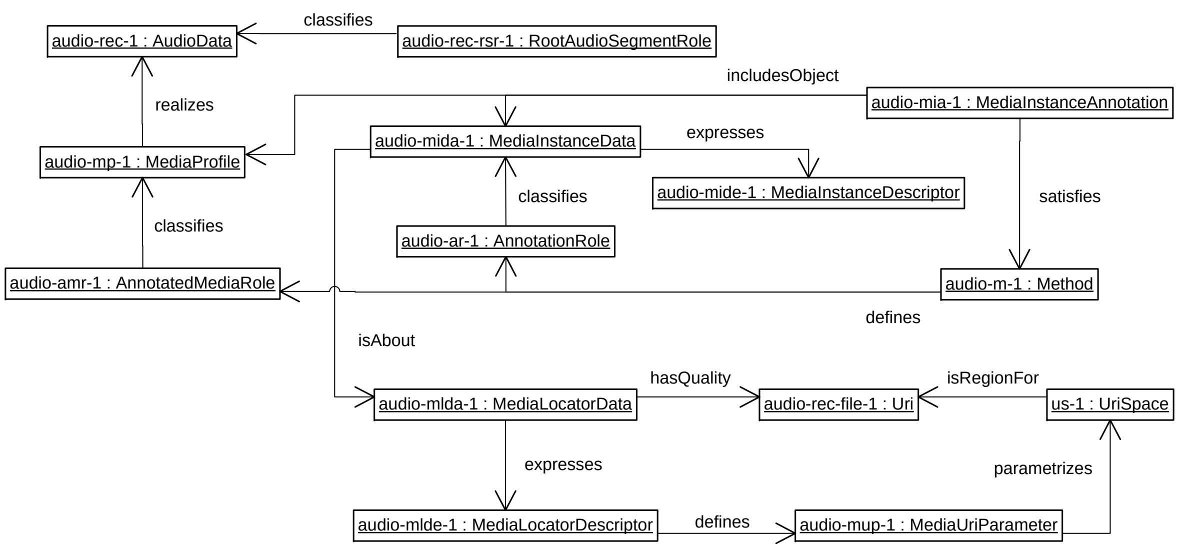


Fig. 5. Representation of an audio recording and the file that realizes the recording

An underlying paradigm of COMM is the distinction of the information represented by any kind of multimedia asset and its digital (or even non-digital) realization. The recording is modeled in COMM as `AudioData`, a sub concept of `InformationObject` and represents the information that expresses some state of affairs (in DUL modeled as a `Description`). This information is clearly separated from the actual realization of that recording. The same information might be stored as `.wav` or `.mp3` file, or even be stored on tape. This separation of the information object and its digital realization is based on the generic ontology of Information Objects (IO) (Borgo and Masolo, 2009). The modeled information, i.e., the fact that the call has been recorded and its digital realization is brought together by the individual `audio-mp-1` of the concept `MediaProfile`. The concept of a media profile originates from the multimedia annotation standard MPEG-7 (MPEG-7, 2001). It allows for modeling various metadata such as color histograms, file type, and file location. This metadata can be complex including nesting of metadata. In COMM, we represent this nested metadata structure with `StructuredDataDescriptions` (see digital data pattern in Section 5.2.1).

In the example of Figure 5, we see two of such `StructuredDataDescriptions`, namely `audio-mide-1` (a `MediaInstanceDescriptor`) and `audio-mlde-1` (a `MediaLocator-Descriptor`). As shown in the figure, the individual `audio-mida-1` of the type `MediaInstance-Data` expresses the `MediaInstanceDescriptor` and is about the individual `audio-mlda-1` of type `MediaLocatorData`. The individual `audio-mlda-1` expresses a `MediaLocator-Descriptor`, represented by the individual `audio-mlde-1`. This `about` relation between `audio-mida-1` and `audio-mlda-1` models the nesting of the `MediaInstanceDescriptor`, which might, among others, contain a `MediaLocatorDescriptor` as subelement.

The location of the digital realization in form of some audio file is represented as an `Uri` named `audio-rec-file-1`. This `Uri` serves as quality of the `MediaLocatorData` represented by `audio-mlda-1` as discussed above. A quality is always located in some region that represents the space of all possible values. Since the location is identified by some Uri, the corresponding region is the space of all Uris, represented by the individual us-1 of type UriSpace. Since the audio-mlda-1 expresses the MediaLocatorDescriptor, we link the quality to the description using the MediaUriParameter, which parametrizes the UriSpace. The audio-rec-1 itself is further classified as RootAudioSegmentRole, which indicates that the information object audio-rec-1 refers to the whole recording, in contrast to other AudioData representing parts of the whole recording. In COMM, audio data is always of type AudioData regardless of whether it refers to a whole or a part. The latter is represented using the decomposition pattern.

As the recording contains the voices of both Paul and Rita participating in the call-1, an automatic speaker change detection is employed to automatically segment the recording into several parts that refer to the different speakers. To model the two voices, we apply the COMM decomposition pattern to audio-rec-1 in order to represent the different segments of the recording. As an example, Figure 6 depicts the decomposition of audio-rec-1 into three segments identified as audio-segment-1 to audio-segment-3. The whole recording audio-rec-1 plays the InputSegmentRole, while the parts play an AudioSegmentRole in this pattern. Please note that in a real phone conversation at an emergency hotline there will typically be more than three speaker changes, i.e., more than three segments detected. However, for the purpose of demonstration it is already sufficient to consider only three segments. Please note further that in the COMM decomposition pattern, we do not identify the concrete person speaking in a segment. We only represent the number of different segments in the call. The annotation of the individual segments with the concrete person speaking is done only in the subsequent step.

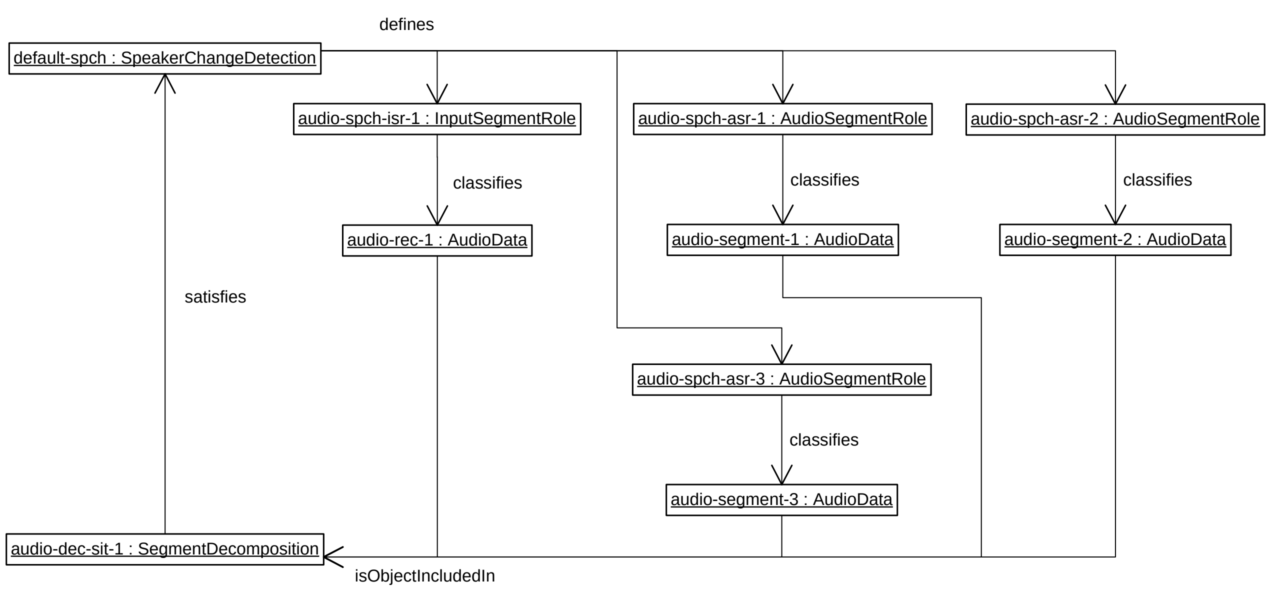


Fig. 6. Decomposition of the call recording into segments

To annotate the individual segments of the audio recording with the person speaking, an algorithm for speech detection is applied. It leverages the known characteristics of the officer's voice Rita to distinguish her from Paul's voice. At the example of audio-segment-2, Figure 7 shows the semantic annotation of the second audio segment with the speaker's voice, paul-1. Following the DnS pattern, the SemanticAnnotation satisfies a Method, which represents the semi-automatic method audio-speaker-method-1 for assigning a person to a segment. The audio-speaker-method-1 defines an AnnotatedDataRole which classifies the AudioData to be annotated, in our example audio-segment-2. It further defines a SemanticLabelRole, which classifies the semantic annotation label for the audio segment. In this case, the individual paul-1 representing the citizen Paul.

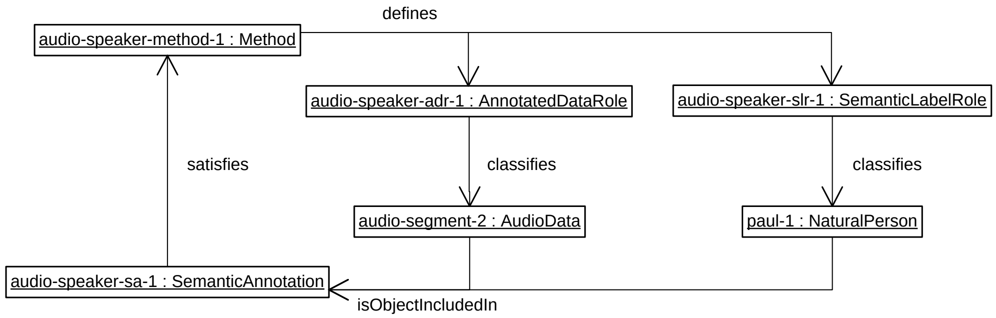


Fig. 7. Annotation of a call segment with the speaker Paul

Similar, also the segments of the audio recording are annotated with `rita-1` speaking. Figure 8 shows at the example of `audio-segment-1` the annotation of a segment with `rita-1`. The system at the emergency hotline has its own repository for the operators at the hotline and uses its own concept `Person` instead of DUL's concept `NaturalPerson`. This example shows that our core ontologies can reuse existing domain ontologies. This issue is discussed later in Section 6.4. Please note that in the case where multiple domain ontologies are reused, an alignment between these ontologies is necessary.

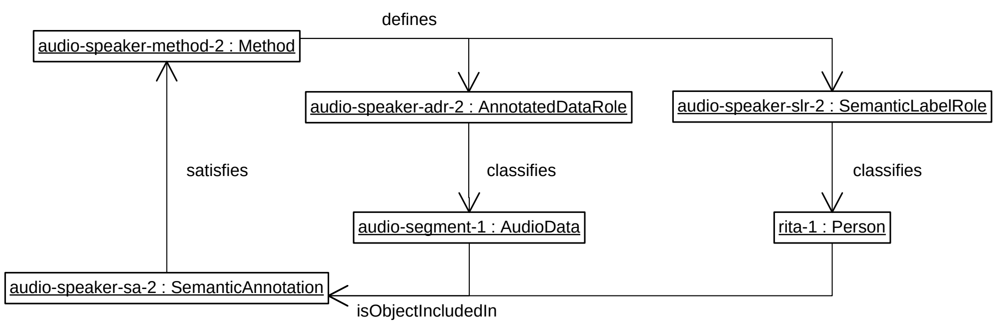


Fig. 8. Annotation of a call segment with the speaker Rita

In his phone call to the emergency hotline, Paul tells Rita that he thinks that the event of a snapped power pole represented by the individual `snapped-pp-1` has caused the event of a `power-outage-1` that shortly after occurred. This subjective but plausible causal relationship between events can be modeled using the causality pattern of the Event-Model-F as indicated in Figure 9. This is represented by two roles, one classifying the causing event `cause-snapped-pp-1` and the other classifying the effect event `effect-power-outage-1`. The causality pattern also foresees to attach a justification to the causal relationship, here the `laws-of-physics-1`.

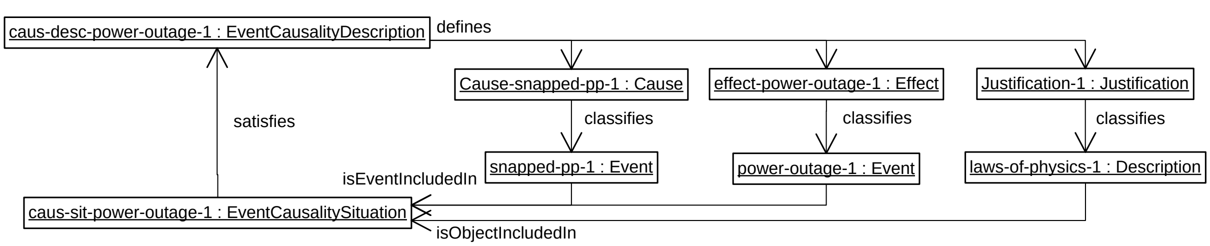


Fig. 9. Causal relationship between a snapped power pole and a power outage

In order to represent that the causal relationship modeled above is Paul's interpretation of how the `power-outage-1` happened, the Event-Model-F allows for representing different contextual views on events by using the interpretation pattern as shown in Figure 10. We interpret the `power-outage-1` event by assembling the different instantiations of the Event-Model-F patterns. These instantiations of the patterns are identified by the individuals part-sit-snapped-pp-1 and caus-sit-power-outage-1 from Figure 2 and Figure 9.

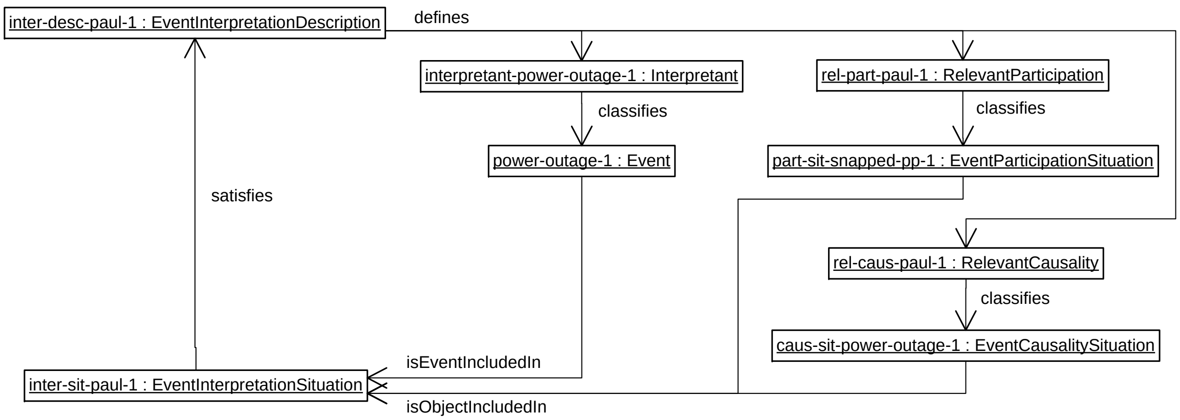


Fig. 10. Representing Paul's interpretation that a snapped power pole has caused the power outage

Subsequently to the call between Paul and Rita, the description of the call-1 event created by Rita in the emergency hotline system and the annotation of the event with the audio-rec-1 is transferred to the emergency control center. The emergency control center also receives Paul's interpretation of the cause for the power-outage-1, namely the event of a snapped power pole he observed. At the emergency control center, Henry represented as henry-1 works as emergency coordination manager. Among others, he supervises and filters incoming event descriptions. He reads the event description just created by Rita and listens to her conversation with Paul. As the snapped power pole might indeed have caused a large power outage in the city, Henry decides to send an urgent email message represented as message-1 with a task description task-1 to his forward liaison officer Marie. Henry attaches the event interpretation inter-sit-paul-1 of Paul to the message as well as the audio file of the recorded call audio-rec-1. The forward liaison officer Marie represented by marie-1 receives the message and drives to the location where the power pole snapped. Marie checks the power cables of the snapped pole and sees that it only serves a few houses in the neighborhood with electricity. Thus, she concludes that the snapped-pp-1 event cannot have caused the large power-outage-1 of the city that happened and that Paul's interpretation is likely wrong. Marie takes out her camera to take a picture of the snapped power pole to document her observation. She attaches the picture together with some manually added tags as documentary support for the snapped-pp-1 event. The results of Marie's investigation are send back to Henry, who decides that Marie's interpretation of the snapped-pp-1 is the right one and that the power outage must have been caused by some other reason.

With our core ontology X-COSIMO, we represent communications, processes, and associated tasks. The message exchanged between henry-1 and his forward liaison officer marie-1 is modeled using the communication pattern and is shown in Figure 11. It provides a contextual view on communication where henry-1 and marie-1 are playing the roles sender-1 and recipient-1 of the email message, respectively. Attached to the message is Paul's interpretation of the power outage inter-sit-paul-1 and the audio-recording-1 is also represented.

The task-1 assigned to Marie is of concept InspectionTask and is about to inspect the reported incident of a snapped power pole. The concept InspectionTask is defined in a domain ontology for tasks in emergency response, which is reused here. The assignment of the InspectionTask to Marie is modeled in X-COSIMO as shown in Figure 12. The action inspect-pp-1 is associated to task-1 and assigned to marie-1, who is considered as the task owner while Henry is considered in the role of an information provider. Several inputs to the task are available and accordingly represented as well: Paul's interpretation of the power outage inter-sit-paul-1, the recorded phone call audio-recording-1, and the message-1 that is described above.

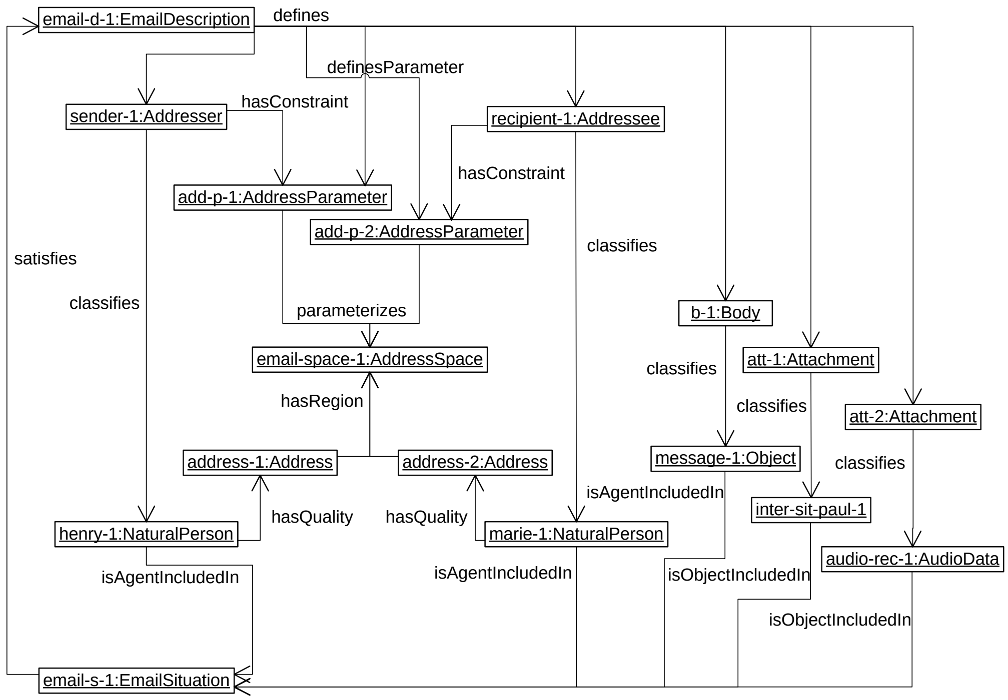


Fig. 11. Henry sends Marie a message with a task description

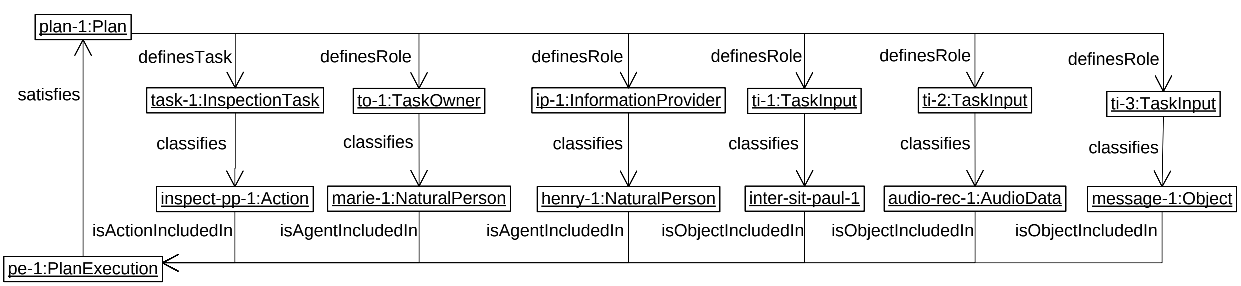


Fig. 12. Description of the task sent to Marie

Finally, from the viewpoint of Henry being a member of the emergency control center, all events that happen are parts of the major `flooding-1` event. For example, the parts of the `flooding-1` incident are the events of a snapped power pole `snapped-pp-1`, a power outage `power-outage-1` that occurred, phone calls the emergency hotline receives such as `call-1`, and many others. This composition of the large `flooding-1` event by smaller events is depicted in Figure 13. It uses the mereology pattern of the Event-Model-F. The composite event is `flooding-1` with its three example component events `snapped-pp-1`, `power-outage-1`, and `call-1`. The temporal constraints by which the composite event is composed into the three and further subevents is not shown in Figure 13. It is important to note that the `flooding-1` is of concept `Flooding`, which is not defined within the mereology pattern. Rather, a domain ontology for emergency response developed by a partner in the WeKnowIt project is being reused here within the Event-Model-F.

By the concrete example of a flooding event, we have shown how the complex, structured knowledge required in our scenario can be modeled. Multiple core ontologies are involved and combined to represent the knowledge shared in the scenario, namely the Event-Model-F, X-COSIMO, and COMM. They are designed such that they allow for an optimal interplay and integration. By this, we allow for an effective exchange of emergency response information between the different emergency response entities in our

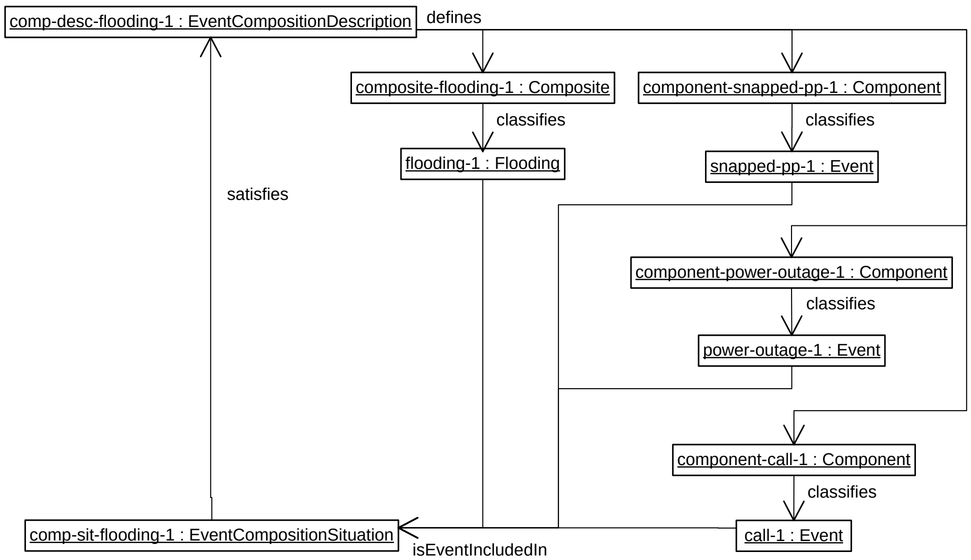


Fig. 13. Modeling the large flooding event by composition of smaller events

scenario and the systems these entities use. For the design of our core ontologies, we follow a specific ontology design approach, which we present in the following section.

## 4. Designing Core Ontologies

In order to provide a proper design approach for core ontologies, it is necessary to clarify what distinguishes a core ontology from foundational ontologies and domain ontologies. Thus, in Section 4.1, we briefly discriminate the concept of core ontologies from foundational ontologies and domain ontologies. In Section 4.2, we describe our design approach for building core ontologies. Essential choice for this design approach are the use of a foundational ontology as modeling basis and the use of ontology design patterns. The approach has been successfully applied to develop our three core ontologies Event-Model-F, COMM, and X-COSIMO.

### *4.1. Definition of Foundational Ontologies, Core Ontologies, and Domain Ontologies*

An ontology is a special kind of information object that allows for formally representing the relevant concepts and relations of a considered domain in a machine readable format (Oberle et al., 2009b; Oberle, 2006). Thus, ontologies are a means to explicitly specify conceptual models with logic-based semantics (Oberle, 2006). In addition, ontologies are often referred to have a collaborative aspect (Oberle et al., 2009b; Pinto et al., 2009), i.e., the formal conceptualization should be expressed by a shared view and consensus between several parties. To be a *shared* conceptualization is of high importance for ontologies intended to support large-scale interoperability (Oberle et al., 2009b) such as the core ontologies Event-Model-F, COMM, and X-COSIMO used in the emergency scenario in Section 2. In the literature, we find different types and classifications of ontologies such as (Oberle, 2006; Gangemi et al., 2002). In the context of this work, we follow the three-layered architecture of ontology libraries (Gangemi et al., 2004) and discriminate between foundational ontologies, core ontologies, and domain ontologies (Oberle, 2006). Core ontologies may use foundational ontologies as well as leverage domain ontologies. Thus, in the following sections, we briefly analyze and define the nature of foundational ontologies, core ontologies, and domain ontologies and discuss their relation to each other.

#### *4.1.1. Foundational Ontologies*

Foundational ontologies are generic across many fields (Oberle, 2006). They have a large scope and are highly reusable in different modeling scenarios (Borgo and Masolo, 2009). Thus, foundational ontologies serve reference purposes (Oberle, 2006) and aim at modeling the very basic and general concepts and relations (Borgo and Masolo, 2009; Oberle, 2006) that make up our world, e.g., objects, events, participation, and parthood. Foundational ontologies are heavyweight as they are rich in axiomatization (Borgo and Masolo, 2009), precisely defining the concepts in the ontology and their relations (Oberle, 2006). Synonyms of the term foundational ontology are generic ontology, upper level ontology, and top-level ontology (Euzenat and Shvaiko, 2007; Oberle, 2006; Cimiano and Reyle, 2006).

Examples of foundational ontologies are the ABC ontology and model (Lagoze and Hunter, 2001), the Basic Formal Ontology (BFO)[7] (Masolo et al., 2003), DOLCE (Borgo and Masolo, 2009; Gangemi et al., 2002; Masolo et al., 2003), the Object-Centered High-level REference ontology (OCHRE) (Schneider, 2003), the General Formal Ontology (GFO) (Herre et al., 2006), the OpenCyc ontology[8] (Lenat et al., 1990), and the Suggested Upper Merged Ontology (SUMO) (Niles and Pease, 2001). A detailed discussion of most of these foundational ontologies can be found in (Oberle, 2006).

Heavyweight foundational ontologies can have lightweight ones (Oberle, 2006), e.g., DOLCE and its lightweight version the DOLCE+DnS Ultralight (DUL) ontology[9]. The main purpose of heavyweight foundational ontologies is to serve as reference ontologies during development time (Oberle, 2006). A lightweight version of the foundational ontology is applied to facilitate reasoning at run time (Oberle, 2006). Foundational ontologies can be used as starting point for building core ontologies and domain ontologies (Oberle, 2006).

#### *4.1.2. Core Ontologies*

In contrast to foundational ontologies that span across many fields and model the very basic and general concepts and relations (Borgo and Masolo, 2009; Oberle, 2006) that make up our world, core ontologies provide a detailed abstract definition of structured knowledge in one of these fields, e.g., medicine, law, software services, personal information management, multimedia annotations, and others. By their nature, foundational ontologies are much broader than core ontologies. Core ontologies can be based on foundational ontologies and provide a refinement to foundational ontologies by adding detailed concepts and relations in their specific field. However, core ontologies are still very generic and span across a set of domains in a specific field (Oberle, 2006).

Examples of core ontologies are the MPEG7 ontology[10] for multimedia annotations (Hunter, 2005), a core ontology for software components and web services (Oberle et al., 2009a; Oberle, 2006; Mika et al., 2004), the ontology on the behavior of technical artifacts (Borgo et al., 2006), our core ontology Event-Model-F for capturing human experiences in terms of events and objects (Scherp et al., 2009a), the X-COSIMO ontology for personal information management (Franz et al., 2007), and the Core Ontology for Multimedia (COMM) for modeling multimedia annotations (Arndt et al., 2009; Staab et al., 2008; Arndt et al., 2007).

Core ontologies are situated in between the two extremes of foundational ontologies and domain ontologies (Oberle, 2006), described next. As foundational ontologies serve as a good modeling basis for core ontologies, so do core ontologies for domain ontologies.

#### *4.1.3. Domain Ontologies*

Finally, with domain ontologies we find representation of knowledge that is specific for a particular domain (Euzenat and Shvaiko, 2007; Oberle, 2006). Domain ontologies use terms in a sense that is relevant only to the considered domain and which is not related to similar concepts in other domains (Euzenat and Shvaiko, 2007). Domain ontologies can be very complex, i.e., they can comprise a very large

[7] http://www.ifomis.org/bfo
[8] http://www.cyc.com/cyc/opencyc/
[9] Available from: http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS_Ultralite
[10] http://metadata.net/mpeg7/

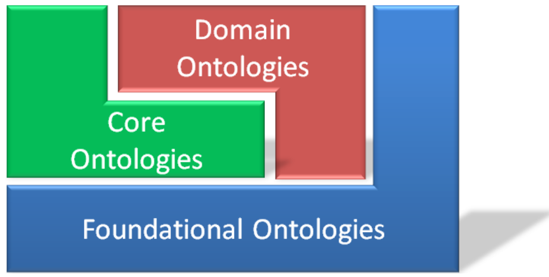


Fig. 14. Ontology stack of foundational ontologies, core ontologies, and domain ontologies

number of concepts and relations. They can make use of and can be based on foundational ontologies or core ontologies by specializing their concepts in the domain ontology (Oberle, 2006). Domain-specific ontologies can be used as external sources of background knowledge (Euzenat and Shvaiko, 2007), e.g., in combination with core ontologies. In addition, domain-specific ontologies may also be based directly on foundational ontologies like the disaster management ontology that is aligned with DOLCE (Babitski et al., 2009, 2011). Further examples of domain ontologies are a soccer ontology developed in the SmartWeb project (Oberle et al., 2007), the Foundational Model of Anatomy[11] as a domain-specific medical ontology describing the anatomy of the human body (Rosse and Mejino, 2003), the RadLex Lexicon for Radiology[12] (Kundu et al., 2009), and other medical ontologies such as Snomed (Cote et al., 1993), the Gene Ontology (Ashburner, 2000), and Galen (Rector and Horrocks, 1997).

#### 4.1.4. Summary

The relation between foundational ontologies, core ontologies, and domain ontologies is illustrated by the ontology stack in Figure 4.1.4. The bottom of the figure shows the foundational ontologies. They may be used and refined by core ontologies in the middle layer (Mika et al., 2004). The further, core ontologies as well as foundational ontologies can be used for defining semantically precise domain ontologies. The borderline from core ontologies to domain ontologies is not clearly defined. Core ontologies intend to be generic within a field that spans across multiple domains (Oberle, 2006). Similarly, also the distinction between foundational ontologies and core ontologies is not clearly defined (Oberle, 2006). However, the distinction is meaningful and useful for building ontology libraries as foundational ontologies, core ontologies, and domain ontologies serve different purposes (Gangemi et al., 2004).

### 4.2. Design Approach for Core Ontologies

We have developed a guideline describing the design approach of core ontologies. This design approach for core ontologies is described in the following sections along the non-functional properties identified, namely axiomatization and formal precision, modularity, extensibility, reuseability, and separation of concerns.

#### 4.2.1. Axiomatization and Formal Precision

When designing an ontology, it is desirable to use a solid and sound modeling basis (Oberle, 2006). Thus, our approach foresees the use of a foundational ontology for designing the core ontology (Mika et al., 2004; Guarino and Welty, 2000b). We align the concepts and relations defined in the core ontology to the basic categories of human cognition investigated in philosophy, linguistics, and psychology (Oberle, 2006). These basic categories are manifested in the foundational ontology.

The alignment of the foundational ontology with the core ontology also includes the adoption and specialization of the formal semantics of the foundational ontology in the core ontology. While the axiomatization of a foundational ontology enables validating the upper-level semantics of the knowledge expressed

[11] http://sig.biostr.washington.edu/projects/fm/index.html
[12] http://www.radlex.org/

with it, the alignment of the core ontology with the foundational ontology provides support for validating the more specific semantics of the concepts and relations defined in the core ontology.

A well-designed foundational ontology is very diligent with respect to the ontological choices to which it commits (Oberle, 2006), e.g., the selection of the most abstract concepts that are modeled. Thus, when reusing such a foundational ontology for designing a core ontology the engineer is also requested to carefully think about his or her design choices (Oberle, 2006). Developing foundational ontologies is extremely hard but it needs to be conducted only once (Oberle, 2006). An ontology engineer should strive for applying a foundational ontology that has proven its usefulness when designing a new core ontology or domain ontology (Oberle, 2006).

For designing our core ontologies, we have based them on the foundational ontology DOLCE and have carefully aligned our core ontologies with DOLCE's lightweight version DOLCE+DnS Ultralight (DUL). DOLCE aims at capturing the ontological categories underlying natural language and human common sense. It has a minimal core that includes only the most general concepts and patterns and is well suited for modularization. Additional theories such as Descriptions and Situations, Ontology on Information Objects, or the Ontology of Plans can be integrated when necessary (Oberle, 2006). We chose DOLCE as it already has proven to serve as good modeling basis for core ontologies. DOLCE has been successfully applied to design ontologies in different domains such as law, biomedicine, agriculture, and software components and web services (Oberle, 2006).

It is important to note that the axiomatization of the core ontology is mainly conducted for checking consistency of the ontology during design time. Thus, when designing a core ontology, we use reasoning to verify if the concepts and relations defined in the core ontology are consistent. Once, the design of the core ontology is finished, it can be used in concrete applications by developing an appropriate application programming interface (API). The API is used to work on the knowledge represented with the core ontology, i.e., to create instances of its patterns and concepts. If the API behaves as expected, i.e., the represented knowledge is conform to the core ontology's axiomatization, the axiomatization of the core ontology may be abstracted into efficiently processable languages, e.g., using the abstraction methods proposed by Ren et al. (2010). To ease the access to ontological knowledge from software applications and to improve the efficiency of designing and implementing ontology APIs, we have developed a semi-automatic API generation process based on model driven engineering (Scheglmann et al., 2010).

#### *4.2.2. Modularity*

Even well designed core ontologies are usually large and cover more knowledge than might be required in a specific application domain (Gangemi and Presutti, 2009). Thus, concrete systems will commonly use only portions of it. In this case, it is hard to reuse only the "useful" pieces of such a monolithic core ontology (Gangemi and Presutti, 2009).

Core ontologies (and domain ontologies) have a better design when applying ontology design patterns captured by the foundational ontology it uses (Oberle, 2006). By using ontology design patterns, it allows for selecting the parts of the ontology in a concrete application that are actually needed and used (Gangemi and Presutti, 2009). Thus, our design approach for core ontologies builds upon a foundational ontology that supports a pattern-oriented design approach (Oberle et al., 2006). DUL provides such a pattern-oriented approach.

Ontology design patterns shall not be too specific or too close to a particular domain (Mika et al., 2004). This would disallow the application of the pattern in other domains of the field covered by the core ontology. On the other hand, the patterns shall also not be too generic as reuse in a concrete domain would be hampered. A too generic pattern is hard to apply in a specific domain (cf. competency question for ontology design patterns in (Presutti and Gangemi, 2008)). The scope of a core ontology itself is defined through the scope of its patterns.

#### *4.2.3. Extensibility*

Foundational ontologies provide a high-level, abstract vocabulary of concepts and relations that are likely to be used in current and future application domains. In order to provide a solid basis for future extensions of core ontologies, a precise alignment of the concepts defined in a core ontology with the high-level concepts of a foundational ontology is conducted using our design approach. By this precise alignment, new patterns can be added to the core ontology without affecting the existing patterns. In addition to adding new patterns, the existing patterns of a core ontology can be extended. This is typically conducted by specializing the existing concepts and properties defined in the patterns. Finally, besides the already connected core ontologies within a specific application also further core ontologies can be developed and integrated if necessary.

#### *4.2.4. Reuseability*

For modeling complex, structured knowledge, reuse can happen on different levels, e.g., on the level of ontology design patterns, core ontologies, and domain-specific ontologies. Different patterns in the core ontology provide different descriptions of concepts defined in it. By splitting up core ontologies into different parts they allow for reusing the structured knowledge defined within the core ontology's design patterns among different applications. This refers to the issue of extensibility discussed in Section 4.2.3.

In addition, the core ontologies can be combined with domain-specific knowledge. In the ideal case, domain ontologies reuse the ontology design patterns defined in core ontologies by specializations of the ontology design patterns (Gangemi and Presutti, 2009). However, our approach does not require that domain ontologies are based on a foundational ontology or a core ontology. In fact, we explicitly consider both options as it cannot be assumed that all domain ontologies are aligned with a foundational ontology or core ontology. This is achieved by using the Descriptions and Situations ontology design pattern of DOLCE. Here, the roles defined within a contextual situation can refer to a domain ontology that is either carefully aligned with DOLCE, aligned to the core ontology, or that is completely independent.

#### *4.2.5. Separation of Concerns*

Our design approach supports the separation of concerns by defining the structured knowledge in the core ontology and leaving all domain-specific aspects out of it. This is achieved again with the Descriptions and Situations ontology design pattern. The structured knowledge of the concrete field of the core ontology is captured by its ontology design patterns, e.g., participation, causality, and documentation for the Event-Model-F, the annotation and decomposition in COMM, and the communication pattern in X-COSIMO. This structured knowledge is specified by the roles defined within these patterns, i.e., the `defines` relations of the `Descriptions`. The domain knowledge is only referred to by the roles classifying the events and objects used. By this, the core ontologies are independent of any concrete domain that makes use of the concepts defined by them. In addition, core ontologies provide support to include individuals defined in some domain ontologies.

## 5. Examples of Core Ontologies

In this section, we describe the design of our core ontologies Event-Model-F, COMM, and X-COSIMO with focus on the parts relevant to model the emergency response scenario in Section 3. Our core ontologies are based on the foundational ontology DOLCE+DnS Ultralight (DUL). DUL defines the class `DUL:Event` next to the disjoint upper classes `DUL:Object`, `DUL:Abstract`, and `DUL:Quality`. The definition of `Event` has been specialized from the formal definition in DOLCE as an entity that exists in time. The class `Object` stands for entities that exist in space such as living things as well as non-living and social and cognitive entities. A `Quality`[13] is a characteristic of an object or an event. It has a value that is represented as a point or area in some `Abstract`. The class `Abstract` represents value spaces, e.g., the space of natural numbers or the time of a day. Typically, we do not prescribe specific `Abstracts` that are to be used. We rather refer to the generic `Abstracts` already defined in DUL such as the regions `DUL:TimeInterval`, `DUL:SpatioTemporalRegion`, and `DUL:SpaceRegion`.

For modeling our core ontologies, we make use of the ontology design patterns Descriptions and Situations (DnS) and the ontology of Information Object (IO) (Borgo and Masolo, 2009). The DnS pattern

[13]Also called trope, see `http://plato.stanford.edu/entries/tropes/`

provides an ontological formalization of context (Oberle, 2006; Gangemi and Mika, 2003). With DnS one can reify `Events` and `Objects` and describe the n-ary relation that exists between multiple individuals of them (cf. Minsky (1974)). Thus, it allows for a formally precise representation of different, contextualized views on events. The IO ontology pattern describes the relation between an information object such as a poem, song, and a story and their actual physical realization in form of a printed book, recorded track, and a movie taken (Oberle, 2006). We describe the design of our three core ontologies the Event-Model-F, COMM, and X-COSIMO in the following sections.

### *5.1. Event-Model-F—Core Ontology of Events for Representing Human Experience*

The core ontology Event-Model-F for representing human experience allows for modeling the different relationships between events and objects. The requirements to the core ontology for events have been derived from existing event models in various domains such as music, journalism, multimedia, news, cultural heritage, and knowledge representation (Wang et al., 2007; Raimond and Abdallah, 2007; IPTC, 2008; Doerr et al., 2007; Mueller, 2008; Francois et al., 2005; Jain, 2008; Ekin et al., 2004). Identified requirements are representing (1) participation of living and non-living objects in events, (2) temporal duration of events, and (3) spatial extension of objects. In addition, three kind of event relationships shall be supported, namely (4a) mereological (composition of events), (4b) causal, and (4c) correlation. The common model shall also support the experiential aspect, i.e., the (5) annotation of events with sensor data such as media data, and allow for (6) different interpretations of events. Existing models almost fully support participation, time and space, and the experiential aspect. However, they substantially lack in the mereological, causal, and correlation relationships, and event interpretations. Here, we find different limitations or even no support, e.g., only simple mereological relationships in (Raimond and Abdallah, 2007) and causal relationships in (Doerr et al., 2007). Correlation is not considered at all and event interpretations are only mentioned in (Jain, 2008) but remain future work.

With respect to the requirements, we introduced specialized instantiations of the DnS pattern. Here, the participation of objects in events (1) is implemented by the participation pattern. It also provides for modeling the absolute time and location of events (2) and objects (3). The mereology pattern, causality pattern, and correlation pattern implement the structural relationships between events (4a-4c). In addition, the mereology pattern allows for modeling the relative temporal relations and relative spatial relations between events (2) and objects (3). In order to express such relative temporal relations between events, one can facilitate the provided means of DOLCE such as the formalization of Allen's Time Calculus[14]. The documentation pattern provides for annotating events (5). It can be seamlessly linked with other ontologies, e.g., the Core Ontology for Multimedia (Arndt et al., 2007) for precisely describing digital media data like images and videos. Finally, the interpretation pattern supports different event interpretations (6).

We use the DnS pattern for representing occurrences in the real world, i.e., the events and objects we are modeling. These occurrences are subject to discussion and interpretation and may not be objectively observable. The DnS pattern allows for representing different opinions about events and their participating objects. This feature is not provided by DOLCE's participation relation. In the following, we present the ontology patterns of the Event-Model-F that have been employed to model the scenario in Section 3 and illustrate them in diagrams. This comprises almost all patterns of the Event-Model-F, expect from the correlation pattern. A complete description of the Event-Model-F can be found in (Scherp et al., 2009a).

#### *5.1.1. Participation Pattern*

The participation pattern of the Event-Model-F enables to formally express the participation of objects in events. As shown in Figure 15, participation is expressed by an `F:EventParticipationSituation` that `satisfies` an `F:EventParticipationDescription`. The situation includes the `Event` being described and the `Objects` participating in this event. The `EventParticipationDescription` classifies the described event and its participants by using the concepts `F:DescribedEvent` (specialized from `DUL:EventType`) and the object role `F:Participant` (specialized from `DUL:Role`). The

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

concept `DescribedEvent classifies` the `Event` that is described by the participation pattern, e.g., the event of a flooded cellar. Likewise, instances of `Participant` classify objects as participants of the event. For example, the citizen calling the emergency hotline to report about the flooded cellar. Instances of `Participant` can be roles defined in some domain ontology as indicated in Figure 15. For example, an emergency response ontology that defines the role of a person being affected, i.e., the emergency subject such as a `CitizenRole`, and the role describing the rescue staff such as a `FiremanRole`. Besides the role an object can play in a specific participation pattern, also the described event and its participating objects themselves can be defined in some domain ontology as indicated in Figure 15.

The parameter `F:LocationParameter` describes the general spatial region where the objects are located. It `parametrizes` a `DUL:SpaceRegion` and defines a property `DUL:isParameterFor` to the `Participant` role. The `Object` that is classified by the `Participant` has a `Quality` with the property `DUL:hasRegion` of a `DUL:SpaceRegion`. Thus, using the `F:LocationParameter` we can define the location(s) represented by `DUL:SpaceRegions` that are relevant for describing the event in a given context. For example, when quenching a house fire all firemen have their specific location within and around the building. The `F:LocationParameter` can then be used to describe in general that the firemen where at that specific house, e.g., in form of some longitude-latitude rectangular. Thus, we do not need to explicitly state where the individual firemen are. The `F:TimeParameter` describes the general temporal region when the event happened. It `parametrizes` a `DUL:TimeInterval` and defines a property `DUL:isParameterFor` to the `DescribedEvent` role. For example, one can state that the house fire happened on June 13, 2006.

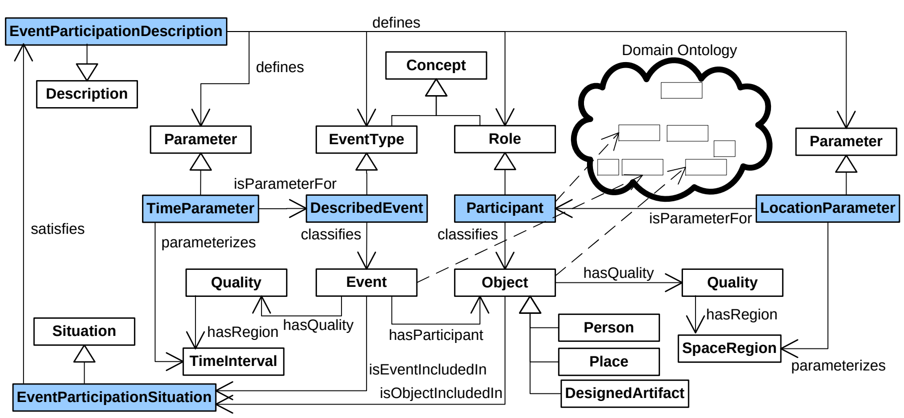


Fig. 15. Participation pattern

### *5.1.2. Mereology Pattern*

Events are commonly considered at different abstraction levels depending on the view and the knowledge of a spectator. For instance, the event of a flooded cellar may be considered as such or as part of the larger event of a flooding in which many other (smaller) incidents occur. The mereology pattern shown in Figure 16 enables expressing such mereological relations as composition of events. The composite event is the "whole" and the component events are its "parts". Formally, a `F:EventCompositionSituation` includes one instance of an event that is classified by the concept `F:Composite` and many events classified as its `F:Component`(s). Accordingly, an `EventCompositionSituation satisfies` a `F:CompositionDescription` that `defines` the concepts `Composite` and `Component` for classifying the composite event and its component events.

Events that play the `Component` role may be further qualified by temporal, spatial, and spatio-temporal constraints. As events are formally defined as entities that exist in time and not in space (Scherp et al., 2009a), constraints including spatial restrictions are expressed through the objects participating in the component event. For instance, a `Component` event may be required to occur within a certain time-interval, e.g., the second week of June 2009. Depending on its objects, a `Component` event may also

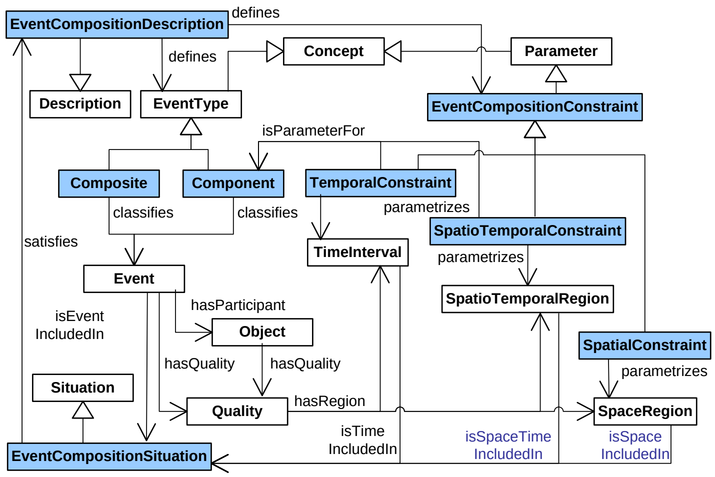


Fig. 16. Mereology pattern

happen in a certain spatial region. For example, the flooding of a town should be composed of events that have objects associated to it, which have some certain range of longitude and latitude. Finally, events and the objects bond to it 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 other areas. Any such constraints are formally expressed by one or multiple instances of the `F:EventCompositionConstraint`. Thus, with the composition pattern, events may be arbitrarily temporally related to each other, i.e., they might be disjoint, overlapping, or otherwise ordered. In order to express such relative temporal relations between events, one can facilitate the provided means of DOLCE such as the formalization of Allen's Time Calculus[15].

### 5.1.3. *Causality Pattern*

Causality is the philosophical problem investigating the existence of any special "tie" binding causes and effects together (Itkonen, 1983). It can be questioned either as "Why" or "How" (Itkonen, 1983). 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 (Itkonen, 1983). Events are the most natural concept to serve for defining causal relations (Quinton, 1979). In fact, causes and effects are two specific types of events (Itkonen, 1983; Lombard, 1986). A causal relationship is always justified by some (maybe implicit) underlying causal theory.

We designed a causality pattern as depicted in Figure 17. The pattern defines two `EventTypes` called `F:Cause` and `F:Effect` which `classify` `Events`. It further defines a `DUL:Description`, which is classified by a `F:Justification`. By this, the pattern explicitly expresses the causal relationship between the cause and the effect under the justification of some theory. A theory might be an opinion, a scientific law, or not further specified. For example, during a heavy storm, a power outage event might occur caused by a snapped power pole event. The `Justification` of this causal relationship is the laws of physics.

### 5.1.4. *Documentation Pattern*

Documentary evidence for an event may be given by arbitrary objects, e.g., some sensor data or media data, or by other events. Formally, this relation is expressed in F by the documentation pattern depicted in Figure 18. It defines the concept `F:DocumentedEvent` that classifies the documented event and the concept `F:Documenter` that classifies the documentary evidence for that event. This evidence can be expressed by any specialization of an `Object`, e.g., a digital photo taken with a cell phone during an incident, or a specialization of `Event`. For example, digital media data like images and videos can be classified as `Documenter` and precisely described using the Core Ontology for Multimedia (Arndt

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

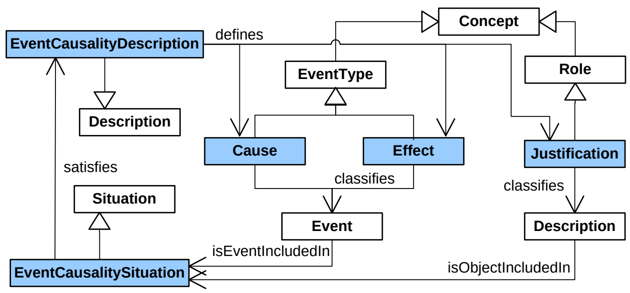


Fig. 17. Causality pattern

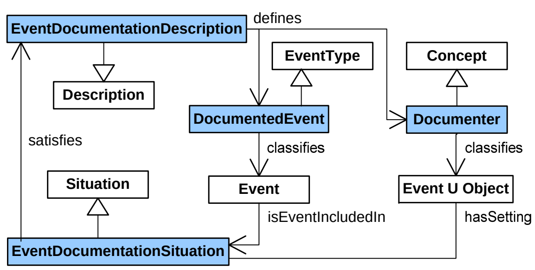


Fig. 18. Documentation pattern

et al., 2007) described in Section 5.2. Objects are documented via the events in which they participate (see participation pattern in Section 5.1.1).

### *5.1.5. Interpretation Pattern*

The perception of events as occurrences in the real world heavily depends on the context and point of view of the observer. Such different, context-dependent event interpretations can be described formally by instantiating the different Event-Model-F patterns presented so far and binding them together with the interpretation pattern depicted in Figure 19. Each pattern models a single, specific interpretation of an event by associating *participations*, *mereological*, *causal*, and *correlative* relationships, as well as *documentations* relevant in the context of a specific *interpretation*. In the emergency scenario, 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 a more serious case of a damaged power plant. Both consider the same event of a power outage, however, consider it from different points of view that involve other events and objects in different patterns.

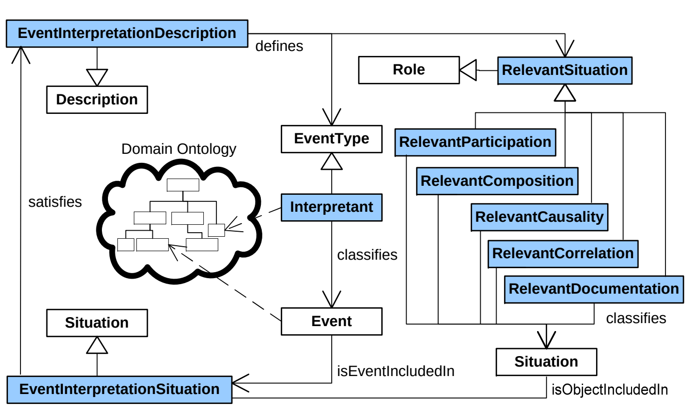


Fig. 19. Interpretation pattern

Formally, the interpretation pattern shown in Figure 19 defines a `F:Interpretant` that is specialized from `EventType` and classifies the interpreted `Event`. The `Interpretant` might be defined in some domain ontology and determines how an event is interpreted, e.g., as emergency incident in the case of the emergency control center or as news event described in a news paper. Within each interpretation, we classify the `F:RelevantSituations`, namely the situations satisfying the participation, mereology, causality, correlation, and documentation. These are defined as specializations of `RelevantSituation`.

### 5.2. COMM—Core Ontology for Multimedia

The Core Ontology for Multimedia (COMM) (Arndt et al., 2007) models the domain of multimedia content and annotation, and is based on MPEG-7. In contrast to other approaches to modeling MPEG-7 as an ontology (Hunter, 2005; Bloehdorn et al., 2005; Garcia and Celma, 2005; Isaac and Troncy, 2004; Tsinaraki et al., 2004), COMM is not designed as a one-to-one mapping, but provides a set of patterns that cover the core and repetitive building blocks of MPEG-7. In the following, we discuss the patterns of COMM used for the emergency response scenario in Section 3. Three repetitive structures have been identified in MPEG-7:

**Decomposition.** MPEG-7 provides descriptors for spatial, temporal, spatio-temporal and media source decompositions of multimedia content into segments. A segment is the most general abstract concept in MPEG-7 and can refer to a region of an image, a piece of text, a temporal scene of a video, or even to a moving object tracked during a period of time.

**Annotation.** MPEG-7 defines a very large collection of descriptors that can be used to annotate a segment. These descriptors can be low-level visual features, audio features or more abstract concepts. They allow the annotation of the content of multimedia documents or the media asset itself.

**Nested Data Structures.** Descriptors in MPEG-7 are nested structures containing different kinds of data. Except for the semantic annotation, data refers to strings or numerical values such as the encoding of an image or the values of a color histogram.

These three structures are modeled as patterns of our ontology COMM. The patterns are the decomposition pattern, annotation pattern, and the digital data pattern. Before we discuss the patterns in detail, we will introduce some central concepts that are present in all the patterns:

**Digital Data.** Within the domain of multimedia annotation, the notion of digital data is central—both the multimedia content being annotated and the annotations themselves are expressed as digital data. We consider `DigitalData` entities of arbitrary size to be `InformationObjects`, which are used for communication between machines. The IO design pattern states that `Descriptions` are expressed by `InformationObjects`, which have to be about facts (represented by individuals of type `Entity`). These facts are settings for `Situations` that have to satisfy the `Descriptions` that are expressed by `InformationObjects`. This chain of constraints allows the modeling of complex data structures to store digital information.

**Multimedia Data.** This encapsulates the MPEG-7 notion of multimedia content such as images, audio, text, open document format (ODF) documents, and is a subconcept of `DigitalData`. `MultimediaData` is an abstract concept that has to be further specialized for concrete multimedia content types (e.g. `AudioData` corresponds to the data representing the recorded audio signal). According to the IO pattern, `MultimediaData` is realized by some physical `Media`, e.g., `MediaProfile`, which contains information about the storage location, file type, and others. This concept is needed for annotating the physical realization of multimedia content.

**Media.** `Media` objects represent the realizations of the information represented as `MultimediaData`. In the context of electronic devices this refers to files accessible via some protocol, but might in a more general setting also refer to physical realizations such as a painting.

**Method.** A `Method` refers to some manual process, and its subclass `Algorithm` to a (semi-)automatic process that processes some input and generates output. Examples are the extraction of features, e.g., a `DominantColorExtractionAlgorithm`, or the manual annotation of content with a semantic concept.

**Input/Output Roles.** Methods always define InputRoles and OutputRoles. The input role classifies the object that is processed, while the output role classifies the results. For example, a segmentation defines exactly one InputSegmentRole which classifies the image that is being segmented and at least one OutputSegmentRole which classifies the resulting subsegments.

In the following subsections, we present the three patterns of COMM. For expressing concrete descriptors of MPEG-7 a specialization of the concepts involved in each pattern is required, but for easier comprehension we discuss the pattern on a more abstract level. The details are comprehended by the ontology itself which is available through our website: http://west.uni-koblenz.de/Research/ontologies/.

*5.2.1. Digital Data Pattern*

The digital data pattern is defined as depicted in Figure 20. DigitalData expresses StructuredDataDescriptions, which define meaningful labels for the information contained by DigitalData. This information is represented by literals such as scalars, matrices, strings, rectangles, or polygons. In DOLCE terms, these values are represented as a Quality that is associated with the DigitalData and located in Regions. In the context of a Description, these Regions are parametrized by Parameters. StructuredDataDescriptions define StructuredDataParameters, for which the qualities located in the parametrized Regions assign values to the DigitalData. Referring to the example in Figure 5 (cf. Section 3), we see that the formalization of data structures so far is not sufficient. Complex MPEG-7 types can include nested types that again have to be represented by StructuredDataDescriptions. In our example, the MediaInstanceDescriptor contains the MediaLocatorDescriptor. The digital data pattern covers such cases by allowing a DigitalData instance to be about another DigitalData instance which expresses the nested StructuredDataDescription.

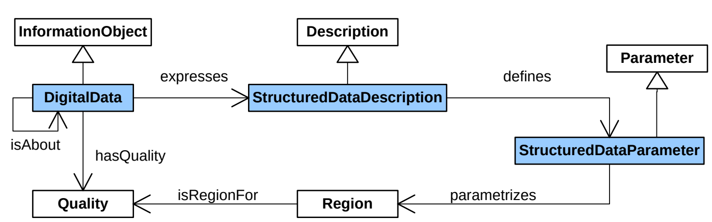


Fig. 20. Digital data pattern

*5.2.2. Decomposition Pattern*

Following the DnS pattern, we consider the decomposition of MultimediaData to be a Situation that satisfies a Method. More specifically the Situation is a SegmentDecomposition. The description refers to an algorithm such as a SegmentationAlgorithm (cf. Figure 6) or to a manual segmentation Method like a tool that allows the user to draw a bounding box around a depicted face. Of particular interest with respect to the decomposition pattern are the roles that are defined by a SegmentationAlgorithm or a Method. The input to a segmentation is MultimediaData that is classified by an InputSegmentRole, while the MultimediaData referring to the output segments are classified by OutputSegmentRoles. These data entities have as setting the SegmentDecomposition situation. OutputSegmentRoles as well as SegmentDecompositions are then specialized for specific types of media (according to the segment and decomposition hierarchies of MPEG-7 (MPEG-7 (2001), part 5, section 11). The decomposition pattern is depicted in Figure 21. Please note that the concept TextData is actually not provided by COMM. It has been added to the decomposition pattern by the Ontology for Knowledge Acquisition (OAK) (Iria, 2009) in order to support the annotation and decomposition of textual data.

In terms of MPEG-7, unsegmented (complete) multimedia content also corresponds to a segment. Consequently, annotations of complete multimedia content start with a root segment. In order to designate

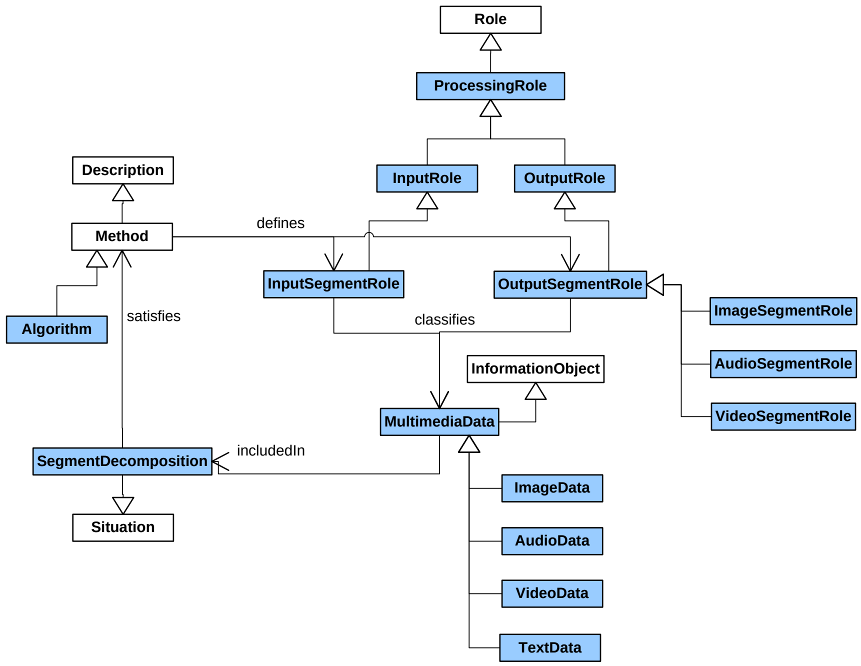


Fig. 21. Decomposition pattern

MultimediaData instances that correspond to these root segments the decomposition pattern provides the RootSegmentRole concept. Note that RootSegmentRoles are not defined by Methods which describe SegmentDecompositions. They are rather defined by Methods which cause the production of multimedia content. These methods as well as annotation modes which allow the description of the production process (e.g. MPEG-7 (2001), part 5, section 9) are currently not covered by our ontology. Nevertheless, the prerequisite for enhancing the COMM into this direction is already given.

The decomposition pattern also reflects the need for localizing segments within the input segment of a decomposition as each OutputSegmentRole requires a MaskRole that classifies some DigitalData which expresses one LocalizationDescriptor. The latter describes the location of a segment, e.g., start and end time of an audio segment. In our example, we did not include this required information in order to reduce the complexity of the diagram.

### *5.2.3. Annotation Pattern*

In this section, we describe the attachment of metadata, i.e., annotations to both MultimediaData or Media as depicted in Figures 22, 23, and 24. In general, we distinguish three annotation patterns. The content annotation pattern models the annotation of MultimediaData with metadata represented as DigitalData, i.e., in general MPEG-7 descriptors such as a dominant color descriptor. The media annotation pattern is used to attach metadata to the media, e.g., the file name or file size. Finally, the semantic annotation pattern formalizes the semantic annotation of MultimediaData. A semantic annotation refers to some individual of a domain ontology.

Each annotation pattern consists of an Annotation (subclass of Situation) that satisfies a Method. The Method defines an InputRole that is specialized to either AnnotatedDataRole or AnnotatedMediaRole, depending on what is annotated. Furthermore it specifies at least one OutputRole, which is specialized to AnnotationRole for the content annotation pattern and media annotation pattern and specialized to SemanticLabelRole in case of the semantic annotation pattern, respectively. In the following, we discuss the individual patterns in more detail.

We start with the content annotation pattern (cf. Figure 22). As ContentAnnotations we understand Situations that include annotated media as MultimediaData classified by an AnnotatedDataRole. The metadata is represented by DigitalData, which is classified by AnnotationRoles. The roles are defined by an Algorithm or a Method, respectively. The actual metadata that is represented by a `DigitalData` entity depends on the `StructuredDataDescription` that the metadata comprises. These structured descriptions are formalized using the digital data pattern (see Section 5.2.1) and typically refer to MPEG-7 descriptors, although this is not required. The content annotation pattern is used to represent rather technical metadata such as extracted low-level features.

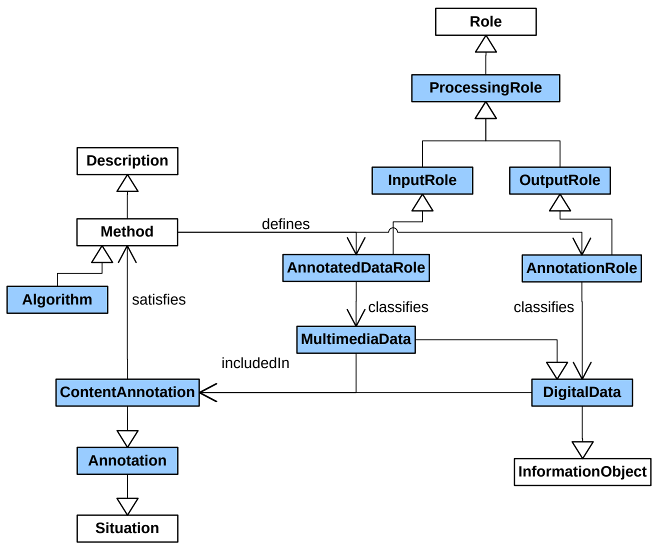


Fig. 22. Content annotation pattern

The media annotation pattern forms the basis for describing the physical instances of multimedia content (cf. Figure 23). It differs from the content annotation pattern in only one respect: it is the `Media` that is being annotated and therefore plays an `AnnotatedMediaRole`. The situation is specialized to a `MediaAnnotation`. An example of the application of this pattern is given in Figure 5 in Section 3. The example depicts a file realizing an audio recording. The audio recording is annotated with the MPEG-7 media profile descriptor, which in the example contains the URI to the file.

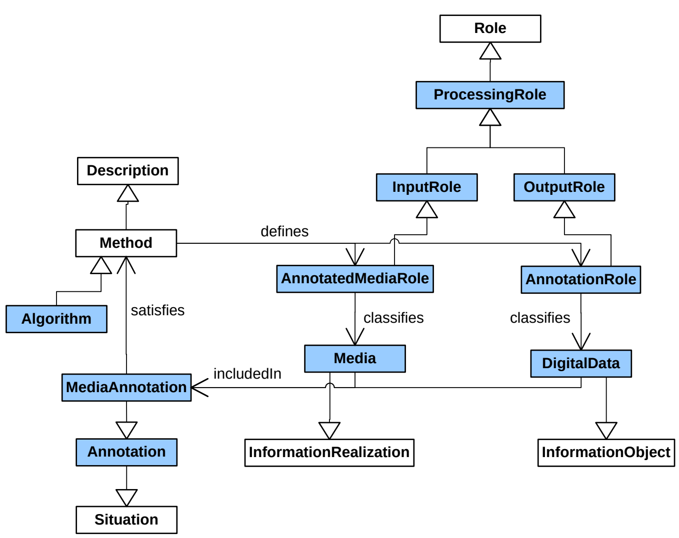


Fig. 23. Media annotation pattern

Finally, COMM also provides the semantic annotation pattern depicted in Figure 24. MPEG-7 provides the means to model semantics as MPEG-7 descriptors (see MPEG-7 (2001), Part 5, Section 12). However, in the context of an ontology-based approach like COMM the integration of domain-specific ontologies is more appropriate then using the MPEG-7 descriptor for semantics. Thus, for the semantic annotation COMM relies on domain-specific ontologies that represent, e.g., real world entities that are depicted in the annotated multimedia content. Consequently, the semantic annotation pattern specializes the content annotation pattern to allow the connection of multimedia descriptions with domain descriptions provided by independent domain ontologies.

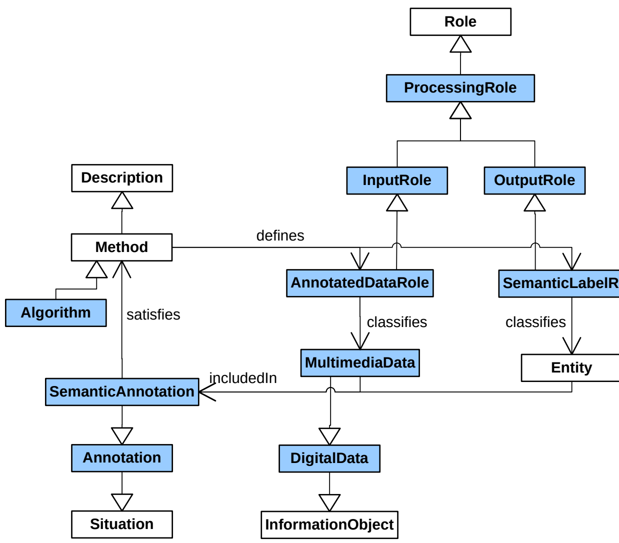


Fig. 24. Semantic annotation pattern

### 5.3. X-COSIMO—Core Ontology for Personal Information Management

The Cross-Context Semantic Information Management Ontology (X-COSIMO) provides a formally precise representation of personal information and associated tasks to foster its reuse across personal information management applications such as task managers, email clients, and file management tools. It is used among others for personal information management in the X-COSIM semantic desktop (Franz et al., 2007). In the following, we detail on two of the patterns provided by X-COSIMO that are employed to describe the emergency scenario, namely the communication pattern and the task pattern.

#### 5.3.1. Communication Pattern

The aim of the communication pattern is to enable a unified view onto communication and to represent information dealt in the context of communication for reuse in further contexts while maintaining information linkage. For achieving the first, the communication pattern provides a conceptual view of communication as developed by Jakobson (Jakobson, 1960). In his work, Jakobson defines the concept of a `Message` that is about something which he calls `Context`. The context is transmitted via a `Contact`—a connection between the `Addresser` and `Addressee`—and that is expressed within a `Code`. Such a model generalizes communication so that arbitrary communication modes such as chat, phone, and email can be represented consistently. For achieving the second goal, the communication pattern is designed using the DnS pattern as a `CommunicationDescription` (cf. Fig. 25) that represents the communication model of Jakobson.

The `CommunicationDescription` defines the roles `Addresser`, `Addressee`, `Contact`, and `Message` as common roles for any kind of communication as defined by Jakobson. Jakobson's concept of `Context` and `Code` are not defined by the description as existing ontologies for media annotation and decomposition such as COMM express these aspects in a more general way and are applicable also to other contexts than communication. Additionally, a `CommunicationDescription` also defines the `CommunicationParameter` `ConversationStart` and `AddressParameter`, which express constraints on the classified `CommunicationEvent` and the `Agents` that occur

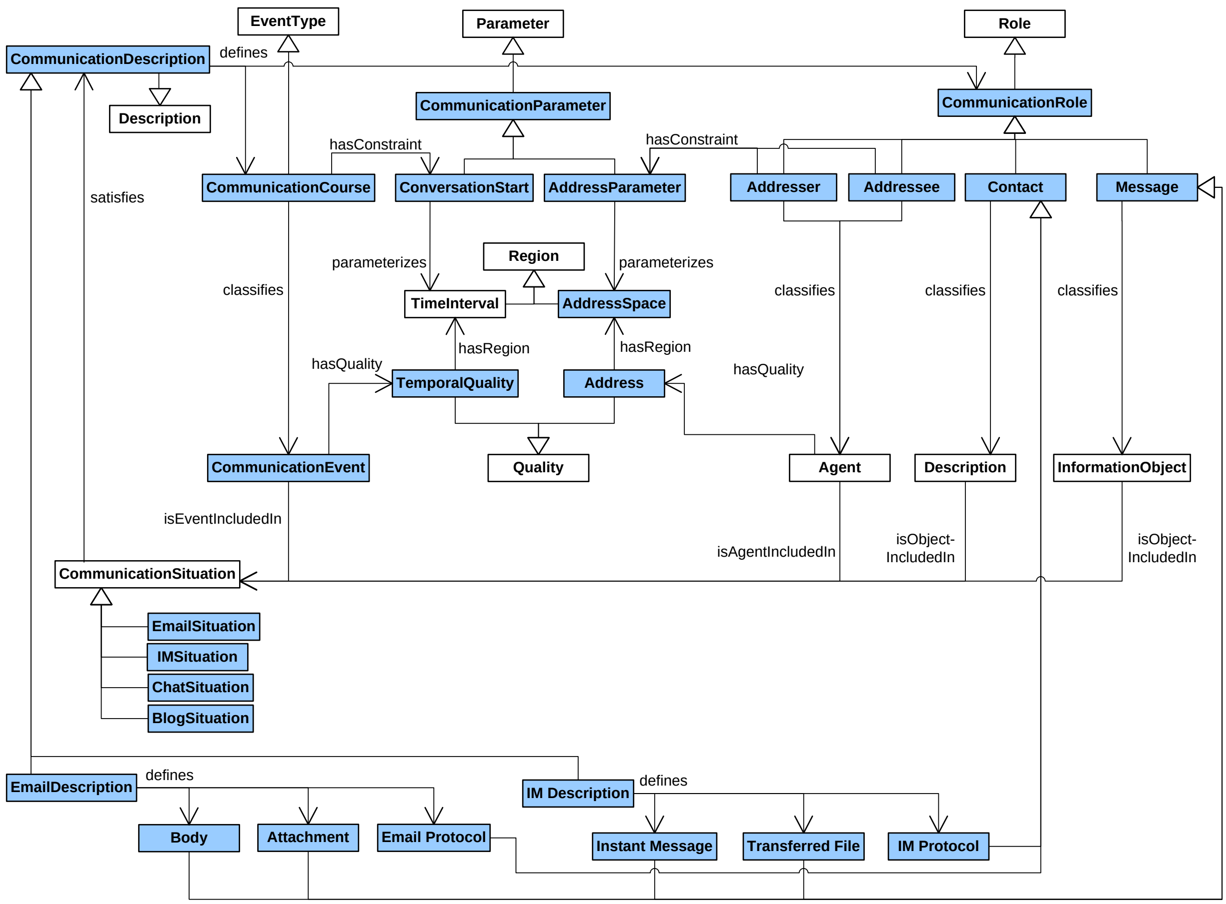


Fig. 25. Communication pattern

in the `CommunicationSituation`. Precisely, it requires that `Agents` have an `Address` quality that is contained in the `AddressSpace parameterized` by the `AddressParameter`. Similarly, `CommunicationEvents` are required to have a `TemporalQuality` that is within the `TimeInterval parameterizedBy` the parameter `ConversationStart`. The quality then expresses temporal information about the time at which a conversation took place. `Agents` have the role of addressers and addressees, while descriptions of protocols play the role of `Contact`, e.g., a description of the simple mail transfer protocol plays the role of contact for email communication. The `Message` role is played by an `InformationObject` that is separated from its realization that is accessible by a `DigitalRealization`.

Next to the communication roles, the `CommunicationDescription` also defines a `CommunicationCourse` which classifies a `CommunicationEvent`. A `CommunicationEvent` can be a conversation that spans several exchanges of messages. While the `CommunicationDescription` represents the abstract model of Jakobson, it subsumes more specific descriptions of particular communication modes, e.g., representing email communication or instant message communication. Descriptions of such communication modes define more specific roles than the abstract `CommunicationDescription`, e.g., the `EmailDescription` defines the role of an `Attachment` and of an `EmailProtocol` as more specific roles of `Message` and `Contact`.

Applying the communication pattern for representing communication information, it can be reused in further contexts, classified according to any classification scheme and linked with further information. Moreover, disadvantages from the separation of different communication modes are alleviated as messages of any mode are represented by a common conceptual model for communication that results in a unified view onto communication. As an example, a conversation where `a-1` sends an email to b-1 while b-1 replies to a-1 by an instant message would be represented by the instantiation of two CommunicationDescriptions, one EmailDescription and one IMDescription. These would define roles such as Addresser and Addressee that are played by the instances of the class Agent representing a-1 and b-1. A CommunicationCourse will also be defined that classifies a CommunicationEvent that has all the players of the roles defined by the particular communication description as participants. That same CommunicationEvent is also classifiedBy a communication course in the second description, i.e., the IMDescription. Within the communication course of the instant messaging, the IMDescription has the players of the roles as further participants. Thus, conversational threads can be represented in spite of the use of different communication modes that are based on different addressing mechanisms, communication protocols, and message serializations.

*5.3.2. Task Pattern*

The task pattern of X-COSIMO has been designed by extending the plan ontology (Gangemi et al., 2005). The refinement is conducted alongside the requirements imposed by the functionalities towards process support as defined in the X-Media project (Uren et al., 2006). Similar to the communication pattern, the DnS pattern is employed to represent a task-specific context onto the information, agents, and actions that take part in the execution of processes. The task pattern is depicted in Figure 26.

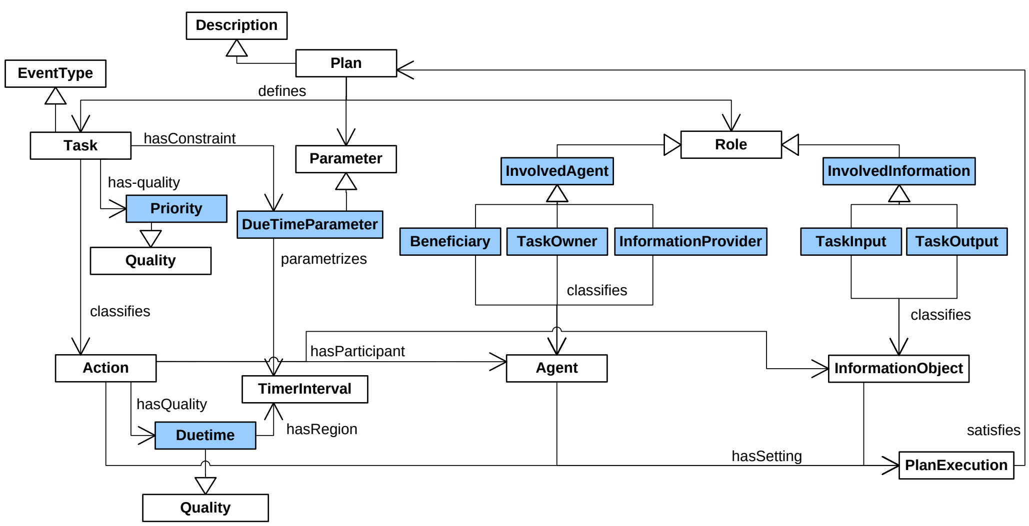


Fig. 26. Task pattern

The notion of a task is expressed in terms of DnS as a Task that classifies Actions. Actions represent activities that take place within the execution of a Plan. For instance, the event that is classified as a communication course with respect to communication may be regarded as an action with respect to a particular task and process. Different information objects and agents take part in an action. This is represented using the hasParticipant property. As tasks are commonly prioritized, the quality Priority associates a priority value to a task. Actions are associated to a Duetime, modeled as a quality of an action. The value of a due time is parametrized by a DueTimeParameter that is a constrained for a task.

Several notions of agents are also defined by the pattern, namely Beneficiary, TaskOwner, and InformationProvider. These are roles of agents and distinguish between different attitudes and stati of agents that take part in a process. The role of a TaskOwner classifies agents that take responsibility for a task, e.g., by ensuring timely or correct execution of a task. The role of an InformationProvider characterizes contributing participation as given, e.g., by domain experts that are consulted to gain further inside knowledge required to execute a task successfully. A similar notion of some agent providing information could also be represented by an additional sub-task associated to the information provider. However, we have introduced the role to make the status of such agents explicit. The role of a `Beneficiary` classifies agents that do not directly participate in actions, but anyway have an attitude towards a task. To give an example, a `Beneficiary` can be a customer that orders a product. While the customer does not participate in the manufacture of the product, the customer might expect certain quality levels, delivery times, and thus influences the process of manufacture. Next to agents, information can be involved in the execution of a process, either as input or as output of a process. Subclasses of `InvolvedInformation` classify information objects accordingly. The concepts like `Task` and `Beneficiary` are always defined by an instance of a `Plan`. A `Plan` is subclass of a `Description` that `defines` agent-driven-roles, tasks, and involved information. The corresponding `Situation` is given by the class `PlanExecution` that provides a setting for the classified activities, agents, and information.

## 6. Comparison with Core Ontology Properties

We have motivated the need for representing and sharing complex, structured knowledge by the use of core ontologies at the example of a socio-technical system for emergency response in Section 2. The use and design of our core ontologies Event-Model-F, COMM, and X-COSIMO has been shown in Section 3. These core ontologies have been designed, implemented, and evaluated over the last years in different European projects such as the WeKnowIt project, the X-Media project, and the network of excellence K-Space. Based on our and reported experiences in designing core ontologies, we have introduced our design approach for core ontologies in Section 4. The concrete design of our core ontologies following this approach has been described in Section 5.

In this section, we compare our Event-Model-F, COMM, and X-COSIMO with the properties of core ontologies introduced in Section 1. These properties are axiomatization and formal precision, modularity, extensibility, reuseability, and separation of concerns. We show examples of how the properties have been put into practice and summarize with a discussion from designing and applying our core ontologies in European projects.

### *6.1. Axiomatization and Formal Precision*

Core ontologies shall provide a high degree of axiomatization and formal precision. This is required to establish a common understanding in a particular field and to ensure interoperability through machine accessible semantics. The axiomatization and formal precision of our core ontologies is achieved by basing the Event-Model-F, COMM, and X-COSIMO on the foundational ontology DOLCE. By this, our core ontologies inherit the rich axiomatization and formal precision of the foundational ontology. In addition, further axioms have been added to the core ontologies to precisely define the structured knowledge in the particular field they cover. These axioms have been specified using description logics (Baader et al., 2003). When precisely specifying the semantics of ontology design patterns with description logics one encounters the question of how many axioms are needed in order to sufficiently define the semantics of a pattern. This is of particular interest for patterns that base on the Descriptions and Situations (DnS) pattern. The `Description` defines a number of concepts that determine the roles of the entities in the pattern. These roles make up the structured knowledge represented by the pattern. For example, the axiomatization of the `EventCausalityDescription` of the causality pattern described in Section 5.1.3 defines the three roles `Cause`, `Effect`, and `Justification` as shown below (a full axiomatization can be found in (Scherp et al., 2009c)).

$$EventCausalityDescription \sqsubseteq \forall defines.(Cause \sqcup Effect \sqcup Justification)$$

No other roles except `Cause`, `Effect`, and `Justification` are allowed in the causality pattern. The other patterns of the Event-Model-F are defined following the same approach. Also the patterns of X-COSIMO are specified very tightly using description logics. For example, the communication pattern described in Section 5.3.1 defines that there can only be a `CommunicationCourse`, ConversationStart, AddressParameter, Adresser, Adressee, Contact, and Message. Other roles are excluded. This tight axiomatization is required for a semantically precise specification and use of the patterns. For example, an ontology engineer might want to use the pattern in a context in which it is not designed for. He might want to add some concepts to it that do not fit the design of, e.g., the causality pattern like introducing aspects of correlation to it. By a high amount of axiomatization, we reduce the risk of wrongly applying the ontology design patterns.

On the other hand, it might be interesting in the future to add some additional information to the patterns like provenance information, i.e., meta-information about the method and parameter setting by which the concrete pattern instantiations have been created. For example, the annotation pattern of COMM described in Section 5.2.3 could be extended to provide information about the source and confidence of the annotations. Thus, the current axiomatization of COMM does not define a restricted set of roles in its patterns. For example, the semantic annotation pattern allows to integrate further roles to the already defined ones, namely AnnotatedDataRole and SemanticLabelRole.

It is a principal design decision whether the patterns of a core ontology are closed with respect to additional roles such as the Event-Model-F and X-COSIMO or if roles can be added to the patterns like in COMM. There is no universal design solution to the question of how many axioms are needed in order to sufficiently define the semantics of a pattern. However, the discussion shows that the amount of axiomatization of a pattern is dependent on how much one wants to exactly define and restrict the pattern's context of use and how open it shall be for future extensions. In the case of the Event-Model-F and X-COSIMO, a very tight specification of the patterns has been chosen in order to best capture the theories that are represented by the structure of the patterns like the philosophical question of causality in the case of the causality pattern in the Event-Model-F and Jakobson's communication theory in the case of the communication pattern of X-COSIMO. With the annotation pattern of COMM, only a loose specification has been conducted in order to allow extensions of COMM towards various specific features of existing metadata models and metadata formats. Please note that the main purpose of the rich axiomatization of the core ontology is consistency checking at design time and may be abstracted into efficiently processable languages later (see discussion in Section 4.2.1).

The axiomatization of our core ontologies is available online from our website and can be investigated in detail in the ontologies themselves: http://west.uni-koblenz.de/Research/ontologies/. In addition, an extensive description of the axiomatization of the Event-Model-F can be found in (Scherp et al., 2009c).

### 6.2. Modularity

Foundational ontologies model the very basic and general concepts and relations (Borgo and Masolo, 2009; Oberle, 2006) that make up our world. They are hard to learn and to understand how to apply them to build a core ontology or domain ontology. Thus, there is a high learning curve when applying a foundational ontology. To alleviate this situation, a good foundational ontology like DOLCE already makes use of ontology design patterns such as DnS and IO.

Based on the very generic ontology design patterns provided by DOLCE, we have carefully chosen the scope of our core ontology design patterns. The patterns provide a solution to a recurrent ontology modeling problem (Gangemi and Presutti, 2009) such as participation of objects in events in the Event-Model-F, the annotation of media data in COMM, and the communication between agents in X-COSIMO. Thus, the scope for each pattern is chosen that its functionality can be encapsulated and provided by a distinct service. They are neither too generic such that they cannot be applied or specialized in a specific domain nor are they too specific such that they are limited in their use (Presutti and Gangemi, 2008). Finally, it is also important to mention that there is no overlap between the scope of the patterns. Thus, there are no redundant patterns. By defining the different ontology design patterns in a well-chosen scope, we provide a modularization of our core ontologies.

### *6.3. Extensibility*

Modularity is a prerequisite to allow core ontologies to be extensible towards new developments and functional requirements that arise (cf. adaptability in (Vrandečić, 2009)). Throughout the lifecycle of ontologies, they need to be extended and modified in order to adapt them to new requirements that may have emerged over time. With their pattern-oriented design, the core ontologies Event-Model-F, COMM, and X-COSIMO are well prepared to handle such new requirements and extensions. With respect to extensibility of our core ontologies, several examples and different work can be mentioned. We describe how we have dealt with newly arising requirements and how we have extended our core ontologies towards them.

An example for extending existing ontology design patterns is from the X-Media project. Here, new requirements for media annotation have been raised. While the COMM ontology supports the description of image data and video data as shown in Section 5.2, the scenarios in X-Media additionally required to describe annotations of textual and numerical data, e.g., textual data given in the open document format. The annotation of portions of the data, e.g., a text passage or a subset of numerical data, also required to describe the decomposition of the data into segments. The extension of COMM towards these media types has resulted in the Ontology for Knowledge Acquisition (OAK) (Iria, 2009). Concepts defined in COMM have been specialized in OAK in order to adapt the core ontology to the requirements on the integration of knowledge acquisition tools. Analogue to the concepts `Image` and `Video`, the concept `Text` has been introduced in COMM. Like `Image` and `Video`, it is a subclass of `Media` that realizes `TextData`. `TextData` is a specialization of the COMM concept `MultimediaData`, which represents information objects. `MultimediaData` already has subclasses such as `ImageData` and `VideoData` as shown in Section 5.2.2. Furthermore, new locators for textual data have been defined, analogue to the locators already existing for image data and video data. For the description of the annotation and decomposition of the newly added media types, existing COMM patterns have been reused.

With respect to adding additional ontology design patterns to an existing core ontology, we consider the Event-Model-F. In the initial design of our Event-Model-F, we did not foresee the documentation pattern described in Section 5.1.4. Our technical report documents this initial design of the Event-Model-F where the documentation pattern is missing (Scherp et al., 2009b). The documentation pattern allows to associate documentary evidence with an event, thus it constitutes the missing link that connects the Event-Model-F with COMM. The documentation pattern of the Event-Model-F described in Section 5.1.4 and used for modeling the emergency response scenario in Section 3 has only been added later (Scherp et al., 2009a). To extend the initial version of the Event-Model-F by an additional pattern and integrating this additional pattern with the core ontology the following steps have been performed: In a first step, the existing patterns of the Event-Model-F have been checked to which extend they already provide the functionality that shall be added with the new pattern. As there is no overlap, the documentation pattern has been designed and carefully aligned to DOLCE+DnS Ultralight in a second step. Thus, we did not need to change or adapt the existing patterns. By using DOLCE as common basis, the documentation pattern could directly be used with the Event-Model-F.

Finally, our core ontologies are extended in a research project conducted in collaboration with the arts department of the University of Koblenz-Landau. The researchers of the arts department are interested in the history of art and conducting iconographic research of scientific art pieces. They analyze the depiction of scientific concepts such as the atom model and track how the depiction has changed over time due to scientific inventions and discoveries. A web-based system has been developed for the collaborative discussion of the scientific art pieces and representing different interpretations on them depending on the point of views the researchers might have. In this work, a pattern-based domain ontology for the history of scientific art has been developed. This domain ontology is based on DOLCE+DnS Ultralight and uses and extends multiple patterns provided by the Event-Model-F and COMM. For example, the life of a person that is relevant in the history of science such as researchers and inventors are modeled as specialization of the Event-Model-F composition pattern. To model a person's life, the pattern defines specific events such as birth, death, events in education, inventions, and scientific discoveries. In addition, the interpretation of scientific art pieces is not always clear and is typically subject to discussion. Here, the Event-Model-F interpretation pattern is applied to model the different opinions the experts have. The metadata information about the digital media data representing the scientific art pieces in the system are modeled using COMM. Here, no changes or specializations of the core ontology have been conducted. Thus, COMM is reused without modifications. There are also new patterns introduced for the domain ontology. These patterns model additional aspects that are not covered by the scope of our three core ontologies. For example, an ontology design pattern is added for modeling the change of qualities of events and objects over time like the name of a researcher that has changed due to marriage. The patterns of the domain ontology are axiomatized using description logics (Baader et al., 2003) to provide a formal semantics of the structural knowledge captured by the system.

### 6.4. *Reuseability*

Closely related to extensibility is the property of reuseability. Being modularly defined, a core ontology supports reuse of its ontology design patterns in various different domains. At the same time it still guarantees formal precision of the overall knowledge it represents. Reuseability of our core ontologies is considered in two kinds: First, the structured knowledge defined by core ontologies can be reused (and extended) towards newly arising requirements and specific domains. The structured knowledge of our three core ontologies Event-Model-F, COMM, and X-COSIMO have already been reused in different applications and domains as the discussion in Section 6.3 on extensibility shows. This reuse of the existing ontology design patterns has been conducted without any changes to the particular patterns. For example, the modeling of the emergency response scenario in Section 3 demonstrates the use of various ontology design patterns such as the decomposition pattern of COMM, the task pattern of X-COSIMO, and the participation pattern of the Event-Model-F without the need to modify them. In addition, the existing ontology design patterns have been reused by specializing concepts defined in the patterns. This is the case with the example of COMM to add support for textual data, which has been added with the development of OAK. For example, the concept `TextData` has been introduced as specialization of `MultimediaData` and the concept `Text` as subclass of `Media`.

Second, besides the reuse of the structured knowledge defined in the core ontologies, they are also able to incorporate existing domain knowledge. They make use of that domain knowledge rather than requiring to remodel it. The emergency response scenario modeled in Section 3 demonstrates the reuse of domain ontologies within our core ontologies. This reuse of domain knowledge takes place with the roles defined in the patterns. For example, modeling the participation of the citizen Paul in the event of a snapped power pole in Figure 2 reuses a domain ontology for roles in emergency response. Thus, in the concrete example the citizen `paul-1` is classified by the concept `CitizenRole` taken from the domain ontology. Other roles in this ontology are the `FiremanRole` that `paul-1` can play in other situations as he is also a professional fireman.

Another possibility for reusing domain knowledge takes place with the real-world entities that are described by the patterns, i.e., classified by the roles. For example, the composition pattern of the Event-Model-F depicted in Figure 13 shows a representation of a larger flooding event that is composed of smaller incident events. Here, the `Composite` role classifies the individual `flooding-1` that is of event type `Flooding`. The `Flooding` event is taken from a domain ontology for emergency response that is provided by one of the partners in the WeKnowIt project. Similar to this example is the use of the domain-specific concept `PowerPole` to represent the `power-pole-1` in the participation pattern at the beginning of Section 3. The `PowerPole` is defined in an external domain ontology and reused here.

Besides the demonstrated reuses of domain knowledge in the Event-Model-F, domain ontologies are also reused in COMM and X-COSIMO. In the modeling example of Section 3, an existing repository containing operators at the emergency hotline is reused to annotate the segments of the audio recordings. In Figure 8, Rita is identified in the call segment `audio-segment-1` and thus `rita-1` is associated with the segment. The repository of operators at the hotline is not aligned with DUL. It uses the concept `Person` to represent the individual operators. Thus, the `SemanticLabelRole` in Figure 8 classifies the concept `Person` instead of DUL's `NaturalPerson`. Although `rita-1` is of concept `Person`, we can reuse the individual as label in the semantic annotation pattern. Since `rita-1` plays only in the context of a semantic annotation the `SemanticAnnotationRole`, we do not cause any unwanted inferences or infer any additional class memberships. Besides such closed data repositories like the repository containing the operators at the hotline, one can also make use of linked open data such DBpedia (Bizer et al., 2009).

In the task pattern of X-COSIMO depicted in Figure 12 a domain-specific ontology for tasks in emergency response is applied. Such an ontology allows to distinguish between different types of `Tasks` such as to confirm a situation, inspect a problem, contacting people, and others. The example in Figure 12 shows an `InspectionTask` that is to be carried out by the floating liaison officer `marie-1` to inspect a snapped power pole.

### 6.5. *Separation of Concerns*

The structured knowledge of our core ontologies is clearly separated from the domain-specific knowledge that is reused. As we have shown in the previous section, our core ontologies allow to apply different kinds of domain-specific knowledge and domain-specific ontologies. By the separation of concerns, the domain-specific knowledge is integrated and reused without affecting the core ontology itself. This is achieved by using the DnS ontology design pattern for representing the structured knowledge of the concrete field captured by our core ontologies (see Section 4.2.5). The roles that are defined by the `Description` specify the structure of the ontology design pattern, i.e., represent the structured knowledge of the core ontology. The domain knowledge is only referred to by the roles defined in the pattern. In addition, the individuals and the concepts of the entities classified by the roles may come from a domain ontology. By using the DnS pattern, this domain specific knowledge does not affect the structured knowledge of the core ontology. Independent whether domain ontologies are used for the roles or entities in the patterns, the roles that are defined by the `Description` remain. In addition, with respect to the separation of concerns it does not make a difference if the used domain ontology is aligned to DOLCE or not. The separation of concerns in our core ontologies Event-Model-F, COMM, and X-COSIMO is shown in Section 6.4 at the examples of reusing domain ontologies. The examples demonstrate the reuse of existing domain ontologies for the roles defined in the DnS-based patterns of the Event-Model-F, COMM, and X-COSIMO and the entities classified by the roles in these patterns.

The separation of concerns is nicely visualized in the example of the participation pattern in Section 5.1.1. The `EventParticipationDescription defines` the different roles required in an event participation situation, namely a `DescribedEvent` and a `Participant`. As shown in Figure 15, subconcepts of the `Participant` role can be defined in some domain-specific ontology. For example, an emergency response ontology that defines different emergency roles like the `CitizenRole` and `FiremanRole`. Besides reusing domain ontologies for specifying the roles in the participation pattern, also the classified entities such as the described event and its participating objects in the participation pattern can be provided by some domain ontology. Such a domain ontology can provide both individuals as well as concepts that are reused with the pattern. Independent of the domain ontologies actually used, the participation pattern always foresees two roles of exactly one `DescribedEvent` role and at least one `Participant` role.

### 6.6. *Concluding Discussion*

The core ontologies we have developed, namely COMM, X-COSIMO, and Event-Model-F are employed in larger European projects as briefly described in Section 2. Partly, they are in use since 2007. Thus, it is first of all important to mention that since their creation no major changes on the principal design of our core ontologies were necessary. In the course of applying and extending our core ontologies, no additional patterns had to be introduced. Thus, no essential patterns or functionality is missing in our core ontologies to serve the purpose they have been designed for. This means, the scope of our core ontologies has been well chosen. For example, COMM is designed to provide support for annotations of media. In the X-Media project the requirement of new media types arised that so far were not supported by COMM, namely text and numerical data (see Section 6.3). However, no new patterns needed to be added in COMM for providing support for text and numerical data. Rather, the existing patterns for decomposition and annotation could be reused. For the interpretation of scientific art pieces, the modeling of peoples' life with events was implemented as specialization of the Event-Model-F composition pattern. Thus, none of the additions required the modification of existing patterns of the core ontology. In addition, none of them introduced inconsistencies or violated the intentional use of any of the patterns defined by our core ontologies. In cases where new ontology design patterns have been created such as the change of qualities over time, we can state that these resulted from requirements that are not within the scope of our core ontologies presented here. Rather, such newly introduced patterns complement our three core ontologies and are already designed such that they are interoperable with them. This shows that our core ontologies are of some stability and allow for reuse of the structured knowledge defined in the ontology.

Designing core ontologies based on an existing foundational ontology is a cumbersome and tedious task. It requires excellent knowledge of the foundational ontology used as modeling basis. In addition, it requires to precisely align the core ontology with the foundational ontology. Core ontology designers also need in depth knowledge about the field that is to be covered by the core ontology they design. Here, an extensive analysis of the related work is necessary, i.e., the analysis of existing models and ontologies in the field as well as existing applications from different domains in the field. Based on this analysis, the functional requirements to the core ontology can be derived. A core ontology covers a specific field that is an abstraction of different concrete domains (see Section 4.1). Thus, it requires a reasonable level of abstraction when designing it.

Core ontologies are not only of theoretical interest. They are used for representing knowledge in different, concrete application scenarios. For example, the Event-Model-F is used for the creation and sharing of emergency response information in SemaPlorer++ (Scherp et al., 2010) and managing emergency response log files in the WeKnowIt log-merger application (Papadopoulos et al., 2010). The COMM is used for representing multimedia metadata in the K-Space Annotation Tool (Saathoff et al., 2008) and X-COSIMO is applied for personal information management in the semantic desktop applications COSIMail and COSIFile (Franz et al., 2009, 2007). In order to make practical use of core ontologies in concrete application scenarios, appropriate application programming interfaces (APIs) are essential to access the ontological knowledge (see Section 4.2.1). Our experience with developing such APIs for ontologies is that they should not have a fixed programming interface. Although the design of a core ontology should be very stable with respect to its use in different domains, the API should be redesigned based on the requirements of the concrete application domain in which it is used. By this, we can enable a much more efficient access to the knowledge modeled and represented with the core ontology. To address this requirement, we have developed a model driven approach for generating APIs to access ontological knowledge from software applications (Scheglmann et al., 2010).

## 7. Beautiful Aspects of Our Core Ontologies

We have presented a design approach for developing core ontologies in Section 4.2 that meets all requirements posed in Section 1. These requirements are axiomatization and formal precision, modularity, extensibility, reuseability, and separation of concerns. We have exemplified core ontologies developed according to this design approach in Section 5 and have highlighted how they fulfill the requirements in Section 6. Core ontologies that have been created following this design approach and fulfilling the requirements above can be easily combined with each other to describe highly complex knowledge structures although they have been designed for different purposes and for different fields. Concrete examples demonstrating how our core ontologies are combined in the emergency response scenario are presented in Section 7.1. In Section 7.2, we compare our approach with the use of other, more domain-oriented ontologies to model complex scenarios and argue for the benefits of our core ontologies.

## 7.1. Combining Core Ontologies

The core ontologies Event-Model-F, COMM, and X-COSIMO presented in Section 5 have been designed for different purposes and to provide support for different fields. However, by the nature of their design they can be flexibly combined. The ability for a flexible combination is essential for facilitating the representation of complex, structured knowledge that spans across several different domains. As an example of the application of our core ontologies, we have illustrated how they support the representation of complex, structured knowledge in an emergency response system in Section 3. However, they have also been applied successfully to support complex systems in the aviation and automotive industry. In the following, we refer to the concrete emergency scenario from Section 3 to highlight the beauty in the integration and smooth interplay of our core ontologies. It becomes visible when different patterns of different core ontologies share common real-world entities, namely events and objects.

For example, the audio recording at the emergency hotline illustrates these aspects. The information that an audio recording took place during a phone call is used within the documentation pattern of the Event-Model-F as shown in Figure 3. It is represented by `audio-rec-1` and is associated to the event of `paul-1`'s call to the emergency hotline to report about a snapped power pole. The same individual `audio-rec-1` representing the audio recording is used within the digital data pattern of COMM as depicted in Figure 5. The `audio-rec-1` of concept `AudioData` is a specialization of `DigitalData`, which is defined in the digital data pattern. Again, the same `audio-rec-1` is used in the communication pattern and task pattern of X-COSIMO for modeling a message and task description send between two emergency response officers as shown in Figures 11 and 12.

Another example demonstrating the interplay of the Event-Model-F and COMM is the representation of the citizen `paul-1`. The individual `paul-1` is used in the participation pattern of the Event-Model-F as shown in Figure 2. It represents the call of `paul-1` to the emergency response hotline to report about a snapped power pole. During the call a recording of the conversation between `paul-1` and the officer at the hotline answering the call takes place. In order to provide efficient access to the different parts of the conversation, the recorded call is segmented and annotated with the speakers' names. Here, the participation pattern of the Event-Model-F interplays with the semantic annotation pattern of COMM as shown in Figure 7. Each segment of the audio call is annotated with a semantic label representing the person speaking. In our example, the `audio-segment-2` is annotated with the individual `paul-1`.

The interplay between the Event-Model-F and X-COSIMO is nicely shown in Figure 10. It depicts the attachment of Paul's interpretation `inter-sit-1` for the power outage event to the task pattern. The interpretation is modeled using the Event-Model-F, whereas the task pattern is defined in X-COSIMO. The task pattern is depicted in Figure 12 and takes Paul's interpretation for the power outage as `TaskInput`. The example shows that not only some individuals can be shared between our different core ontology patterns. It demonstrates that even entire instantiations of the patterns can be integrated and reused in other patterns. In our example, an instantiation of the event interpretation pattern of the Event-Model-F identified by the individual `inter-sit-1` is reused in the task pattern of X-COSIMO.

## 7.2. Comparison with other Ontologies

Besides our core ontologies, there exist also other ontologies for modeling events such as the Event Ontology[16] and the Linking Open Descriptions of Events (LODE)[17] ontology. For representing multimedia metadata we find ontologies like the W3C Ontology for Media Resource[18]. Finally, there are also ontologies supporting knowledge management such as the Personal Information Model (PIMO)[19]. These ontologies are suitable for being applied in the specific domain they have been designed for. Some of these ontologies are designed within the context of a specific project such as the Event Ontology in the context

---

[16] `http://motools.sourceforge.net/event/event.html`
[17] `http://linkedevents.org/ontology/`
[18] `http://www.w3.org/TR/2010/WD-mediaont-10-20100309/`
[19] `http://www.semanticdesktop.org/ontologies/pimo/`

of a music management system and PIMO as part of the semantic desktop NEPOMUK. Other ontologies like LODE and the Ontology for Media Resource aim at integrating existing ontologies in their domain. The complexity of these ontologies is often low, i.e., they typically consist only of a few concepts and properties. Some of these ontologies exist already for a while like the Event Ontology that has been first released in 2004 and is already well known in the community.

These ontologies may be a good choice for modeling the domain of emergency response as they provide for representing events, multimedia metadata, and personal information. However, these ontologies are not designed to collaborate and to be combined in a complex, socio-technical system such as emergency response. Although some of these ontologies reuse existing ontologies like the Event Ontology reuses the Friend-of-a-Friend[20] vocabulary it remains unclear how they should be connected with each other. For example, the Event Ontology does not know about PIMO and the Ontology for Media Resource. LODE and the Ontology of Media Resource allow for integrating other ontologies, but this is limited to ontologies in their specific domain. They also do not know how to handle PIMO and other ontologies in further domains.

If one wants to combine these ontologies in a concrete information system they have to be harmonized and integrated a posteriori. However, this can become a big challenge as the ontologies typically do not follow a systematic development approach and are often semantically ambiguous. For example, the semantics of the `factor` and `product` properties of the Event Ontology is quite vague and no axiomatization of the causal relation expressed with these properties is provided. The property `location` of the Ontology of Media Resource is provided to express where a specific media asset has been recorded. However, this property may cause confusion as it is often mixed up with the location the content is about. For example, US president Barack Obama may make some statements in a video captured at the White House in Washington about events happening in the area of the Middle East. Matching the concepts and properties of the different ontologies also becomes a challenging task. For example, it is unclear which properties of the Ontology of Media Resources that are related to *agents* such as `contributor`, `creator`, `publisher`, and `targetAudience` should be aligned with the `involvedAgent` property of LODE to describe events. In the worst case, the ontologies that shall be combined need to be redesigned from scratch.

In contrast, our core ontologies provide a precise semantics by a rich axiomatization and follow a common design approach. By the nature of their design, our core ontologies allow to be combined in a complex scenario such as emergency response although the core ontologies have been designed for different fields. This is achieved by using a common foundational ontology. Thus, the beauty of our core ontologies lies in their ability to be combined within complex scenarios. This is achieved just by the way how we design the core ontologies. We belief that it is better to design ontologies such that they are prepared for being combined with others like our core ontologies rather than integrating ontologies a posteriori in order to use them together in a concrete information system.

## 8. Conclusions

We have presented the design and use of three core ontologies, the Event-Model-F, COMM, and X-COSIMO. These core ontologies are characterized by a high degree of axiomatization and formal precision. This is achieved by basing on the common foundational ontology DOLCE+DnS Ultralight. Our core ontologies follow a pattern-oriented design approach, which make them modular and extensible. In addition, they support reuse of the structured knowledge defined within the core ontologies as well as support reuse of existing domain ontologies. By applying the Descriptions and Situations pattern of DOLCE+DnS Ultralight they support the separation of concerns, i.e., clearly separating the structured knowledge defined in the core ontology from the domain-specific aspects. Due to these characteristics, our core ontologies allow for both formally conceptualize their particular fields and to be flexibly combined to cover the needs

[20] `http://www.foaf-project.org/`

of concrete, complex application domains. Thus, from our perspective they are to be considered *beautiful* ontologies.

With the design of our core ontologies, we contribute to the overall discussion and methodologies for improving ontology design and ontology quality such as OntoClean (Guarino and Welty, 2002a) and DILIGENT (Pinto et al., 2009) and bring the field further forward from an art to an engineering discipline (d'Aquin and Gangemi, 2011). As such, our core ontologies are of interest to the ontology engineering community. We assume that in the future many further beautiful core ontologies will emerge and that the need to combine multiple core ontologies to support the information exchange in complex applications will increase. With beautiful core ontologies such as the Event-Model-F, X-COSIMO, and COMM, we can fulfill the requirement of future information systems to model and exchange very complex, structured information in a heterogeneous setting of different client applications and systems in use. The core ontologies Event-Model-F, COMM, and X-COSIMO have been created in the Web Ontology Language[21] (OWL) and axiomatized using description logics (Baader et al., 2003). The core ontologies as well as tool support we have developed are available online from our website: `http://west.uni-koblenz.de/Research/ontologies/`.

## Acknowledgements

This research has been co-funded by the EU in FP7 in the WeKnowIt project (215453) and in FP6 in the X-Media project (026978).

## References

Arndt, R., Troncy, R., Staab, S., and Hardman, L. (2009). *Handbook on Ontologies*, chapter COMM: A Core Ontology for Multimedia Annotation. Springer, 2nd edition.

Arndt, R., Troncy, R., Staab, S., Hardman, L., and Vacura, M. (2007). COMM: Designing a well-founded multimedia ontology for the web. In *ISWC*. Springer.

Ashburner, M. (2000). Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25:25–29.

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Babitski, G., Bergweiler, S., Grebner, O., Oberle, D., Paulheim, H., and Probst, F. (2011). Soknos - using semantic technologies in disaster management software. In Antoniou, G., Grobelnik, M., Simperl, E. P. B., Parsia, B., Plexousakis, D., Leenheer, P. D., and Pan, J. Z., editors, *The Semanic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29 - June 2, 2011, Proceedings, Part II*, volume 6644 of *Lecture Notes in Computer Science*, pages 183–197. Springer.

Babitski, G., Probst, F., Hoffmann, J., and Oberle, D. (2009). Ontology design for information integration in disaster management. In Fischer, S., Maehle, E., and Reischuk, R., editors, *GI Jahrestagung*, volume 154 of *LNI*, pages 3120–3134. GI.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009). DBpedia: A crystallization point for the web of data. *Web Semant.*, 7(3):154–165.

Bloehdorn, S., Petridis, K., Saathoff, C., Simou, N., Tzouvaras, V., Avrithis, Y., Handschuh, S., Kompatsiaris, Y., Staab, S., and Strintzis, M. G. (2005). Semantic annotation of images and videos for multimedia analysis. In *European Semantic Web Conference*. Springer.

Borgo, S., Carrara, M., Vermaas, P. E., and Garbacz, P. (2006). Behavior of a technical artifact: An ontological perspective in engineering. In Bennett, B. and Fellbaum, C., editors, *Formal Ontology in Information Systems, Proceedings of the Fourth International Conference, FOIS 2006, Baltimore, Maryland, USA, November 9-11, 2006*, volume 150 of *Frontiers in Artificial Intelligence and Applications*, pages 214–225. IOS Press.

Borgo, S. and Masolo, C. (2009). *Handbook on Ontologies*, chapter Foundational choices in DOLCE. Springer, 2nd edition.

Cimiano, P. and Reyle, U. (2006). Towards foundational semantics - ontological semantics revisited. In Bennett, B. and Fellbaum, C., editors, *Formal Ontology in Information Systems, Proceedings of the Fourth International Conference, FOIS 2006, Baltimore, Maryland, USA, November 9-11, 2006*, volume 150 of *Frontiers in Artificial Intelligence and Applications*, pages 51–62. IOS Press.

Cote, R., Rothwell, D., Palotay, J., Beckett, R., and Brochu, L. (1993). The systematized nomenclature of human and veterinary medicine. Technical report, SNOMED International, Northfield, IL: College of American Pathologists.

d'Aquin, M. and Gangemi, A. (2011). Is there beauty in ontologies? *Applied Ontology*, (Special Issue on Beautiful Core Ontologies).

[21] `http://www.w3.org/2004/OWL/`

Doerr, M., Ore, C.-E., and Stead, S. (2007). The CIDOC conceptual reference model: a new standard for knowledge sharing. In *Conceptual modeling*, pages 51–56. Australian Computer Society, Inc.

Ekin, A., Tekalp, A. M., and Mehrotra, R. (2004). Integrated semantic-syntactic video modeling for search and browsing. *IEEE Transactions on Multimedia*, 6(6):839–851.

Euzenat, J. and Shvaiko, P. (2007). *Ontology matching*, chapter Classifications of ontology matching techniques. Springer.

Francois, A. R. J., Nevatia, R., Hobbs, J., and Bolles, R. C. (2005). VERL: An ontology framework for representing and annotating video events. *IEEE MultiMedia*, 12(4).

Franz, T., Scherp, A., and Staab, S. (2009). Are semantic desktops better?: summative evaluation comparing a semantic against a conventional desktop. In Gil, Y. and Noy, N. F., editors, *Proceedings of the 5th International Conference on Knowledge Capture (K-CAP 2009), September 1-4, 2009, Redondo Beach, California, USA*, pages 1–8. ACM.

Franz, T., Staab, S., and Arndt, R. (2007). The X-COSIM integration framework for a seamless semantic desktop. In *Knowledge capture*, pages 143–150. ACM.

Gamma, E. (2007). *Design patterns : elements of reusable object oriented software*. Addison-Wesley.

Gangemi, A. (2005). Ontology design patterns for semantic web content. In Gil, Y., Motta, E., Benjamins, V. R., and Musen, M. A., editors, *International Semantic Web Conference*, volume 3729 of *Lecture Notes in Computer Science*, pages 262–276. Springer.

Gangemi, A. (2007). *Trends in Legal Knowledge: The Semantic Web and the Regulation of Electronic Social Systems*, chapter Design Patterns for Legal Ontology Construction. European Press Academic Publishing.

Gangemi, A. (2008). Norms and plans as unification criteria for social collectives. *Autonomous Agents and Multi-Agent Systems*, 17(1):70–112.

Gangemi, A., Borgo, S., Catenacci, C., and Lehmann, J. (2005). Task taxonomies for knowledge content. Deliverable 07 of the EU FP6 project Metokis, http://metokis.salzburgresearch.at/.

Gangemi, A., Fisseha, F., Keizer, J., Lehmann, J., Liang, A., Pettman, I., Sini, M., and Taconet, M. (2004). A core ontology of fishery and its use in the fishery ontology service project. In Borgo, A. G. S., editor, *EKAW*04 Workshop on Core Ontologies in Ontology Engineering*, Northamptonshire, UK. CEUR Proceedings Vol. 118.

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., and Schneider, L. (2002). Sweetening Ontologies with DOLCE. In *Knowledge Engineering and Knowledge Management*, pages 166–181. Springer.

Gangemi, A. and Mika, P. (2003). Understanding the semantic web through descriptions and situations. In *CoopIS/DOA/ODBASE*, pages 689–706. Springer.

Gangemi, A. and Presutti, V. (2009). *Handbook on Ontologies*, chapter Ontology Design Patterns. Springer, 2nd edition.

Garcia, R. and Celma, O. (2005). Semantic integration and retrieval of multimedia metadata. In *Knowledge Markup and Semantic Annotation Workshop*, Galway, Ireland.

Guarino, N. and Welty, C. (2002a). Evaluating ontological decisions with OntoClean. *Commun. ACM*, 45(2):61–65.

Guarino, N. and Welty, C. A. (2000a). A formal ontology of properties. In Dieng, R. and Corby, O., editors, *Knowledge Acquisition, Modeling and Management, 12th International Conference, EKAW 2000, Juan-les-Pins, France, October 2-6, 2000, Proceedings*, volume 1937 of *Lecture Notes in Computer Science*, pages 97–112. Springer.

Guarino, N. and Welty, C. A. (2000b). Ontological analysis of taxonomic relationships. In *ER*, pages 210–224.

Guarino, N. and Welty, C. A. (2002b). Evaluating ontological decisions with ontoclean. *Commun. ACM*, 45(2):61–65.

Herre, H., Heller, B., Burek, P., Hoehndorf, R., Loebe, F., and Michalek, H. (2006). General formal ontology (GFO): A foundational ontology integrating objects and processes. part I: Basic principles (version 1.0). Onto-Med Report 8, Research Group Ontologies in Medicine (Onto-Med), University of Leipzig.

Hunter, J. (2005). *Multimedia Content and the Semantic Web: Standards, Methods and Tools*, chapter Adding Multimedia to the Semantic Web: Building an MPEG-7 ontology, pages 261–283. John Wiley & Sons.

IPTC (2008). EventML. http://iptc.org/.

Iria, J. (2009). A core ontology for knowledge acquisition. In *European Semantic Web Conference*, pages 233–247.

Isaac, A. and Troncy, R. (2004). Designing and using an audio-visual description core ontology. In *Workshop on Core Ontologies in Ontology Engineering, CEUR Proceedings Vol. 118*, Whittlebury Hall, Northamptonshire, UK.

Itkonen, E. (1983). *Causality in Linguistic Theory*. Indiana Univ. Press.

Jain, R. (2008). EventWeb: Developing a human-centered computing system. *Computer*, 41(2):42–50.

Jakobson, R. (1960). *Style in Language*, chapter Closing statement: Linguistics and Poetics, pages 350– 377. Cambridge, Mass., MIT Press.

Kundu, S., Itkin, M., Gervais, D., Krishnamurthy, V., Wallace, M., Cardella, J., Rubin, D., and Langlotz, C. (2009). The IR radlex project: An interventional radiology lexicon. *Vascular and Interventional Radiology*, 20(4):433–435.

Lagoze, C. and Hunter, J. (2001). The ABC ontology and model. In *Proc. of the Int. Conf. on Dublin Coreand Metadata Applications*, pages 160–176. National Institute of Informatics, Tokyo, Japan.

Lenat, D. B., Guha, R. V., Pittman, K., Pratt, D., and Shepherd, M. (1990). Cyc: toward programs with common sense. *Commun. ACM*, 33(8):30–49.

Lombard, L. (1986). *Events: A metaphysical study*. Routledge & Kegan Paul.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., and Oltramari, A. (2003). WonderWeb deliverable D18 ontology library; IST WonderWeb project.

Mika, P., Oberle, D., Gangemi, A., and Sabou, M. (2004). Foundations for service ontologies: aligning OWL-S to DOLCE. In *World Wide Web*. ACM.

Minsky, M. (1974). A framework for representing knowledge. Technical report, Cambridge, MA, USA.

MPEG-7 (2001). Multimedia content description interface. Technical report, Standard No. ISO/IEC n15938.

Mueller, E. T. (2008). *Handbook of Knowledge Representation*, chapter Event Calculus. Elsevier.

Niles, I. and Pease, A. (2001). Towards a standard upper ontology. In *Formal Ontology in Information Systems*, pages 2–9, New York, NY, USA. ACM.

Oberle, D. (2006). *Semantic Management of Middleware*. Springer.
Oberle, D., Ankolekar, A., Hitzler, P., Cimiano, P., Sintek, M., Kiesel, M., Mougouie, B., Baumann, S., Vembu, S., and Romanelli, M. (2007). DOLCE ergo SUMO: On foundational & domain models in the SmartWebIntegrated Ontology. *Web Semant.*, 5(3):156–174.
Oberle, D., Grimm, S., and Staab, S. (2009a). *Handbook on Ontologies*, chapter An Ontology for Software. Springer, 2nd edition.
Oberle, D., Guarino, N., and Staab, S. (2009b). *Handbook on Ontologies*, chapter What is an Ontology? Springer, 2nd edition.
Oberle, D., Lamparter, S., Grimm, S., Vrandečić, D., Staab, S., and Gangemi, A. (2006). Towards ontologies for formalizing modularization and communication in large software systems. *Appl. Ontol.*, 1(2):163–202.
Papadopoulos, S., Scherp, A., Ireson, N., Tsampoulatidis, I., and Kompatsiaris, Y. (2010). Using event representation and semantic enrichment for managing and reviewing emergency incident logs. In *Proceedings of the 2nd ACM international workshop on Events in multimedia*, EiMM '10, pages 41–46, New York, NY, USA. ACM.
Pinto, H. S., Tempich, C., and Staab, S. (2009). *Handbook on Ontologies*, chapter Ontology Engineering and Evolution in a Distributed World Using DILIGENT. Springer, 2nd edition.
Presutti, V. and Gangemi, A. (2008). Content ontology design patterns as practical building blocks for web ontologies. In *Proceedings of the 27th International Conference on Conceptual Modeling*, ER '08, pages 128–141, Berlin, Heidelberg. Springer-Verlag.
Quinton, A. (1979). Objects and events. *Mind*, 88(350):197–214.
Raimond, Y. and Abdallah, S. (2007). The event ontology. http://motools.sf.net/event.
Rector, A. L. and Horrocks, I. R. (1997). Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Workshop on Ontological Engineering*. AAAI Press.
Ren, Y., Pan, J. Z., and Zhao, Y. (2010). Soundness preserving approximation for TBox reasoning. In Fox, M. and Poole, D., editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press.
Rosse, C. and Mejino, J. L. V. (2003). A reference ontology for bioinformatics: The foundational model of anatomy. *Biomedical Informatics*, 36:478–500.
Saathoff, C., Schenk, S., and Scherp, A. (2008). KAT: The K-Space annotation tool. In *SAMT 2008, Demo Session Proceedings*.
Scheglmann, S., Scherp, A., and Staab, S. (2010). Model-driven generation of APIs for OWL-based ontologies. Technical Report 07/2010, Institut WeST, UniversitÃd't Koblenz-Landau, Germany.
Scherp, A., Franz, T., Saathoff, C., and Staab, S. (2009a). F-a model of events based on the foundational ontology DOLCE+DnS Ultralight. In *Accepted for publication at the International Conference on Knowledge Capturing (K-CAP), Redondo Beach, CA, USA*.
Scherp, A., Franz, T., Saathoff, C., and Staab, S. (2009b). A model of events based on a foundatiional ontology. Technical Report 2/2009, Arbeitsberichte aus dem Fachbereich Informatik, University Koblenz-Landau, Koblenz, Germany. ISSN 1864-0850.
Scherp, A., Franz, T., Saathoff, C., and Staab, S. (2010). A core ontology on events for representing occurrences in the real world. *Multimedia Tools and Applications*, pages 1–39. 10.1007/s11042-010-0667-z.
Scherp, A., Papadopoulos, S., Kritikos, A., Schwagereit, F., Saathoff, C., Franz, T., Schmeiss, D., Staab, S., Schenk, S., and Bonifacio, M. (2009c). D5.2.1 prototypical knowledge management methodology (intermediate report). Technical report, WeKnowIt Project. http://www.weknowit.eu/sites/default/files/D5.2.1.pdf.
Schneider, L. (2003). Designing foundational ontologies - the object-centered high-level reference ontology OCHRE as a case study. In *Conceptual Modeling - ER 2003, 22nd International Conference on Conceptual Modeling*. Springer.
Staab, S., Scherp, A., Arndt, R., Troncy, R., Grzegorzek, M., Saathoff, C., Schenk, S., and Hardman, L. (2008). Semantic multimedia. In *Reasoning Web; San Servolo Island, Venice, Italy*. Springer.
Tsinaraki, C., Polydoros, P., and Christodoulakis, S. (2004). Integration of OWL ontologies in MPEG-7 and TVAnytime compliant semantic indexing. *Advanced Information Systems Engineering, 16th International Conference, CAiSE04*, pages 398–413.
Uren, V., Petrelli, D., Cimiano, P., Franz, T., Lanfranchi, V., Gilardoni, L., Busse, J., Slavazza, P., Dadzie, A.-S., Ciravegna, F., Lei, Y., Rabus, D., and Thanh, T. D. (2006). X-media deliverable 4.1: Specification of knowledge sharing systems. Technical report, X-Media Consortium.
Vrandečić, D. (2009). *Handbook on Ontologies*, chapter Ontology Evaluation. Springer, 2nd edition.
Wang, X., Mamadgi, S., Thekdi, A., Kelliher, A., and Sundaram, H. (2007). Eventory – an event based media repository. In *Semantic Computing*, pages 95–104. IEEE.
Welty, C., Fikes, R., and Makarios, S. (2006). A reusable ontology for fluents in owl. In *Proceeding of the 2006 conference on Formal Ontology in Information Systems*, pages 226–236. IOS Press.