CASAM:

Computer-Aided Semantic Annotation of Multimedia

European Seventh Framework Programme

Project Acronym: **CASAM**

Project Title: **Computer-Aided Semantic Annotation of Multimedia**

Contract Number: **FP7-217061**

Deliverable Number: **D6.2**

Title of the Deliverable: **Domain ontology**

Task/WP related to the Deliverable: **WP6 Modelling of the target domain**

Type: **Deliverable**

Distribution:

Author(s): **TUHH**

Partner(s) Contributing: **TUHH**

Versioning and Contribution History			
Version	**Date**	**Modification reason**	**Modified by**
01	16/01/09	First Completed Version	TUHH
02	25/03/09	Final Version	TUHH
03	07/04/09	Final Version After Internal Review	TUHH

Executive Summary

This report describes the first version of the CASAM domain ontology, and documents the design decisions behind it. In particular, the report identifies patterns which will guide the development of future parts of the ontology such that media information extraction can be formalized using ontologies (and probabilistic ontologies in the future).

Contents

1 Introduction

The goals of the CASAM[1] project are to develop a system for computer-aided semantic annotation of multimedia documents. A representation of an environmental domain has been chosen as the application scenario. It is a large domain covering many aspects such as environmental pollution, conferences, catastrophes, hazards and conservation attempts. In order to represent general knowledge of this domain, a domain ontology has to be developed.

The CASAM domain ontology will evolve as long as the main technical facilities of the CASAM systems are worked out, especially the analysis facilities called "Knowledge-Driven Multimedia Analysis" (KDMA), detecting objects within the multimedia document and the interpretation facilities called "Reasoning for Multimedia Interpretation" (RMI), trying to find explanations for the observed objects. The agreed set of all concept and property names is called *signature* and will be the basis for all further versions of the ontology.

For the design of the ontology we stick to a paradigm called "Grounded Ontology Design", where the design is based on the data that can be detected within the documents, the information that can be logically inferred form the data, and the retrieval scenarios that will be actually used in the application. In this deliverable, we also present general patterns required for all subsequent versions of the CASAM domain ontology as well as a proposal for the first version with several examples.

We did not start with "upper model" ontologies including SUMO (Suggested Upper Merged Ontology)[2], DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering)[3] and Cyc[4], because these ontologies consist of a huge amount of concepts which we assume are not required for the CASAM project. Besides the amount of unnecessary concepts, most "upper model" ontologies have a high expressivity. Both facts have an impact on the performance of reasoning. However, special concepts stored in these or other ontologies can easily added to later versions of the ontology if useful. To fulfill the requirements of a real-time application, we propose to apply the rather less expressive description logic language $\mathcal{ALH}_f(\mathcal{D})$.

In Chapter 2, the syntax and semantics of the description logic language $\mathcal{ALH}_f(\mathcal{D})$ is explained as well as the specification of retrieval problems in this language. Chapter 3 introduces design patterns for ontologies in the context of CASAM in general. All specifications in this chapter provide an ontology scheme and are relevant for all subsequent versions of the ontology. Afterwards, according to these patterns, in Chapter 4 the first version of the environmental domain ontology is presented which mainly evolved from initial non-speech audio analysis and text analysis results as well as from asking DW, LUSA and EJC for desired retrieval concepts. Then, in Chapter 5 the process of hypothesizing interpretations is explained. We present our proposal for abduction rules relevant for CASAM. In Chapter 6, the expressivity of $\mathcal{ALH}_f(\mathcal{D})$ with respect to CASAM is justified.

2 Description Logics

Description Logics (DLs) [Baader et al., 2003] in most cases are decidable fragments of first-order logic. Some of these logics are very expressive, though, and provide well understood means to establish ontologies. Therefore they are also used as representation languages for the Semantic Web [Baader et al., 2005].

The vocabulary of description logic languages consists of *concepts*, *roles* and *constants*. Concepts denote sets of objects, roles binary relations between objects and constants specific objects.

[1]CASAM = **C**omputer-**A**ided **S**emantic **A**nnotation of **M**ultimedia
[2]http://www.ontologyportal.org/
[3]http://www.loa-cnr.it/DOLCE.html
[4]http://www.cyc.com/

2.1 Syntax and Semantics of $\mathcal{ALH}_f(\mathcal{D})$

One of the main targets of the CASAM project is to significantly speed up the task of manual annotation, since in this project real-time issues are important. Thus, we assume that a less expressive representation language should be applied to facilitate fast computations. We decided to represent the domain knowledge with the DL $\mathcal{ALH}_f(\mathcal{D})$ (attributive concept language with role hierarchies, functional roles and concrete domains).

A DL *signature* is a tupel $\mathcal{S} = (\mathbf{A}, \mathbf{R})$, where $\mathbf{A}$ and $\mathbf{R}$ are the sets of all atomic concepts and all atomic roles, respectively. Further, $\mathbf{FR}$ is the set of *functional roles* with $\mathbf{FR} \subseteq \mathbf{R}$ and $\mathbf{AT}$ the set of *concrete domain attributes* with $\mathbf{AT} \subseteq \mathbf{FR}$. Concrete domain attributes (e.g. *hasValue*) provide a means to relate objects to concrete domains such as integers, strings, etc. A specification of the introduction of concrete domains to DLs is e.g. given in [Baader and Hanschke, 1991]. For brevity, in the following, we refer to $\mathcal{ALH}_f$ and consider concrete domains ($\mathcal{D}$) only for examples and querying purposes.

Let $A \in \mathbf{A}$ be an atomic concept (e.g. *AssociationActivist*, *Conference* or *Journalist*) and $R \in \mathbf{R}$ an atomic role (e.g. *interviews*). Then arbitrary $\mathcal{ALH}_f$ concept descriptions C or D are inductively defined with

$$C,\, D \;\longrightarrow \top \mid \bot \mid A \mid \neg A \mid C \sqcap D \mid \forall R.C \mid \exists R.\top,$$

i.e., besides atomic concepts A and their negations $\neg A$, $\mathcal{ALH}_f$-concept descriptions are composed of the logical constants $\top$ and $\bot$ (anything resp. nothing), concept conjunctions ($C \sqcap D$), value restrictions ($\forall R.C$) and limited existential restrictions ($\exists R.\top$). Every string contained in a concept description C or D itself being a concept is called a *subconcept*.

The semantics of $\mathcal{ALH}_f$ is defined with interpretations $\mathcal{I} = (\triangle^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\triangle^{\mathcal{I}}$ is a non-empty set of all objects considered in $\mathcal{I}$ (called the domain of $\mathcal{I}$) and $\cdot^{\mathcal{I}}$ is an interpretation function which maps constants to objects of the domain ($a^{\mathcal{I}} \in \triangle^{\mathcal{I}}$), atomic concepts to subsets of the domain ($A^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}}$) and roles to subsets of the cartesian product of the domain ($R^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}} \times \triangle^{\mathcal{I}}$). The interpretation of arbitrary $\mathcal{ALH}_f$ concept descriptions then is defined by extending $\cdot^{\mathcal{I}}$ to all $\mathcal{ALH}_f$ concept constructors as follows:

$$\begin{array}{lcl}
\top^{\mathcal{I}} & = & \triangle^{\mathcal{I}} \\
\bot^{\mathcal{I}} & = & \emptyset \\
(\neg A)^{\mathcal{I}} & = & \triangle^{\mathcal{I}} \setminus A^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} & = & \{u \in \triangle^{\mathcal{I}} \mid (\forall v)\,[(u,v) \in R^{\mathcal{I}} \rightarrow v \in C^{\mathcal{I}}]\} \\
(\exists R.\top)^{\mathcal{I}} & = & \{u \in \triangle^{\mathcal{I}} \mid (\exists v)\,[(u,v) \in R^{\mathcal{I}}]\}
\end{array}$$

The interpretation of $\top$, $\bot$, $\neg A$ and $C \sqcap D$ is defined apparantly with the set of all objects in the domain of $\mathcal{I}$, the empty set, the difference of the sets $\triangle^{\mathcal{I}}$ and $A^{\mathcal{I}}$ and the intersection of the sets $C^{\mathcal{I}}$ and $D^{\mathcal{I}}$, respectively. Value restrictions $\forall R.C$ are the most characteristic constructor of DLs. They are interpreted in $\mathcal{I}$ with the set of all objects u being in relation R only to objects which are in the extension of C. Finally, limited existential restrictions $\exists R.\top$ are interpreted with the set of all objects u being indeed in relation R to at least one arbitrary object v.

In order to relate concepts and roles to each other and in order to assign constants to concepts and roles, a knowledge base has to be specified. An *$\mathcal{ALH}_f$-knowledge base* $\Sigma_{\mathcal{S}} = (\mathcal{T}, \mathcal{A})$ with respect to a signature $\mathcal{S}$ is comprised of a terminological component $\mathcal{T}$ called *TBox* and an assertional component $\mathcal{A}$ called *ABox* (in the following Σ is used as an abbreviation for $\Sigma_{\mathcal{S}}$). Let C and D be $\mathcal{ALH}_f$ concept descriptions, R and S roles and a and b constants denoting objects of the chosen domain. Then $\mathcal{T}$ consists of a set of axioms

$$C \sqsubseteq D, \quad C \equiv D, \quad R \sqsubseteq S, \quad R \equiv S$$

called concept inclusions, concept definitions, role inclusions and role definitions, respectively,[5] and $\mathcal{A}$ is a set of concept assertions $C(a)$ and role assertions $R(a,b)$. In addition, concept inclusions of the form $\top \sqsubseteq (\leq 1\, R)$ are allowed in $\mathcal{T}$ to specify functional roles (indicated by $_f$).

We now present the *satisfiability* of axioms and assertions of an $\mathcal{ALH}_f$-knowledge base Σ in an interpretation $\mathcal{I}$. A concept inclusion $C \sqsubseteq D$ (concept definition $C \equiv D$) is satisfied in $\mathcal{I}$, if

[5] Note that $C \equiv D$ *iff* $C \sqsubseteq D$ *and* $D \sqsubseteq C$ and $R \equiv S$ *iff* $R \sqsubseteq S$ *and* $S \sqsubseteq R$

$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (resp. $C^{\mathcal{I}} = D^{\mathcal{I}}$) and a role inclusion $R \sqsubseteq S$ (role definition $R \equiv S$), if $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$ (resp. $R^{\mathcal{I}} = S^{\mathcal{I}}$). Concept inclusions specifying that a role is functional, $\top \sqsubseteq (\leq 1\, R)$ ("all objects are related over R to at most one object"), are satisfied in $\mathcal{I}$, if

$$\forall a \in \triangle^{\mathcal{I}} : \forall b, c\, [(a, b) \in R^{\mathcal{I}} \wedge (a, c) \in R^{\mathcal{I}} \rightarrow b = c].$$

Finally, assertions $C(a)$ and $R(a, b)$ are satisfied in $\mathcal{I}$, if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ resp. $(a, b)^{\mathcal{I}} \in R^{\mathcal{I}}$. If an interpretation $\mathcal{I}$ satisfies all axioms of $\mathcal{T}$ resp. $\mathcal{A}$ it is called a *model* of $\mathcal{T}$ resp. $\mathcal{A}$. If it satisfies both $\mathcal{T}$ and $\mathcal{A}$ it is called a model of Σ. Finally, if there is a model of Σ (resp. $\mathcal{T}$ resp. $\mathcal{A}$), then Σ (resp. $\mathcal{T}$ resp. $\mathcal{A}$) is called satisfiable.

2.2 Logical Entailment - Querying with Description Logics

A DL knowledge base Σ *logically entails* an assertional axiom α (symbolically $\Sigma \models \alpha$), if α is satisfied in all models of Σ. In other words, if an assertional axiom is logically entailed by a knowledge base, then it is proved to be true with respect to the assumed knowledge. Processes deciding whether $\Sigma \models \alpha$ holds or not are called (ABox-)Reasoning Services in the following. If $\alpha = C(a)$, these services are called *instance-checks* and if $\alpha = R(a, b)$ they are called *relation-checks.*

Example 2.1 Consider the TBox $\mathcal{T} = \{AssociationActivist \sqsubseteq Person\}$ consisting only of axioms with atomic concepts and the ABox $\mathcal{A} = \{AssociationActivist(act_1)\}$, i.e. the knowledge base is

$$\Sigma = (\{AssociationActivist \sqsubseteq Person\}, \{AssociationActivist(act_1)\}).$$

$Person(act_1)$ is satisfied in all models of Σ, i.e., a correct instance-check deciding whether this concept assertion is logically entailed by Σ will be successful.

Furthermore, the *retrieval* inference problem is to find the set of constants a (mentioned in an ABox) that can be proved to be instances of a certain concept C, i.e. $\{a \mid \Sigma \models C(a)\}$. In addition to this service more expressive query languages are required in practical applications: *Conjunctive Queries CQ* are defined as

$$CQ := \{(Y_1, ..., Y_n) \mid atom_1, ..., atom_m\}.$$

Such queries consist of a query body $atom_1, ..., atom_m$ specifying the retrieval conditions and a query head $(Y_1, ..., Y_n)$ specifying the format of the answer. A query body atom in general is a concept expression $C(X_i)$, a role expression $R(X_i, X_j)$ or a same-as expression $X_i = X_j$. In addition, it is possible to specify concrete domain atoms $CD(X_i)$.[6] CD is an expression $(P\ At\ d)$, where P is a concrete domain predicate (e.g =), $At \in \mathbf{AT}$ is a concrete domain attribute (e.g. *hasValue*) and d is a concrete domain value (e.g. "João Falcato"). It is required that variables Y_i appearing in the query head must also appear in the query body. To derive an answer, these variables are bound to possible ABox constants and instantiated query atoms are checked for logical entailment with respect to Σ: Let $vars(CQ)$ be the set of all variables of the query. If φ is a substitution assigning variables to constants for all $v \in vars(CQ)$, then a solution to CQ is a set

$$\{(\varphi(Y_1), ..., \varphi(Y_n)) \mid \Sigma \models \varphi(atom_1), ..., \Sigma \models \varphi(atom_n)\},$$

where $\varphi(C(X_i)) := C(\varphi(X_i))$, $\varphi(R(X_i, X_j)) := R(\varphi(X_i), \varphi(X_j))$, $\varphi(X_i = X_j) := \varphi(X_i) = \varphi(X_j)$ and $\varphi(CD(X_i)) := CD(\varphi(X_i))$.

In CASAM, there are two main concerns for logic based querying:

- The ability of RMI to derive new assertions in order to add new annotations to multimedia documents
- The ability to retrieve annotated multimedia documents

[6] In expressive query languages like the new Racer Query Language (nRQL) [Wessel and Möller, 2005] there are a lot of additional query atoms possible

The derivation of new assertions is performed by applying abduction rules (see Chapter 5) in order to hypothesize multimedia annotations. Regarding the second point, since journalists querying for annotated documents will only be interested in documents about a specific event, person etc. and not in the individual names of these events, persons etc., each multimedia document will be retrieved, if there is at least one binding of the head of the corresponding query (independent of the form of this binding). For example, a user could be interested in videos about forest clearing. Then all multimedia documents annotated with at least one assertion $ForestClearing(f)$ with an arbitrary constant f will be retrieved.

The specification of axioms for an ontology due to a given signature (together with a given set of assertions) determines which assertions can be proved or not and is therefore of great importance for the derivation of multimedia interpretations and answers to queries from users. In Chapter 3, a general structure of the CASAM domain ontology is proposed and in Chapter 4, the first version of the environmental domain ontology is presented.

3 Specification of ontologies for CASAM multimedia analysis and interpretation

In the following, an ontology framework is presented. At first, we introduce a DL signature as a basis for the specification of a TBox $\mathcal{T}$. Afterwards, in Section 3.2 we specify ontology design patterns specially suited for the CASAM analysis and interpretation. These patterns further restrict the language of the ontology to provide even more runtime performance as well as a general structure for all upcoming versions of the ontology.

3.1 Ontology represented by a DL signature

Definition 3.1 *A DL **signature** of a knowledge base* Σ *is a tupel* $\mathcal{S} = (\boldsymbol{A}, \boldsymbol{R})$ *consisting of the set* $\boldsymbol{A} = \{A_1, ...A_n\}$ *of all concept names (atomic concepts)* A_i *and the set* $\boldsymbol{R} = \{R_1, ..., R_m\}$ *of all role names (atomic roles)* R_i *to be considered for* Σ *with* $\boldsymbol{A} \cap \boldsymbol{R} = \emptyset$.

In other words, the signature of Σ is the set of all atomic concepts A_i and roles R_i agreed on to be specified in Σ. In the context of CASAM, it can be seen as a contract of all project partners, determining which things and properties of the chosen domain can be analyzed from KDMA, recognized by HCI[7], retrieved from the journalists and therefore involved in reasoning procedures of RMI. Since the signature considered for CASAM provides the information which concepts and relations are assumed to exist in a chosen domain, it represents the CASAM *domain ontology.*

3.2 Design Patterns for the CASAM domain ontology

As already stated in Section 2, an $\mathcal{ALH}_f$ TBox $\mathcal{T}$ consists of a set of axioms

$$C \sqsubseteq D, \quad C \equiv D, \quad R \sqsubseteq S, \quad R \equiv S$$

to restrict that in any case the set of objects C is a subset (resp. is equal to) the set of objects D, where C and D denote arbitrary $\mathcal{ALH}_f$-concept descriptions (analogously for atomic roles R, S). These restrictions allow for inferences to obtain implicit knowledge. In order to provide even more effectivity as well as a general structure for all upcoming versions of the ontology, we further restrict $\mathcal{T}$ to specific patterns, i.e., we propose not to specify concept inclusions $C \sqsubseteq D$ and concept definitions $C \equiv D$ with arbitrary $\mathcal{ALH}_f$-concept descriptions C, D.

3.2.1 Atomic Specification / Generalisation

At first, a requirement for each ontology is to provide the modeling of more specific resp. more general objects. This is done with inclusion axioms of the form

$$A_1 \sqsubseteq A_2 \tag{1}$$

[7]Human-Computer-Interaction component of CASAM

to state that if sth. is an instance of A_1 in any case it is also an instance of A_2 (A_1 is subsumed by A_2), where A_1 and A_2 are atomic concepts. For example, suppose the ontology $\mathcal{T}$ contains the axiom $\mathit{ForestFire} \sqsubseteq \mathit{Incident}$ and that it is known that the individual f is an instance of a forest fire event, i.e. there is an assertion $\mathit{ForestFire}(f)$ specified in an ABox. Then the ontology is able to infer $\mathit{Incident}(f)$. Reasoning is applied in a transitive way, i.e. if e.g. $\{A_1 \sqsubseteq A_2, A_2 \sqsubseteq A_3\} \subseteq \mathcal{T}$ and $A_1(obj_1) \in \mathcal{A}$, then it is possible to infer $A_3(obj_1)$. Note that "multiple inheritance" restrictions such as $A_1 \sqsubseteq A_2 \sqcap A_3$ (A_1 is subsumed by both A_2 and A_3) can also be expressed with the inclusions $A_1 \sqsubseteq A_2$ and $A_1 \sqsubseteq A_3$. Pairs of concept names A_i, A_j both subsumed by the same concept name, but not subsuming each other, are called *concept siblings*. By specifying axioms (1), a concept hierarchy (also called taxonomy) is constructed.

Analogously to (1), it should be possible to specify role hierarchies

$$R \sqsubseteq S \tag{2}$$

For example, the atomic role *interviews* can be modeled more specific than *talksTo*.

3.2.2 Disjointness

In CASAM, it is required to be able to model that atomic concepts A_1 and A_2 are mutually disjoint. In description logics, this is not the default assumption and has to be specified with the concept inclusions

$$A_1 \sqsubseteq \neg A_2 \tag{3}$$

For example, with respect to the environmental domain it holds that factories in any case are not bridges, represented with $\mathit{Factory} \sqsubseteq \neg \mathit{Bridge}$, and that planting events in any case are not conference events, represented with $\mathit{Planting} \sqsubseteq \neg \mathit{Conference}$. Usually, all concept siblings are assumed to be mutually disjoint. A lot of disjointness axioms constrain models of the ontology to avoid irrelevant interpretations built up by abduction rules (cf. Chapter 5).

3.2.3 Domain and Range Restrictions of Roles

In $\mathcal{ALH}_f$, it is possible to constrain the domain and the range of an atomic role R with the inclusion axioms

$$\exists R.\top \sqsubseteq A\,, \qquad \top \sqsubseteq \forall R.A \tag{4}$$

Consider the role *interviews*. This role could in either case be supposed to relate instances of *Journalist* always with instances of *Person*. In other words, *Journalist* is referred to as the **domain** of the role *interviews* and *Person* is referred to as the **range** of this role.[8] If now, e.g., there is the information that $\mathit{interviews}(\mathit{obj_1}, \mathit{obj_2})$ holds then $\mathit{Journalist}(\mathit{obj_1})$ as well as $\mathit{Person}(\mathit{obj_2})$ are satisfied in all models of the corresponding knowledge base.

3.2.4 Functional Roles

In order to define that a role R is assumed to be **functional**, the inclusion axiom

$$\top \sqsubseteq (\leq 1\, R) \tag{5}$$

has to be specified, meaning that R is always relating to at most one object. Examples of functional roles could be *hasDate* or possibly *hasLocation*, while e.g. *interviews* is assumed to be non-functional.

3.2.5 Value Restrictions

In contrast to global range restrictions $\top \sqsubseteq \forall R.A$, local range restrictions (or value restrictions) in $\mathcal{ALH}_f$ are specified with

$$A_1 \sqsubseteq \forall R.A_2 \tag{6}$$

[8]Note that an axiom $\mathit{Journalist} \sqsubseteq \mathit{Person}$ is independent of these restrictions.

meaning that all instances of A_1 are only related over R to instances of A_2. Local range restrictions are intended to deliver implicit knowledge from more generic to more specific (i.e. from more abstract to more concrete) atomic concepts. For example, if $\mathit{FireFight} \sqsubseteq \forall \mathit{hasParticipant}.\mathit{FireFighter}$ is specified in the ontology and, in addition, $\mathit{FireFight}(f_1)$ as well as $\mathit{hasParticipant}(f_1, obj_2)$ is given, then $\mathit{FireFighter}(obj_2)$ can be inferred.

3.2.6 Definitions including value restrictions

Except for the disjointness axioms, for each atomic concept $A \in \mathbf{A}$ consider the set

$$A_{incl} = \{A \sqsubseteq B, A \sqsubseteq \forall R_1.B_1, \cdots, A \sqsubseteq \forall R_m.B_m\}$$

of all inclusion axioms with A on the left side explicitly specified in the ontology,[9] where the concepts A and B as well as all concepts $B_j, j = 1, ..., m$ are atomic. If all roles $R_j, j = 1, ..., m$ are assumed to be functional (i.e. are modeled in the ontology with $\top \sqsubseteq (\leq 1\, R_j)$), then it is possible to specify a definition axiom

$$A \equiv B \sqcap \forall R_1.B_1 \sqcap \cdots \sqcap \forall R_m.B_m \quad (7)$$

instead of all the inclusion axioms of A_{incl}. Since $C \equiv D$ holds if and only if $C \sqsubseteq D$ and $D \sqsubseteq C$ holds, with axioms (7) it is possible to reason from left to right *and* from right to left. Under the open world assumption, usually there is no proof for value restrictions $\forall R_j.B_j$. However, all roles R_j are assumed to be functional: If $R_j(obj_1, obj_2)$ is known, then obj_2 is the only filler of R_j such that there is a proof for $\forall R_j.B_j$ if in addition $B_j(obj_2)$ holds.

4 Environmental Domain Ontology - First Version

In this chapter we present a first version of the ontology representing the environmental domain of CASAM. It has to be mentioned that for the exact definition of the signature (the list of all concept names and role names), the RMI component requires all analysis results of the KDMA component. Since we are still in a rather early project phase it is not possible to receive the whole results of the analysis process. So far, we only received a list of keywords with non-speech audio analysis results as well as a list of concept names from initial text analysis experiments regarding LUSA video material from the first content set. Unfortunately, no relations have been detected at the moment and we suggested a short list of role names which could be a part of the result of multimedia analysis. Following to this, the signature $\mathcal{S} = (\mathbf{A}, \mathbf{R})$ of the first version of the ontology will be rather unprecise.

We left out several available ontologies including SUMO (Suggested Upper Merged Ontology), DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) and Cyc. As mentioned in the introduction, these ontologies consist of a huge amount of concepts which we assume mainly are not required for in the CASAM project and therefore will only slow down the reasoning process. Furthermore, these ontologies are specified in more expressive languages than $\mathcal{ALH}_f$.

Instead we propose to apply **grounded ontology design**, i.e. to represent only concepts and roles of the domain that are assumed to be relevant for the multimedia interpretation of RMI.

In order to get a basic concept structure of the ontology, we propose that all atomic concepts A of the signature are either

- events
- physical things or
- entities

represented with the generic concepts *Event*, *PhysicalThing* and *Entity*, respectively. Events state what a multimedia document is about with respect to the chosen domain. They are supposed to be of most interest for the end users of CASAM and are expected to be mainly queried for. Therefore,

[9] For each A there is exactly one atomic concept B explicitly specified to subsume A. If A is one of the most general concepts, it is subsumed by $\top$.

RMI will apply abduction rules to hypothesize instances of specific events. Apart from manual annotations, physical things as well as entities initially are expected to result from multimedia analysis techniques. However, there will also be events delivered from multimedia analysis as well as queries regarding physical things and entitites. We assume events, physical things and entities to be mutually disjoint:

$$\begin{array}{lcl} Event & \sqsubseteq & \neg PhysicalThing \\ Event & \sqsubseteq & \neg Entity \\ PhysicalThing & \sqsubseteq & \neg Entity \end{array}$$

4.1 Events

Events are supposed to be mainly the search keywords of the journalists. In the following we present the most general events of the environmental domain:

- *EnvironmentalProcess*: This concept indicates the changes in the environment including *CarbonCycle*, *ClimateChange*, *Condensation* and *Evaporation*.
- *Hazard*: This event conceptualizes the dangers and threats to the surrounding environment including e.g. *ForestClearing*, *DamBuilding* or different types of *Pollution*.
- *HumanActivity*: This concept indicates approaches to protect the environment, including *Planting*, *CleanUp* and *Protest*. In this respect, it also contains the concept *PoliticalEvent* and more specific concepts such as *Conference*, *Demonstration*, etc.
- *Incident*: This event indicates catastrophes and disasters which are originated from the nature. They can take people's lives and lead to financial damage. Examples of these events are *Earthquake*, *Flood* and *Drought*.
- *Interview*: This concept refers to the interviews regarding the environment.
- *TechnologicalProcess*: This concept refers to the technological approaches to protect the environment including e.g. *Recycling*, *EnergySaving*, *Mining*.

All these events are concept siblings and therefore usually assumed to be disjoint (but there are exceptions).

4.2 Physical Things

Physical things are atomic concepts which can be touched or felt and exist as an object. Some examples for the most general physical things are *Animal*, *Person*, *Resource* and *Technology*.

According to the video material of LUSA, KDMA provides the detection of Non-Speech Audio Kewords. These keywords are repesented by sound concepts in the ontology such as "ChairShifting", "Laughing" or "CarEngineSound". We assume that based on the appearance of these sounds other physical things or events can be interpreted.

4.3 Entities

Entities are concepts that provide more information about events or physical things. Examples of entities assumed to be useful for the CASAM environmental domain are *Name*, *Date*, *Degree*, *Location*. A lot of entities can be subsumed by value restrictions $\forall hasValue.D$, where *hasValue* is a generic concrete domain attribute and D is a concrete domain such as string, integer, etc. For example, in the first version of the ontology *Name* is assumed to be related over *hasValue* only to the concrete domain of strings.

4.4 Example

In this section, a concrete example including concepts, roles and corresponding restrictions is given. Consider the concept *Interviewer*. Its generalizing concepts are given to the following subsumption hierarchy (cf. (1), Chapter 3):

$$\begin{array}{lll} Interviewer & \sqsubseteq & Journalist \\ Journalist & \sqsubseteq & Person \\ Person & \sqsubseteq & PhysicalThing \end{array}$$

With respect to the provided multimedia material, we assume that e.g. *Journalist* is disjoint (3) with only some of its concept siblings:

$$\begin{array}{lll} Journalist & \sqsubseteq & \neg AssociationActivist \\ Journalist & \sqsubseteq & \neg ProminentPerson \\ Journalist & \sqsubseteq & \neg Politician \end{array}$$

In order to relate instances of these concepts, consider e.g. the role names *isOrganisedBy*, *interviews* and *hasName*. In the following, the domain and range restrictions (4) of these roles are given:

$$\begin{array}{rcl} \exists isOrganisedBy.\top & \sqsubseteq & Event \\ \top & \sqsubseteq & \forall isOrganisedBy.Organisation \\ & & \\ \exists interviews.\top & \sqsubseteq & Interviewer \\ \top & \sqsubseteq & \forall interviews.Interviewee \\ & & \\ \top & \sqsubseteq & \forall hasName.Name \end{array}$$

Note that, since the domain of *hasName* could be nearly everything, this role is not domain restricted. Besides the concept *HumanActivity*, *Interview* is subsumed by a value restriction (5):

$$Interview \sqsubseteq HumanActivity \sqcap \forall hasPart.Microphone$$

Analogously, necessary restrictions of the concept *Name* are

$$Name \sqsubseteq Entity \sqcap \forall hasValue.string$$

where *string* is a concrete domain.

The role *isOrganisedBy* is not functional since an event can be organized by several organisations. In contrast, *hasLocation* and *hasName* are assumed to be functional roles (6):

$$\begin{array}{lll} \top & \sqsubseteq & (\leq 1\ hasLocation) \\ \top & \sqsubseteq & (\leq 1\ hasName) \end{array}$$

Therefore, *Person* can be specified with the definition (7)

$$Person \equiv PhysicalThing \sqcap \forall hasName.PersonName$$

instead of a set of inclusions. Following to this, there are cases in which it is also possible to reason from the right side to the left side, i.e., if there is an individual i for which there is a proof for both conjuncts of the right side of the definition, $Person(i)$ can be inferred.

5 Multimedia Interpretation

This chapter briefly describes the core component of the multimedia interpretation engine that is planned to be used for the CASAM project. Since standard inference services, namely satisfiability-, subsumption-, instance- and relation-checks (as described in Chapter 2) for knowledge bases are not sufficient for an interpretation of multimedia content, the process of abduction is introduced. Abduction is the process of adopting an explanatory hypothesis and covers two operations. The first step comprises the selection and the second step includes the formation of plausible hypotheses.

5.1 Abduction Rules

In contrast to the well known deduction process, where the conclusion goes from some causes to an effect, the abduction process goes from an effect to some causes. When talking about the effect in the context of the CASAM project, the observations generated by the KDMA component are meant and the causes are the explanations in the ontology. Therefore abduction can be considered as reasoning from observations to explanations. The signature of an abduction rule is given as

$$\begin{aligned} A_1(X_1), AR_1(X_1, X_2), A_2(X_2) \leftarrow & E(W), \\ & IR_1(W, X_1), \\ & IR_2(W, X_2) \end{aligned}$$

where $A_{1,2}$ are atomic concepts, $AR_1(X_1, X_2)$ is an analysis role, $E(W)$ is an event and $IR_{1,2}$ are interpretation roles. An atomic concept can be an event, a physical thing or an attribute. The analysis role will be derived directly from the KDMA analysis component. It is important to know that the consequent of an abduction rule (the part of the rule the arrow is pointing at) can consist of any plausible combination of the occurring terms which gives a total of three combinations (only $A_1(X_1)$, $A_1(X_1)$ together with $A_2(X_2)$, $A_1(X_1)$ together with $AR_1(X_1, X_2)$ and $A_2(X_2)$) for the above signature. Later on it is planned to have even more complex consequents. The antecedent (the right-hand side of the rule) is composed according to the consequent. To clarify this fact, the following example is given.

$$\begin{aligned} Person(X_1), holds(X_1, X_2), Chainsaw(X_2) \leftarrow & ForestClearing(W), \\ & hasCauser(W, X_1), \\ & isDoneByTool(W, X_2) \end{aligned}$$

During the abduction process it is possible that there are more than one possible explanations for one observation. Image a second example given as follows:

$$\begin{aligned} Person(X_1), holds(X_1, X_2), Chainsaw(X_2) \leftarrow & LumberJackChampionchip(W), \\ & hasParticipant(W, X_1), \\ & hasCommercialFor(W, X_2) \end{aligned}$$

The consequents of both abduction rules are identical and both explanations $ForestClearing$ and $LumberJackChampionchip$ are possible explanations for the analysis result. In such a situation it is necessary to choose the most probable one. This is where probabilities come into play and will be used to solve this problem. How this is done is part of further research.

6 Expressivity Justification of the Language $\mathcal{ALH}_f$

We have selected $\mathcal{ALH}_f$ language as the representation language in CASAM project. In this section we discuss the expressivity of this language.

6.1 What can be expressed in $\mathcal{ALH}_f$?

Everything that can be represented in **Entity-Relationship** (**ER**) Modelling and, additionally, **IsA-Relationships** can also be expressed in $\mathcal{ALH}_f$ except existential restriction, cardinality constraint and partitioning. (The combination of Entity-relationship modelling and IsA relationship is called extended ER model.) In the following sections we have short definitions for Entity-relationship modelling and IsA relationship.

6.1.1 Entity-Relationship Modelling

Definition 6.1 *The **signature** of entity-relationship modelling $\mathcal{M}$ is a tuple $\mathcal{S} = (\boldsymbol{E}, \boldsymbol{R})$ consisting of a set $\boldsymbol{E} = \{E_1, \ldots, E_n\}$ of all entity names E_i and the set $\boldsymbol{R} = \{R_1, \ldots, R_m\}$ of all relationship names R_i to be considered for $\mathcal{M}$ whereas $\boldsymbol{E} \cap \boldsymbol{R} = \emptyset$.*

Entities model the involved objects whereas the relationships model the connections among the entities. There is a aduality beween the signature of an ontology and the signature of ER modelling. An *entity* in ER modelling corresponds to a *concept* in an ontology. Similarly, a *relationship* in ER modelling corresponds to a *role* in an ontology. In ER modelling an entity is depicted by a rectangle and a relationship by a diamond.

Consider the following example: A conference is organized by an organization. In this example there are two disjoint entities $Conference$ and $Organization$ and a relationship *isOrganizedBy*. The next figure depicts the signature of this example:

Figure 1: Example of a signature in an ER diagram.

Domain of *isOrganizedBy* is $Conference$ and the range is $Organization$. The functional restriction of this example is N to 1 since a $Conference$ is organized by at most one $Organization$ whereas an $Organization$ can organize multiple $Conference$s. The next figure depicts the ER-diagram of this example with the related functionalities:

Figure 2: Example of an ER diagram with the functionalities

In this example there is a functionality on the right side.

6.1.2 IsA-Relationship

To model entity type hierarchies we use IsA in ER approach which means that some entity types are subtypes of others. For example, entity type *Earthquake* is a subtype of the entity type $EnvironmentalIncident$:

$$Earthquake \sqsubseteq EnvironmentalIncident \tag{8}$$

Consequently, all attributes of $EnvironmentalIncident$ are also attributes of $Earthquake$. This example is depicted in the following figure:

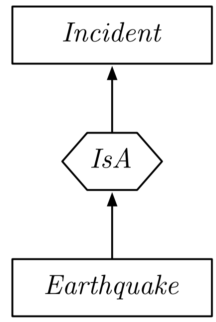

Figure 3: Example of an IsA hierarchy

The other point is that multiple inheritance is also possible.

6.2 What cannot be expressed in $\mathcal{ALH}_f$?

As it was mentioned before, $\mathcal{ALH}_f$ does not have the complete expressivity of Entity-Relationship modelling. The concepts Which cannot be covered are:

1. Existential restriction

2. Cardinality constraint

3. Partitioning

In the following sections we have short definitions for the above concepts.

6.2.1 Existential Restriction

Existential restriction called also totality or participation constraint indicates the presence of each instance of an entity in a relationship. Consider the next example: A conference is organized by exactly one organization. In other words, every $Conference$ occurs on the left-hand side of *isOrganizedBy* relationship. The ER diagram of this example with existential restrictions is depicted in the following figure:

Figure 4: Example of an ER diagram with totality on the left side.

6.2.2 Cardinality Constraint

Cardinality constraint called also general participation constraint is a statement in the form $min \ldots max$ assigned to a relationship regarding to an entity. It is actually a restriction for the number of relationship instances which can occur. Assume that a conference is organized by exactly one single organization. This means that any $Conference$ instance can occur in more than one relationship of type *isOrganizedBy*. Additionally, an *Organization* can organize multiple $Conference$s. Consequently, the cardinalities of this example are for $Conference$ $(1, \infty)$ and for *Organization* $(0, 1)$. The next figure illustrates the ER diagram of this example with cardinalities:

Figure 5: Cardinality in ER diagram

The cardinality on the left hand side cannot be represented whereas the representation of the right functionality is possible.

6.2.3 Partitioning

Assume set A is divided into two disjoint sets B and C where $A = B \sqcup C$. In this case it is said that B and C are partitions of A. This feature is called completeness.

Consider *EnvironmentalIncident* is divided into two disjoint events *Earthquake* and *Flood* where the completeness feature is indicated as follows:

$$EnvironmentalIncident = Earthquake \sqcup Flood \tag{9}$$

The next figure shows the partitioning of *EnvironmentalIncident*:

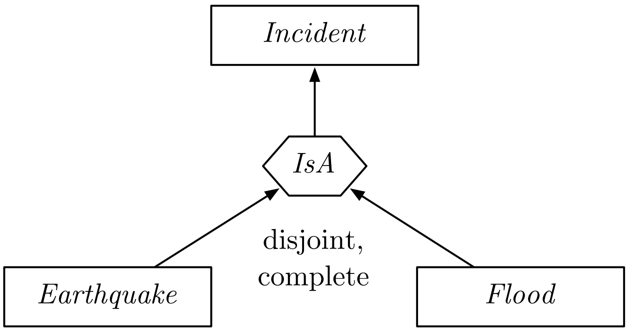

Figure 6: Partioning of *EnvironmentalIncident* into *Earthquake* and *Flood*.

6.3 Why is the union operator left out?

In this section we explain why the union operator is not required in the representation language. Assume the following axioms:

$$Manager \sqsubseteq AreaManager \sqcup TopManager \tag{10}$$
$$AreaManager \sqsubseteq \neg TopManager \tag{11}$$

and consider that Andrea is a manager indicated by $Manager(andrea)$. Consequently, we have:

$$(AreaManager \sqcup TopManager)(andrea) \tag{12}$$

But it does not follow neither $AreaManager(andrea)$ nor $TopManager(andrea)$. Since this operator often does not provide us any further information, we exclude it from the representation language, gaining significantly less complex reasoning.

6.4 $\mathcal{ALH}_f$ vs. $\mathcal{ELH}_f$

In the following we compare the language $\mathcal{ALH}_f$ with the language $\mathcal{ELH}_f$ by means of an example and explain why we did not select $\mathcal{ELH}_f$ as the representation language in the CASAM project.

The difference between these two languages is that in $\mathcal{ALH}_f$ value restrictions can be expressed on the right side of inclusion axioms (resp. definitions) while in $\mathcal{ELH}_f$ this holds for existential restrictions.

Consider the following $\mathcal{ALH}_f$ inclusion-axiom:

$$FireFight \sqsubseteq \forall hasParticipant.FireFighter \tag{13}$$

and assume $FireFight(f_1)$ and $hasParticipant(f_1, obj_2)$. Consequently we have the following assertion:

$$FireFighter(obj_2) \tag{14}$$

This means there is an explicit $FireFighter$ individual.

Analogously, consider the following $\mathcal{ELH}_f$-inclusion axiom:

$$FireFight \sqsubseteq \exists hasParticipant.FireFighter \tag{15}$$

and assume $FireFight(f_1)$. Consequently we have the following assertions:

$$hasParticipant(f_1, ?)$$
$$FireFighter(?)$$

This means there is an anonymous $FireFighter$. In other words, this individual is unknown and cannot be queried for. This is one of the disadvantages of $\mathcal{ELH}_f$ language and a reason not to take this language as the representation language for the CASAM project.

Besides the disadvantages regarding the retrieval scenario, another argument for not using $\mathcal{ELH}_f$ with its existential restrictions is that they are not needed because the interpretation process states explicit individuals in the ABox for objects that could be inferred with respect to the ontology. Automatically generated names are given to these individuals and therefore retrieval on them is possible.

For example, assume the KDMA component is able to detect a $FireFight$ in a media document and has stated this with an assertion $FireFight(f_1)$. Also assume that the interpretation process, which is based on abduction, uses an abduction rule as follows:

$$FireFight(x) \longleftarrow Event(x), hasParticipant(x, y), FireFighter(y) \tag{16}$$

Informally speaking, this rule states that a $FireFight$ can be explained as an event which has a participant of the type $FireFighter$.

Using this explanation, interpretation will add the following assertions to the assertion already stated by KDMA.

$$FireFighter(new\_Ind1)$$
$$hasParticipant(f_1, new\_Ind1)$$

Note that an explicit name has been given to the $FireFighter$, namely new_Ind1.

This example shows the advantage of using abduction for interpretation and the language $\mathcal{ALH}_f$ for the ontology over a language with existential restrictions like $\mathcal{ELH}_f$. As in the case of unions, by leaving out this kind of expressivity the performance of reasoning processes will increase significantly.

7 Conclusion

In this deliverable we defined an ontology framework for the environmental domain of the CASAM project. In Chapter 2 we described the syntax and semantics of the selected representation language $\mathcal{ALH}_f$. Additionally, querying in description logic languages is discussed. In Chapter 3 we introduced the concept of the signature of an ontology as the set of concept and role names all partners should agree on. Based on this signature, we presented design patterns for the CASAM domain ontology. Besides the ability to build up a taxonomy, we propose to specify disjoint concepts, domain and range restrictions for roles, functional roles, value restrictions as well as definitions including value restrictions. In Chapter 4 we described the environmental domain ontology. In this chapter we have categorized the atomic concepts into *Event*, *PhysicalThing* and *Entity*. Furthermore, in this chapter an example considering a specific part of the first version of the environmental domain is given. In Chapter 5 the multimedia interpretation process based on abduction rules is explained. In Chapter 6 the expressivity of the selected language $\mathcal{ALH}_f$ is justified. It has been discussed what can be expressed and what cannot be expressed. This is done based on the comparison of this language with the entity relationship model. We explained why the union operator as well as existential quantification are not worthwhile in the representation language, since without these kinds of expressivity, we gain significantly better performance.

References

[Baader et al., 2003] Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F. (January 2003). *The Description Logic Handbook: Theory, Implementation and Application.* Cambridge UP: Cambridge, NY.

[Baader and Hanschke, 1991] Baader, F. and Hanschke, P. (1991). A scheme for integrating concrete domains into concept languages. *International Conference on Artificial Intelligence*, page 452457.

[Baader et al., 2005] Baader, F., Horrocks, I., and Sattler, U. (2005). *Description logics as ontology languages for the Semantic Web Mechanizing*, pages 228–248. LNAI 2605, Springer. http://www.cs.man.ac.uk/˜horrocks/Publications/download/2003/BaHS03.pdf.

[Wessel and Möller, 2005] Wessel, M. and Möller, R. (2005). A high performance semantic web query answering engine. In *International Workshop on Description Logics.*

A Screenshots from the ontology

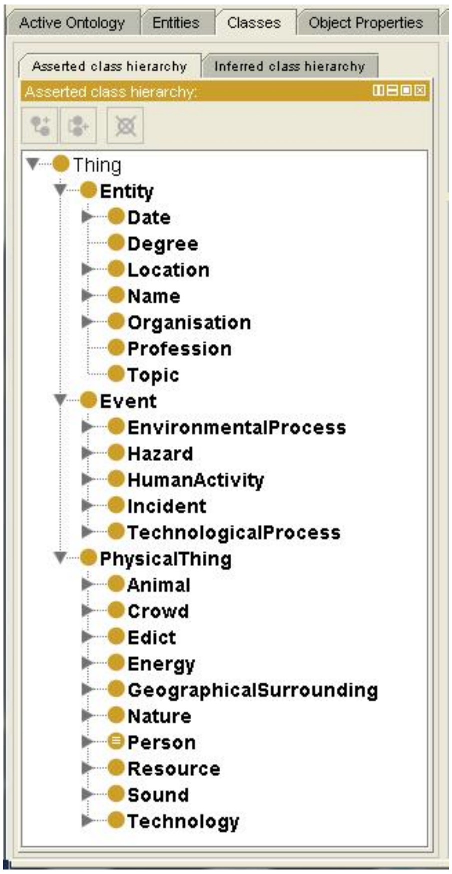

Figure 7: Taxonomy with second level of concept depth

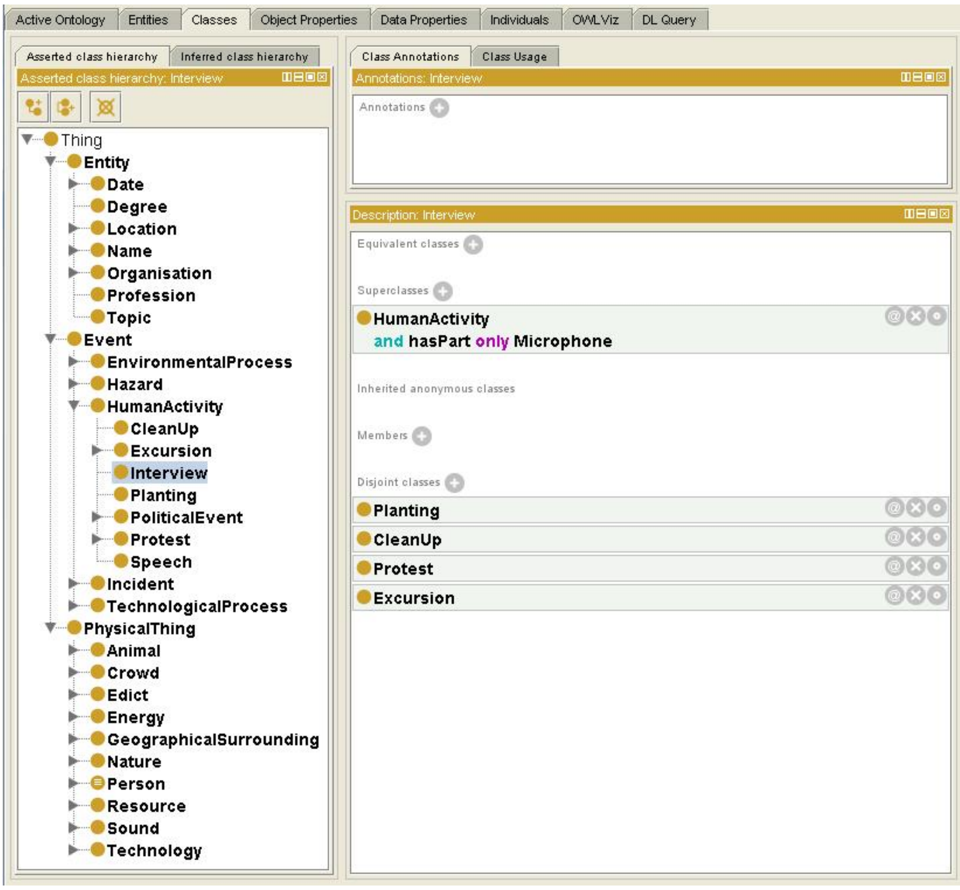

Figure 8: Necessary conditions for the concept *Interview*

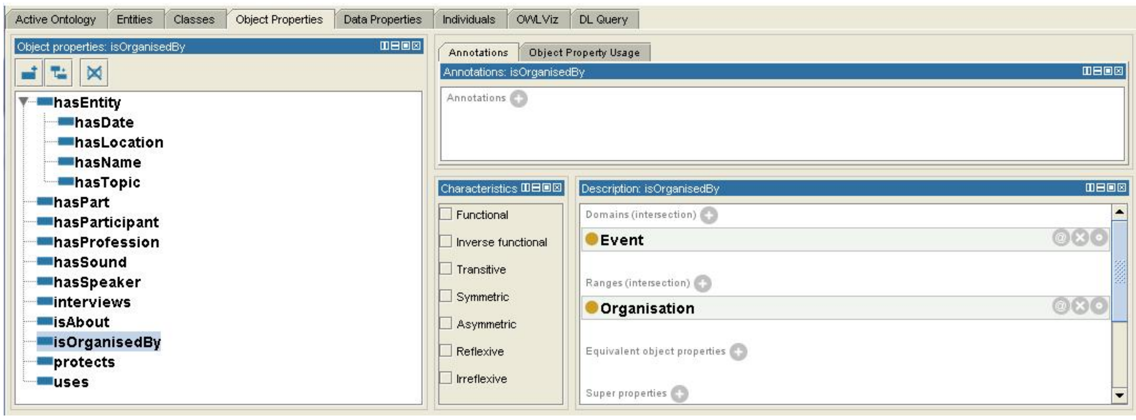

Figure 9: Domain and range restrictions for the role *isOrganisedBy*