Dealing Efficiently with Ontology-Enhanced Linked Data for Multimedia

Oliver Gries[1], Ralf Möller[2], Anahita Nafissi[3], Maurice Rosenfeld[4], Kamil Sokolski[5], and Sebastian Wandelt[6]

[1] Hamburg University of Technology, Hamburg, Germany
[2] University of Lübeck, Lübeck, Germany
moeller@ifis.uni-luebeck.de
[3] Amirkabir University of Technology, Tehran, Iran
[4] Lufthansa Industry Solutions, Hamburg, Germany
[5] Nordex Energy GmbH, Hamburg, Germany
[6] Humboldt University of Technology, Berlin, Germany

Abstract. In order to provide automatic ontology-based multimedia annotation for producing linked data, scalable high-level media interpretation processes on (video) streams are required. In this paper we shortly describe an abductive media interpretation agent, and based on a Multimedia Content Ontology we introduce partitioning techniques for huge sets of time-related annotation assertions such that interpretation as well as retrieval processes refer to manageable sets of metadata.

Keywords: linked data, multimedia interpretation, stream processing

1 Introduction

A large amount of multimedia content is available on the Web, and these days appropriate multimedia documents can hardly be systematically found using keyword-based search. Therefore, the field of Linked Data has emerged [1]. Linked data are also called rich semantic media in the literature [2]. These research fields investigate the derivation and management of symbolic descriptions for multimedia content. Symbolic descriptions are anchored at various parts of a multimedia object, and they can be used to link various (parts of) multimedia objects. Hence the term linked data has emerged. Symbolic content descriptions approximate human-level interpretations of media content, and, therefore, can be used for systematic document retrieval based on high-level topic-based queries. Retrieval based on linked data can be enhanced if retrieval processes are based on ontologies, namely a domain ontology and a general ontology for describing document structure and content (see, e.g., [3]).

To some extent linked data can be automatically derived using existing data-driven media analysis systems. However, there still exists a gap between, for instance, low-level image/video analysis and high-level image/video interpretation, not to mention human-level understanding. Thus, analysis-level results obtained from state-of-the-art tools have to be augmented with more abstract symbolic descriptions. This is accomplished in an automatic process which we call *media interpretation*. Recent research in the area of ontology-based media interpretation has shown enormous advances, and we assume that media interpretation processes can safely generate linked data, to be used in ontology-oriented media retrieval processes. For brevity, linked-data generation is also called (automatic) annotation in this paper, and we focus on videos as multimedia objects in order to be as concrete as possible.

The sheer amount of assertions for appropriately describing the content of large media objects makes media interpretation as well as annotation-based retrieval increasingly difficult. In this paper we advance the state-of-the-art in several areas by: 1. Proposing a description language for video annotations that supports scalable high-level reasoning about video content (interpretation as well as retrieval). 2. Explaining ontology-based reasoning techniques for an annotation agent, which is used to compute high-level interpretations of videos. 3. Showing how to support decomposition-based scalability for reasoning in the context of long streams of video content.

There already exist various proposals for annotation languages (see, e.g., early approaches based on MPEG-7 [4, 5] or newer ones dedicated to knowledge management [6]). However, none of the languages has been developed while keeping in mind scalable stream-based reasoning w.r.t. an ontology (rather than mere data retrieval). Reasoning is used for media interpretation, which is a service being used for computer-aided semantics annotation of multimedia (see the EU project CASAM `http://www.casam-project.eu/`). The CASAM Multimedia Content Ontology introduced in this paper (called MCO for short) is an extension and modification of a previous multimedia ontology described in [7].

Scalability is a significant issue in at least two respects. If we talk about interpretation of a video document, then, on the one hand, there is the time dimension to be considered. On the other hand, another dimension is the interpretation depth. As we have argued above, we can assume that interpretation is based on explicit (symbolic) low-level information for each perceptive unit. A perceptive unit, such as a video shots, is called “segment” in MCO. The aim of interpretation is to compute high-level information for a segment given the knowledge acquired so far. Thus, in our annotation language we need to be able to represent time information as well as to support the ability to draw conclusions on higher levels of interpretation. Hence, the notion of a segment has to be appropriately defined using an ontology, and assertions representing interpretation results at various levels of detail have to be attached to segments using an appropriate annotation language.

It can easily be seen that this kind of two-dimensional streaming scenario, with multiple streams for multiple modalities, yields a significant growth of assertions over time. Although our low-level annotation language is based on a description logic for which efficient typical-case reasoning systems are known, we need to exploit new partitioning techniques to break down the data descriptions used for interpretation into smaller pieces to be handled over time. This is even more important if low-level results become available for time frames in an asynchronous way (maybe with substantial time delays according to the intricacies of certain tools for different modalities). In order to improve scalability, we identify and use locality in the video stream. Given an annotation of a video stream, we use split-operations to compute so called "islands," which are sufficient for reasoning with respect to the current state of annotation. Running interpretation tasks on separate islands (instead of the whole set of assertions) improves performance significantly.

The remaining part of the paper is structured as follows. Section 2 presents our motivation for choosing the description logic $\mathcal{ALH}^{-}_{f\,\mathcal{R}^+}(\mathcal{D})$ as an ontology language for definition of an annotation language MCO. In Section 3, we introduce the Multimedia Content Ontology in detail and explain how $\mathcal{ALH}^{-}_{f\,\mathcal{R}^+}(\mathcal{D})$ is used to represent the content of multimedia documents. We discuss scalability issues and present solutions for partitioning large sets of assertions into manageable "islands" or "chunks" in Section 4 such that interpretation processes run efficiently. Several examples demonstrate the effectiveness of the proposed techniques. We conclude in Section 5. Due to space constraints, *retrieval* is not covered in this paper (We refer to [8] for details).

2 Representation of Multimedia Content

In order to describe multimedia documents in terms of annotations (stored as metadata), the Moving Pictures Experts Group (MPEG) has specified the ISO standard Multimedia Content Description Interface, also denoted as MPEG-7 [4]. In this framework, XML descriptions of multimedia data are associated with content, with the objective to allow for efficient search and retrieval of multimedia documents. The MPEG-7 schema language provides for restrictions on valid media descriptions, for which XML query languages are defined. If, in the context of linked data, the inherent problem of XML query languages comes into play, namely the lack of facilities for querying the name of a relation of which a certain annotation tuple is an element, then RDF-based representations are beneficial. Proposals for using RDF in the context of MPEG-7-alike representations have been discussed in the literature as well (e.g., [5]), and retrieval languages such as SPARQL can be used to find media objects based on RDF content descriptions. RDF query answering with respect to ontologies means that data (tuples, or triples to be more precise) that can be inferred w.r.t. the ontology are implicitly added to what is given explicitly in the RDF annotation. Efficient query engines might not materialize implicit tuples, though. In this case, given the implicit tuples (deductive closure), more media objects are likely to be found if ontologies come into play for query answering. Note that w.r.t. an ontology, a set of RDF triples can also become inconsistent. Inconsistencies can be detected with reasoning engines, but this is not relevant throughout this paper (although inconsistencies also restrict the set of possible annotations in the same way as an XML schema restricts the set of valid annotations).

Ontology languages such as RDFS or OWL2 are languages which have a formal semantics, a feature that is beneficial for formally defining decision problems and checking correctness of corresponding decision procedures (aka inference algorithms). However, specific ontologies (aka knowledge bases) specified using an expressive ontology language reveal more of the "semantics" of media documents as mere RDF triples do. Ontologies achieve this by adding (lots of) implicit content description tuples to the annotations given explicitly as part of the documents' annotations. Thus, we have "semantics" in the sense of formal semantics of a representation language and "semantics" in the sense of implicit tuples added to the explicitly given ones. Many papers in the Semantic Web literature amalgamate these two kinds of "semantics", suggesting that semantics in the sense of content descriptions come for free using formal representation languages. For the latter notion of "semantics," we prefer the name content description in order not to confuse the reader. Content descriptions do not come for free but must be derived using media interpretation processes, which require dedicated knowledge bases for interpretation knowledge [9].

In MPEG-7, a multimedia document is related to its modality specific content, composed of, e.g., video, audio, or text, and each of these parts consists of a set of segments specifying "regions" of modality specific data. At first, we believe it is necessary to be able to specify more general resp. more specific concepts and roles (e.g. that video content also is of type multimedia content) for building up a taxonomy. Further, it is important to be able to specify concept disjointness. In order to represent relations (e.g. from content to segments) roles can be specified, whose domain and range usually are constrained to a specific concept (e.g., the role *hasMediaDecomposition* is constrained to only relate instances of multimedia content to multimedia segments) and which are possibly functional or transitive. In addition, we propose that for modality specific concepts the range of roles is further restricted to modality specific concepts (e.g. audio content is only allowed to be related to audio segments). Finally, for representing multimedia content it is usually necessary to be able to specify concrete domains such as integers or strings.

We argue that this expressivity is sufficient for the representation and interpretation of multimedia content for a large range of problems. For example, we propose to abandon existential restrictions on the right side of inclusion axioms, since we believe that it is not required to constrain multimedia content descriptions to consist of "anonymous" individuals of a specific type (which cannot be retrieved explicitly [10]). The respective DL is denoted $\mathcal{ALH}^{-}_{f\mathcal{R}^+}(\mathcal{D})$ (restricted attributive concept language with role hierarchies, functional roles, transitive roles and concrete domains). We made several experiments with the DL reasoner RacerPro [11] strongly indicating that reasoning with $\mathcal{ALH}^{-}_{f\mathcal{R}^+}(\mathcal{D})$ is efficient.

We now shortly introduce the descripton logic (DL) nomenclature. A DL *signature* is a tuple $\mathcal{S} = (\mathbf{CN}, \mathbf{RN}, \mathbf{AN}, \mathbf{IN})$, where $\mathbf{CN} = \{A_1, ..., A_n\}$ is the set of concept names (we also use A for concept names in the sequel). $\mathbf{RN} = \{R_1, ..., R_m\}$ is the set of role names. Further, $\mathbf{AN}$ is a set of concrete domain attributes (i.e., roles whose range is a concrete domain). The signature also contains a component $\mathbf{IN}$ indicating a set of individuals. A DL knowledge base

$\mathcal{O}_{\mathcal{S}} = (\mathcal{T}, \mathcal{A})$, defined 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 we just write $\mathcal{O}$ if the signature is clear from context. An $\mathcal{ALCH}^{-}_{f\mathcal{R}^+}(\mathcal{D})$ Tbox is a set of axioms $A_1 \sqsubseteq A_2$ and $R_1 \sqsubseteq R_2$ (atomic subsumption), $A_1 \sqsubseteq \neg A_2$ (disjointness), $\exists R.\top \sqsubseteq A$ and $\top \sqsubseteq \forall R.A$ (domain and range restrictions on roles), $\top \sqsubseteq (\leq 1\, R)$ (functional roles), $Trans(R)$ (transitive roles) and $A_1 \sqsubseteq \forall R.A_2$ (local range restrictions on roles). An Abox $\mathcal{A}$ i is a set of concept assertions $A(a)$ and role assertions $R(a, b)$, where A is a concept name, R is a role name, and a, b represent individuals. Aboxes can also contain equality $(a = b)$ and inequality assertions $(a \neq b)$ as well as attribute assertions of the form $Attr(a, val)$ where $Attr$ is an attribute and val is either a string or an integer (with the obvious denotation). For a detailed introduction to the incorporation of concrete domains into DLs, to the semantics of concepts and roles, as well as an introduction to the satisfiability conditions for axioms and assertions we refer to [12] and [13], respectively. Standard DL decision problems are also formally defined in [13] (e.g., computing the concept and role hierarchies, as well as concept-based and conjunctive instance retrieval).

3 The Multimedia Content Ontology

In this section, the CASAM Multimedia Content Ontology is presented to an extent that the solution to scalabilitiy problems can be understood. The full MCO can be found at `http://www.sts.tu-harburg.de/casam/mco.owl`. In contrast to approaches transforming the complete MPEG-7 standard to RDFS [5] or OWL [14], our approach is inspired by using only those parts of MPEG-7 describing a general structure for multimedia documents. The main objective is to effectively exploit quantitative and qualitative time information in order to relate co-occurring observations. Co-occurrences are detected either within the same or between different modalities regarding the video shots. In the following, we focus on axioms relating concept and role names required for these capabilities.

3.1 Concept Hierarchy and Role Hierarchy

In Fig. 3.1 the concept hierarchy (on the left) and the role hierarchy (on the right) of the multimedia ontology is shown. A complete multimedia document is represented by the concept *MultimediaDocument*. Only one instance of type *MultimediaDocument* should be specified for a document to be annotated.

Individuals which are instances of (subconcepts of) *MultimediaContent* represent different modalities of the video. The concept *VideoContent* represents the video modality and holds all video segments. In the same way *AudioContent* holds all segments from the audio modality.

TextContent represents text paragraphs associated with certain segments or the whole video. Auxiliary text documents that are related with the whole annotated video are represented by the subconcept *AuxiliaryContent*. During the annotation process, a user can make free text annotations, which describe the

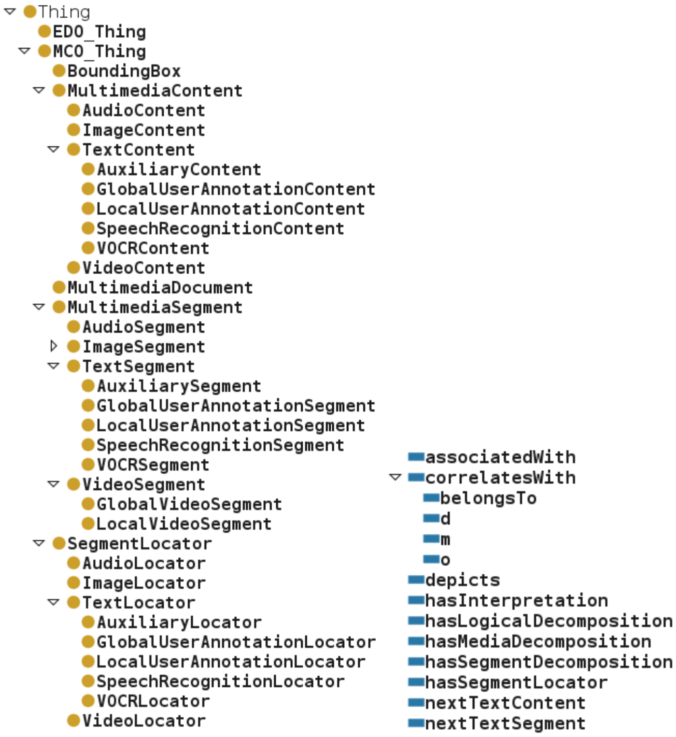

Fig. 1. Concept Hierarchy (lhs) and role hierarchy (rhs) structure of the CASAM Multimedia Content Ontology

whole multimedia document or a single segment (shot) of the video. These free text annotations are represented by *GlobalUserAnnotationContent* resp. *LocalUserAnnotationContent*. As speech, recognized in the video, is transformed into text, the concept *SpeechRecognitionContent* is also subsumed by *TextContent*.

To represent parts of the content, *MultimediaContent* instances can be decomposed into *MultimediaSegment* instances. *TextSegment* refers to words in the text modality. The concept *SegmentLocator* is used to specify start and end of segments. The concrete values of start and end represent temporal position information for audio and video, or denote character positions for text. *BoundingBox* is used to determine the position of a recognized object in a video frame. All concepts within the same hierarchy level are disjoint.

The role *hasLogicalDecomposition* decomposes the whole media document into the different parts by relating instances of type *MultimediaDocument* with modality specific content description individuals (that is, instances of the concept *MultimediaContent*). An individual of the type *MultimediaContent* is associated to its segments by the role *hasMediaDecomposition*. To relate an individual of type *MultimediaSegment* with its locators, the role *hasSegmentLocator* is used.

The roles *nextTextContent* and *nextTextSegment* are used to specify the order in the text paragraphs resp. words. Both are transitive roles. Subroles of *correlatesWith* can be used to represent associations between content descriptors. A *TextContent* instance can be related to a *VideoSegment* using the role *belongsTo*. We also use a small subset of the Allen relations [15] to relate video segment, or, more precisely, the locators associated with video segments. Note that we do not require reasoning on Allen relations since the corresponding relations are generated based on quantitative data. While o (overlap) describes an intersection between audio and video locators, m (meets) describes the alignment of two video or two audio segments. Please note that we compute (qualitative) relations such as o using (quantitative) information about locator objects. Quantitative information is given in terms of restrictions on values for attributes *hasStart* and *hasEnd* (see Section 3.3).

The role *depicts* is used to establish a mapping from individuals of the Multimedia Content Ontology to observations from the domain ontology that were extracted by analysis modules. In a similar way as *depicts*, *hasInterpretation* provides a map to individuals that were generated as a part of interpretations of observations. To represent the aggregating characteristic of high-level interpretations, the role *associatedWith* is used to related high-level interpretations with other interpretations or directly with observations.

3.2 Range Restrictions

Range restrictions on roles constrain the corresponding role fillers to be of a specific type. For example,

$$\top \sqsubseteq \forall hasMediaDecomposition.MultimediaSegment$$

defines the range restrictions on the role $hasMediaDecomposition$ such that the role filler is constrained to be of type $MultimediaSegment$.

Local range restrictions constrain the range of roles further when the role is applied to a specific concept. The local range restriction

$$AudioContent \sqsubseteq \forall hasMediaDecomposition.AudioSegment$$

specifies that the range of the role $hasMediaDecomposition$ associated with the concept $AudioContent$ is further restricted to $AudioSegment$.

3.3 Attribute Values

The attributes *hasStart* and *hasEnd* are used to specify time information of video or audio segments. For example:

$$AudioLocator(as_1), AudioLocator(al_1), hasSegmentLocator(as_1, al_1)$$

$$hasStart(al_1, \text{“00:43”}), hasEnd(al_1, \text{“00:52”})$$

defines the starting time of an $AudioSegment$, e.g., as_1 by specifying concrete values to its corresponding $AudioLocator$ al_1. Integer values are used to specify character positions to identify words in larger text strings. Also regarding the text modality, the property *hasConcreteValue* is used to associate strings to instances of specific types such as $CityName$.

Given quantitative information about start and end time, qualitative relations between locator instances are computed by the media interpretation agent. From the potential 13 qualitative relations defined by Allen [15] we explicitly represent o ($overlaps$) , d ($during$), and m ($meets$) between segment locators. As shown in the next section the main motivation for switching from a quantitative temporal representation to a qualitative one is to achieve scalability. Given the MI_Agent introduced above (see 4.1 for more details), qualitative relations allows to partition the interpretation Abox(es) that always grow(s) over time.

4 Scalable Video Interpretation

As we have seen, for improving shot-based video annotation, interpretations are computed for co-occurrences of locator individuals according to temporal information. In the course of the video interpretations, Aboxes grow significantly. If, e.g., a particular video segment is focused on because there is a new assertion coming in, referring to this video segment, only very few other assertions are relevant. Large parts of Aboxes (e.g., for temporally far away parts of the video) need not be processed. In this section we formalize the subdivision of large Aboxes into meaningful parts (partitions) such that reasoning problems are handled in the same way as with the large Abox. Reasoning on the small partitions, also called island reasoning, is known to improve reasoning performance significantly [16]. We start with the introduction of some important aspects of the media interpretation agent.

4.1 The Multimedia Interpretation Agent

In [9], an agent for Multimedia Interpretation (*MI_Agent*) was introduced. It uses a probabilistic interpretation engine which, among others, is based upon abduction. The idea is to generate explanations for observations in the form of hypothesized Abox assertions. Given the added assertions and a set of rules as part of the agent's knowledge base, the observations are then entailed. The agent computes assertions that “support” the observations. The *MI_Agent* receives percepts in the form of assertions that represent the ongoing video analysis and annotation process. The assertions are received in a streaming way by the *MI_Agent* in small bunches, which we formalize as sets Γ here.

Each Γ is added to the Abox that the agent maintains. Subsequently, a set of *forward-chaining rules* is applied. The general form of these rules is

$$Q_1(\underline{Y_1}), \ldots, Q_n(\underline{Y_n}) \rightarrow P(\underline{X})$$

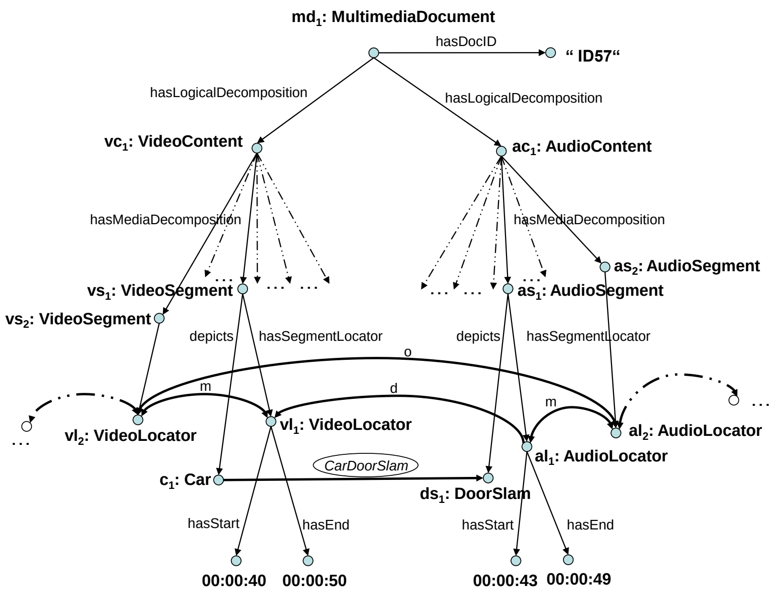

Fig. 2. A multimedia content structure with co-occurring domain ontology individuals

where $Q_1, \ldots, Q_n, P$ denote concept or role names and underlined letters denote (possible) tuples of all-quantified variables with the condition that each variable appearing in $P(\underline{X})$ does also appear in at least one $Q_i(\underline{Y_i})$. In order to be able to apply a rule, appropriate individuals have to be substituted for the variables. Conclusions $P(\underline{i})$ are then added to the Abox. For the conclusions $P(\underline{i})$ the MI_Agent seeks further explanation using an abduction process. The main idea is to backward chain a set of rules of the form introduced above. Due to space restrictions, this process cannot be explained in detail, and we refer to [9]. In any case, if the Aboxes get larger and large, performance will degrade if there are no specific techniques employed.

Example 1 *A car is shown in a video shot, represented by an assertion $Car(c_1)$, and there is the sound of a door slam, represented by an assertion $DoorSlam(ds_1)$. The car and the door slam are associated to video and audio segments, resp. Those, in turn, are associated with locator objects. Now assume that the car and the door slam co-occur, i.e., the locator objects for the audio segment is located during the video segment. Figure 2 depicts a complete scenario for the example.*

Using relations between time points, one might use rules to define a during relation as a view based on the quantitative temporal information for the locator objects. However, using relations between time points, in principle, every locator might be associated with every other locator, and thus the agent can hardly partition the large Abox into smaller parts. Therefore, we have designed the agent in such a way that it adds qualitative relations such as overlaps (o), during (d), and meets (m) to make certain temporal information explicit that is hidden in the quantitative locator time specifications. The motivation for the agent to switch to the more verbose qualitative representation is that the input Abox becomes partitionable.

Qualitative temporal relations are used in forward-chaining rules to compute assertions that are then explained by the agents (see above). For instance, based on the forward-chaining rule

$$\begin{gathered}\forall x, xl, y, yl, w, z \;\; VideoSegment(x), hasSegmentLocator(x, xl), \\ VideoLocator(xl), AudioSegment(y), hasSegmentLocator(y, yl), \\ AudioLocator(yl), d(yl, xl), depicts(x, w), depicts(y, z), \\ Car(w), DoorSlam(z) \rightarrow CarDoorSlam(w, z)\end{gathered}$$

the role assertion $CarDoorSlam(c_1, ds_1)$ (marked with an ellipse in Figure 2) is generated and added to the Abox. This new assertion is seen as a specific observation that requires an explanation [9]. Possible explanations, e.g., are car entry or car exit events, which might be represented using assertions $CarEntry(i_1)$ or $CarExit(i_2)$, where i_1 and i_2 are new individuals. Both individuals are associated with the car end the door slam individuals (role $associatedWith$, see above). Inevitably, in the course of explanation generation, the Abox grows again significantly. This leads to very large Aboxes (imagine the annotation of a two-hour movie) and the application of forward-chaining rules (as well as the abduction process) will be very inefficient, since complex joins for huge relations can hardly be avoided in order to check whether rules are applicable (and to compute the bindings for variables). Pretty soon, the video description Abox does not fit into main memory any longer. In the following, we present a proposal to overcome the problem of Aboxes becoming too large.

4.2 Island Reasoning

As stated before, the input can be considered as a stream. The information content derived from a stream is collected over time and stored together with the interpretations in an Abox or in multiple ones, respectively, if more than one interpretation is possible. These Aboxes are put to the previously introduced agenda $\mathfrak{A}$. The more knowledge is gathered, the larger those Aboxes become and the longer it takes to complete all computations such as applying the forward-chaining rules or arranging the interpretation process itself. Current state-of-the-art DL reasoning systems cannot deal with this amount of information any more, because they rely on in-memory structures. To overcome this problem, in [16] island-based reasoning for $\mathcal{ALCHI}$ ontologies is proposed as a solution. In the meantime the island approach is extended to $\mathcal{SHIQ}(\mathcal{D})$ by a more fine-grained syntactical analysis. Since $\mathcal{SHIQ}(\mathcal{D})$ is a more expressive description logic than $\mathcal{ALH}^{-}_{f\,\mathcal{R}^+}(\mathcal{D})$, the mechanism is also applicable for our annotation language.

The underlying idea is that only a small subset of concept and role assertions called *island* is necessary to perform instance checking for a particular given individual i and a given (complex) concept C. The approach chosen here is to identify role assertions which can be used during the application of a tableau algorithm for instance checking [13] (note that $(\mathcal{T}, \mathcal{A}) \models^? C(i)$ can be reduced to checking whether $(\mathcal{T} \cup \mathcal{A} \cup \{\neg C(i)\})$ is unsatisfiable via a tableau algorithm). First, the ontology is transformed into a normal form, called *shallow normal form*. For the details of the transformation please refer to [16]. Given the shallow normal form, a so-called $\forall$-info structure for an ontology $\mathcal{O}$ is used to determine which concepts are (worst-case) propagated over role assertions in an Abox. This helps to define a notion of separability. The following definition of $\mathcal{O}$-separability is used to determine the importance of role assertions in a given Abox $\mathcal{A}$.

Definition 1. *Given an ontology $\mathcal{O} = (\mathcal{T}, \mathcal{A})$, a role assertion $R(a, b)$ is called* $\mathcal{O}$-separable*, if we have $INC(\mathcal{O})$ iff $INC(\langle \mathcal{T}, \mathcal{A}_2 \}\rangle)$, where*

$$\mathcal{A}_2 = \mathcal{A} \setminus \{R(a,b)\} \cup \{R(a,b'), R(a',b)\} \cup \{b' : C | b : C \in \mathcal{A}\} \cup \{a' : C | a : C \in \mathcal{A}\},$$

s.t. a' and b' are fresh individual names and $INC(\mathcal{O})$ denotes an inconsistent ontology $\mathcal{O}$.

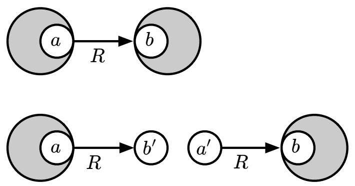

Fig. 3. $\mathcal{O}$-separability of a role R

Figure 3 shows a graphical representation of a role that matches the definition of $\mathcal{O}$-separability. Informally speaking, the idea is that $\mathcal{O}$-separable assertions will never be used to propagate "complex and new information" via role assertions. The extraction of islands for instance checking in an ontology $\mathcal{O}$, given an individual i, is now straightforward. A graph search can be used that starts from an individual i and follows each non-$\mathcal{O}$-separable role assertion in the original Abox, until at most $\mathcal{O}$-separable role assertions are left. All visited assertions are then worst-case relevant for the reasoning process. Regarding the proposed MCO, the objective is that implicit information due to value restrictions $\forall R.A(i)$ prevents a separation for role assertions $R(i,j)$, if $A(j)$ is not explicitly specified in the respective Abox.

Example 1 (cont.) *Applying the definition of $\mathcal{O}$-separability to the Abox depicted in Figure 2, islands are computed as shown in Figure 4. Instead of applying all possible substitutions, the forward-chaining rule does only need to be applied to the island with the locators vl_1 and al_1 in order to add $CarDoorSlam(c_1, ds_1)$. This enables parallel processing for abduction and retrieval scenarios. However, given the local range restriction for $AudioContent$, if as_1 is not explicitly specified as $AudioSegment$ but rather as $MultimediaSegment$, the definition of $\mathcal{O}$-separability would be violated for $hasMediaDecomposition(ac_1, as_1)$—so that the respective island would be larger than before.*

This general Abox modularization approach has proven well regarding scalability issues. For more details, in particular regarding a theoretical and practical underpinning of the island approach, we refer the reader to [17].

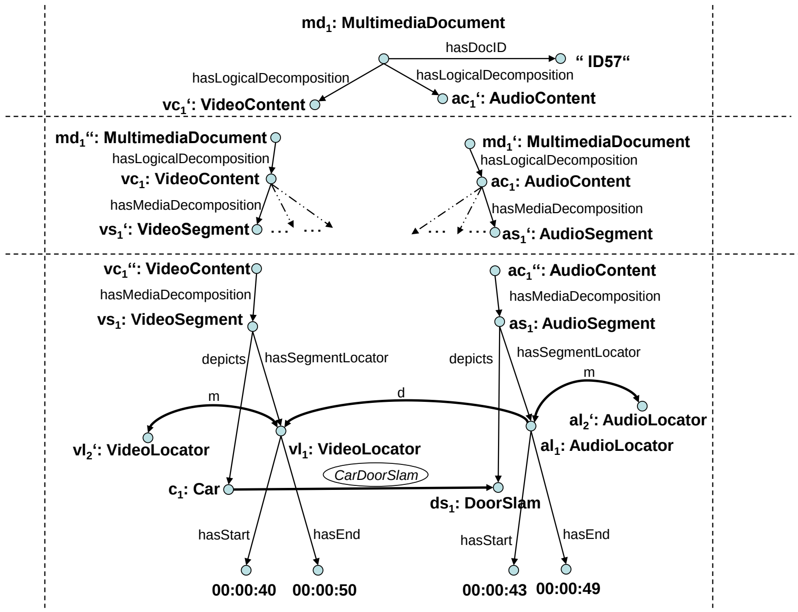

Fig. 4. Multimedia content structure divided into islands

5 Conclusion

Under the consideration of MPEG-7 and [7], a multimedia content ontology has been introduced that is represented with the DL $\mathcal{ALH}^-_{f\,\mathcal{R}^+}(\mathcal{D})$. The expressiveness of this logic has proven to be sufficient for the MCO arising from the scenarios considered in the CASAM project. As the MCO covers most of the relevant concepts, roles and attributes to be expected in video streaming scenarios, the $\mathcal{ALH}^-_{f\,\mathcal{R}^+}(\mathcal{D})$ can be safely assumed to be sufficient for similar multimedia interpretation scenarios. Based on time information in Aboxes corresponding to this MCO, a multimedia agent performs stream-based forward-chaining and abductive backward chaining in order to obtain interpretation possibilities. Partitioning techniques ensure that interpretation Aboxes can be decomposed into manageable parts such that even large videos can be handled (Aboxes can be swapped to secondary memory).

Some initial experiments were performed to see how the approach behaves in the CASAM context. The results are very promising and almost all roles were $\mathcal{O}$-separable after qualitative assertions were added to Aboxes such that quantitative information is no longer required. Thus, switching from a quantitative to a qualitative representation provides practical benefits for the agent.

Our work complements other work on stream reasoning, i.e., for efficiently maintaining materialized views as described in [18]. We show that in some cases the views based on quantitative information can be avoided.

References

1. Castano, S., Espinosa, S., Ferrara, A., Karkaletsis, V., Kaya, A., Möller, R., Montanelli, S., Petasis, G., Wessel, M.: Multimedia interpretation for dynamic ontology evolution. Journal of Logic and Computation **19**(5) (2008) 859–897
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web and Information Systems (IJSWIS), **5**(3) (2009) 1–22
3. Espinosa, S., Kaya, A., Möller, R.: The BOEMIE Semantic Browser: A semantic application exploiting rich semantic metadata. In: Proceedings of the Applications of Semantic Technologies Workshop (AST-2009), Lübeck, Germany. (2009)
4. ISO/IEC15938-5FCD: Multimedia content description interface (MPEG-7) (2002)
5. HunterJ, J.: Adding multimedia to the semantic web: Building an MPEG-7 ontology. In: Proc. of the 1st Semantic Web Working Symposium, Stanford University, California, USA. (2001) pp. 261–283
6. Staab, S., Franz, T., Görlitz, O., Saathoff, C., Schenk, S., Sizov, S.: Lifecycle knowledge management: Getting the semantics across in X-Media. In: Foundations of Intelligent Systems, 15th International Symposium, ISMIS 2006, Bari, Italy, September 2006. LNCS, Springer 1–10
7. Dasiopoulou, S., Dalakleidi, T., Tzouvaras, V., Kompatsiaris, Y.: D3.4 - Multimedia ontologies. Boemie, project deliverable, National Technical University of Athens (2008)
8. Wandelt, S., Möller, R.: Updatable island reasoning over alchi ontologies. In: Conference on Knowledge Engineering and Ontology Development (KEOD). (2009) CEUR Workshop Proceedings (Vol. 477).
9. Gries, O., Möller, R., Nafissi, A., Rosenfeld, M., Sokolski, K., Wessel, M.: A probabilistic abduction engine for media interpretation. In Alferes, J., Hitzler, P., Lukasiewicz, T., eds.: Proc. International Conference on Web Reasoning and Rule Systems (RR-2010). (2010)
10. Haarslev, V., Möller, R.: On the scalability of description logic instance retrieval. Journal of Automated Reasoning **41**(2) (2008) 99–142
11. Haarslev, V., Moeller, R.: Racer: A core inference engine for the semantic web. In: Proc. of the 2nd International Workshop on Evaluation of Ontology-based Tools, located at the 2nd International Semantic Web Conference ISWC. (2003)
12. Baader, F., Hanschke, P.: A scheme for integrating concrete domains into concept languages. International Conference on Artificial Intelligence (1991)
13. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation and Application. Cambridge UP: Cambridge, NY. (2003)
14. Garcia, R., Celma, O.: Semantic integration and retrieval of multimedia metadata. In: Proc. of the 4th International Semantic Web Conference (ISWC), Galway, Ireland. (2005)
15. Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. ACM **26**(11) (1983) 832–843
16. Wandelt, S., Möller, R.: Island Reasoning for ALCHI Ontologies. In: Proceedings of the 5th International Conference on Formal Ontology in Information Systems (FOIS-04), IOS Press (2008)
17. Wandelt, S., Möller, R.: Towards abox modularization of semi-expressive description logics. Journal of Applied Ontology **7**(2) (2012) 133–167
18. Barbieri, D., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: Incremental reasoning on streams and rich background knowledge. In Aroyo, L., Antoniou, G.,

Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T., eds.: The Semantic Web: Research and Applications. Volume 6088 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2010) 1–15