GraphOnto: OWL-Based Ontology Management and Multimedia Annotation in the DS-MIRF Framework

Panagiotis Polydoros, Chrisa Tsinaraki and Stavros Christodoulakis
Lab. Of Distributed Multimedia Information Systems,
Technical University of Crete (MUSIC/TUC)
University Campus, Kounoupidiana, Chania, Greece,
{panpolyd, chrisa, stavros}@ced.tuc.gr,
WWW home page: http://www.music.tuc.gr

Abstract. We present here GraphOnto, a software component and an API that allow the definition and editing of both standard-based and domain OWL ontologies and their use in multimedia information system components. In our working environment an OWL Upper Ontology that captures the MPEG-7 MDS is utilized and OWL domain ontologies extend the upper ontology with domain knowledge. The ontologies imported in GraphOnto are parsed so that graphical ontology browsing and editing interfaces are automatically generated. The ontologies are used to guide the semantic multimedia annotation in a standardized manner. OWL Ontology mappings, expressed using OWL constructs, are supported and allow the definition of personalized ontologies. Personalization of the GraphOnto user interface on the application, task and user levels is provided. In addition, GraphOnto interoperates with an MPEG-7 Repository both for semantic entity reuse during multimedia content annotation and for semantic description storage. GraphOnto has been evaluated against Protégé and clearly outperforms it.

1. Introduction

The advent of the Internet and the digital multimedia demonstrated the importance of standardization for the multimedia industry. Interoperability among the well-accepted standards is needed in open environments, in order to allow multimedia content services built on top of different standards to communicate, exchange information, and co-operate. The interoperability support is usually based on ontologies, which essentially represent the consensus reached for semantic descriptions in an application domain. This way, the semantics of the standards are usually captured in upper ontologies. In addition to the multimedia standards, domain knowledge is very useful for important functionality of multimedia information repositories (e.g. indexing, query specification, retrieval, filtering, etc.). Since the structures provided by the generic standards (like MPEG-7 (Chang & al., 2001)) are general-purpose, domain knowledge cannot be adequately represented using them. Thus, we have developed methodologies for extending the standard-based upper ontologies with domain knowledge captured in domain ontologies in the **DS-MIRF (Domain-Specific Multimedia Indexing, Retrieval and Filtering) framework** (Tsinaraki & al., 2004a; Tsinaraki & al., 2004b). However, the size and the complexity of the standards and the domain ontologies make them very difficult and time-consuming to use.

We present here **GraphOnto**, a component that facilitates the definition and population of both Upper (usually standard-based) and domain ontologies and their use in multimedia information system components. Since both upper and domain ontologies may describe very complex concepts, the GraphOnto component provides mechanisms for information hiding, so that the multimedia content annotators can concentrate only in the concepts they intend to use during their work. GraphOnto is based on OWL (Mc Guiness & van Harmelen, 2004), the dominant standard in ontology definition, and can be viewed as both a generic OWL ontology editor and as a semantic annotation component for multimedia content:

- As a complete graphical OWL ontology editor, GraphOnto provides functionality supported by existing ontology editors (among which Protégé is the most advanced) as well as several important additional features: Multiple ontology management, OWL Ontology mappings that allow the definition of personalized ontologies expressed using OWL constructs, and personalization of the GraphOnto user interface on the application, task and user levels. In addition, GraphOnto has been shown to have much better performance than Protégé.
- As a semantic annotation component for multimedia content, GraphOnto is used in the context of the DS-MIRF framework, utilizing and extending the DS-MIRF ontological infrastructure. In particular, GraphOnto allows the semantic annotation of multimedia objects utilizing the DS-MIRF upper ontology, which fully captures the MPEG-7 MDS (ISO/IEC; 2003). In addition, GraphOnto allows the utilization and definition of OWL domain ontologies integrated in the DS-MIRF framework, which extend the upper ontology with domain knowledge. Both the domain ontologies and the semantic multimedia annotations defined using them are transformed, using the DS-MIRF transformation rules, to semantic MPEG-7 multimedia content descriptions. The semantic MPEG-7 descriptions may be stored either in MPEG-7 compliant XML files or in the DS-MIRF MPEG-7 Repository. In addition, the GraphOnto API is utilized in several ontology-based retrieval and filtering application interfaces in the DS-MIRF framework.

The rest of this paper is organized as follows: In section 2, a short description of the standards that influenced the GraphOnto development, namely OWL and MPEG-7, is given. In section 3, an overview of the DS-MIRF framework is presented. In section 4, the GraphOnto architecture is described in detail. In section 5, the functionality of GraphOnto is presented as well as the results of the evaluation of GraphOnto against Protégé. Conclusions and future research directions are presented in section 6.

2. Multimedia and Ontology Description Standards

We present in brief in this section the two standards that influenced the development of GraphOnto: (a) The dominant standard in ontology definition languages, the Web Ontology Language (OWL) (see subsection 2.1), the constructs of which are supported by GraphOnto; and (b) The MPEG-7 and in particular the MPEG-7 Multimedia Description Schemes (MDS), where constructs for the definition of metadata for multimedia content and service description are defined (see subsection 2.2). The MPEG-7 MDS is the basis for semantic multimedia content annotation in GraphOnto.

2.1. The Web Ontology Language (OWL)

The **Web Ontology Language (OWL)** (Mc Guiness & van Harmelen, 2004) is the dominant standard in ontology definition. OWL has been developed according to the description logics paradigm and uses RDF(S) (Brickley & Guha, 2004; Manola & Milles, 2004) syntax. Three OWL species of increasing descriptive power have been specified: OWL Lite, OWL DL and OWL Full. The basic functionality provided by OWL is following:

- **Import of XML Schema Datatypes**, through the rdfs:Datatype construct, for the representation of simple types that extend or restrict the basic datatypes (e.g. ranges etc.).
- **Definition of OWL Classes**, using the owl:class construct, for the representation of sets of individuals sharing some properties. Class hierarchies may be defined using the rdfs:subClassOf construct. Complex OWL classes can be defined via set operators (intersection, union or complement of other classes) or via direct enumeration of their members (OWL Individuals). Relations with other classes may be defined such as equivalence or disjointness.
- **Definition of OWL Properties**, for the representation of the features of the OWL class individuals. Two kinds of properties are provided by OWL: (a) Object Properties, which relate individuals of one OWL class (the property domain) with individuals of another OWL class (the property range). Object properties are defined using the owl:objectProperty construct; and (b) Datatype Properties, which relate individuals belonging to one OWL class (the domain of the property) with values of a given datatype (the range of the property). Datatype properties are defined using the owl:datatypeProperty construct. Property hierarchies may be defined using the rdfs:subPropertyOf construct. OWL properties may be specified to be symmetric, transitive, functional or inverse functional. Relations between properties may be defined such as equivalence and inversion.
- **Definition of Restrictions** on class properties, using the owl:Restriction construct, including type restrictions, cardinality restrictions and value restrictions.
- **Definition of OWL Individuals** having the properties available to the OWL class they belong, according to the existing restrictions. Relations between individuals may be defined such as identity and difference.

2.2. The MPEG-7 MDS

The **MPEG-7 MDS (Multimedia Description Schemes)** provides constructs for the definition of metadata for multimedia content and service description. The MPEG-7 MDS has been defined using the MPEG-7 DDL, which is essentially the XML Schema Language (Fallside 2001), extended with the basic datatypes necessary for the definition of the **Description Schemes (DSs)** of the MPEG-7 MDS. The DSs are essentially complex datatypes used for the description of concepts in their scope. In addition to the DSs, the MPEG-7 MDS provides the **Classification Schemes (CSs)**, which essentially are thesauri comprised of term hierarchies. The 40 standardized CSs of the MPEG-7 MDS contain the allowed values of specific attributes of the DSs of the MPEG-7 MDS. The CSs are XML documents compliant to the XML schema of the MPEG-7 MDS.

The MPEG-7 MDS is comprised of the following major components (ISO/IEC; 2003): (a) The **Basic Elements**, which include schema tools, basic datatypes, mathematical structures, linking and media localization tools as well as basic DSs, which are used as elementary components of more complex DSs; (b) The **Content Description and Management Elements**, which provide the Creation and Production, Media and Usage tools in order to capture content management information and the Structural Aspects and Semantic Aspects tools in order to capture content description information; (c) The **Navigation and Access Elements**, where browsing is supported through multimedia content summary descriptions and multimedia content variations; (d) The **User Interaction Elements**, which are used to describe user preferences regarding multimedia content, as well as multimedia consumption aspects; and (e) The **Content Organization Elements**, where the organization of the multimedia content is addressed by classification, modeling and multimedia document collection specification.

3. Overview of the DS-MIRF Framework

In this section we present a brief overview of the **DS-MIRF (Domain-Specific Multimedia Indexing, Retrieval and Filtering) framework** (Tsinaraki & al., 2004a; Tsinaraki & al., 2004b), where GraphOnto is utilized both as a semantic multimedia annotation component and as a basis, through the GraphOnto API, for ontology-based application development. The DS-MIRF Framework aims to facilitate the development of knowledge-based multimedia applications utilizing and extending the MPEG-7 standard.

The architecture of the DS-MIRF framework as well as the information flow between its components and the interaction with the end-users are illustrated in Figure 1. In this figure, the **multimedia content annotator** is a special type of user that is responsible for the semantic annotation of multimedia documents. He uses an **annotation interface** integrated with the GraphOnto semantic multimedia annotation component, which allows ontology-based semantic annotation and utilizes the ontological infrastructure of the DS-MIRF framework. Since all the ontologies in the DS-MIRF framework are expressed in OWL, the result of the annotation process is an OWL description of the multimedia content. The OWL metadata are then transformed, using the **DS-MIRF transformation rules** (implemented in the GraphOnto component), either to **standard MPEG-7 metadata descriptions** or to **extended MPEG-7 metadata descriptions**, which capture application-specific needs. The extensions are usually specified in other standards or sound models that are expressed in (some of) the DS-MIRF application ontologies. The MPEG-7 metadata – standard or extended – are stored in the **DS-MIRF Metadata Repository**, which is accessed by the end-users through appropriate **application interfaces**. The application interfaces may provide the end-users with multimedia content services like multimedia content retrieval, filtering and delivery. Applications based on OWL ontologies are developed using the GraphOnto API. Applications that already utilize the GraphOnto API are the **OntoNL** Ontology-based natural Language interface Generator for Multimedia Repositories (Karanastasi & Christodoulakis, 2005) and the **SyMPA** (System for Multimedia Presentation Authoring), which is being integrated in the DS-MIRF framework (Tsinaraki & al., 2006).

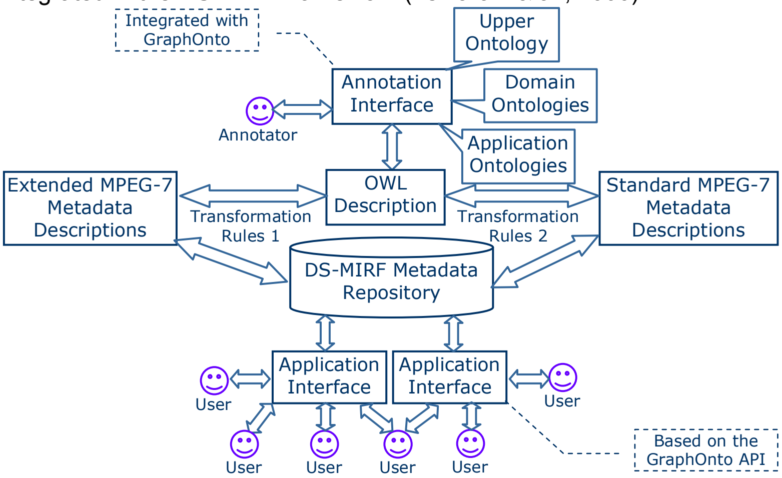

Figure 1: Information Flow in the DS-MIRF Framework

4. The GraphOnto Architecture

We present in this section the GraphOnto architecture (shown in Figure 2) and describe its building blocks. As shown in Figure 2, the following sub-components constitute the GraphOnto architecture:

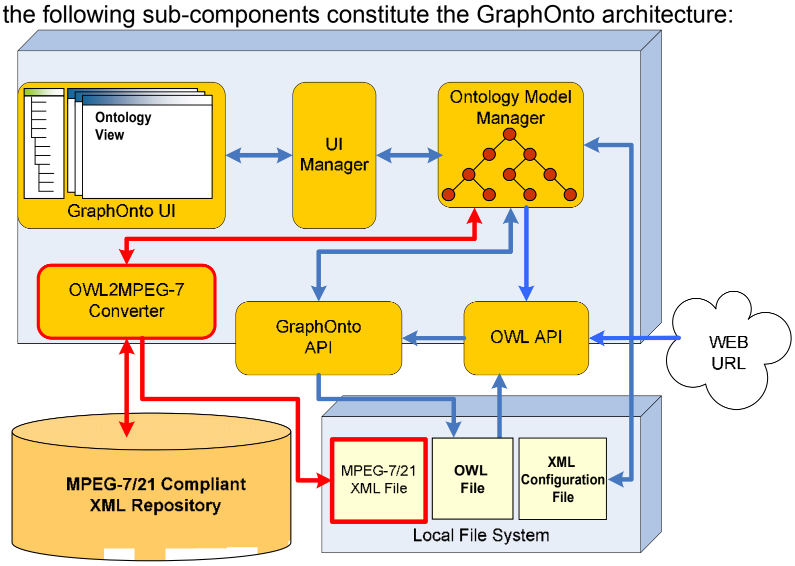

Figure 2: GraphOnto Architecture

- The **GraphOnto User Interface:** It is a user-friendly and simple graphical user interface that interacts with the users and provides them with full editing functionality on OWL-DL ontologies. Multiple ontologies may be loaded at once, and for each of them a different tab is generated. The hierarchies of the Ontology classes, properties and individuals are visualized by tree-like structures in the class, property and individual tabs (inside the ontology tab) respectively and appropriate forms are used to show all the details supported by the OWL constructs. The GraphOnto User Interface is shown in Figure 3.
- The **User Interface Manager:** This component handles the graphical user interface and has a bidirectional role: a) On one hand, it updates the ontology model according to the user input (for example when the user edits or adds a new OWL construct in an ontology); and b) On the other hand, it adapts the user interface according to the model of the currently loaded ontologies. That is, the manager customizes the tree-like views and forms of the user interface so as to agree with the semantics of the OWL elements defined (for example, it alters the ontology class hierarchy if a new class is added or provides forms for the definition of individuals belonging to specific ontology classes).
- The **Ontology Model Manager:** It is responsible for the internal representation of the opened and currently edited ontologies, which are represented using appropriate data structures. It also performs ontology consistency checking and allows the user to modify an ontology only if the OWL rules and semantics are not violated (for example, if a property “A” is of cardinality 1 in class “X”, the user cannot specify a second value for “A” when defining individuals that belong to “X”). The GraphOnto ontology model manager distinguishes upper

from domain ontologies, as specified in the GraphOnto configuration files. The GraphOnto configuration files essentially are the GraphOnto user working environment descriptions, where the upper and domain ontologies to be used are specified.

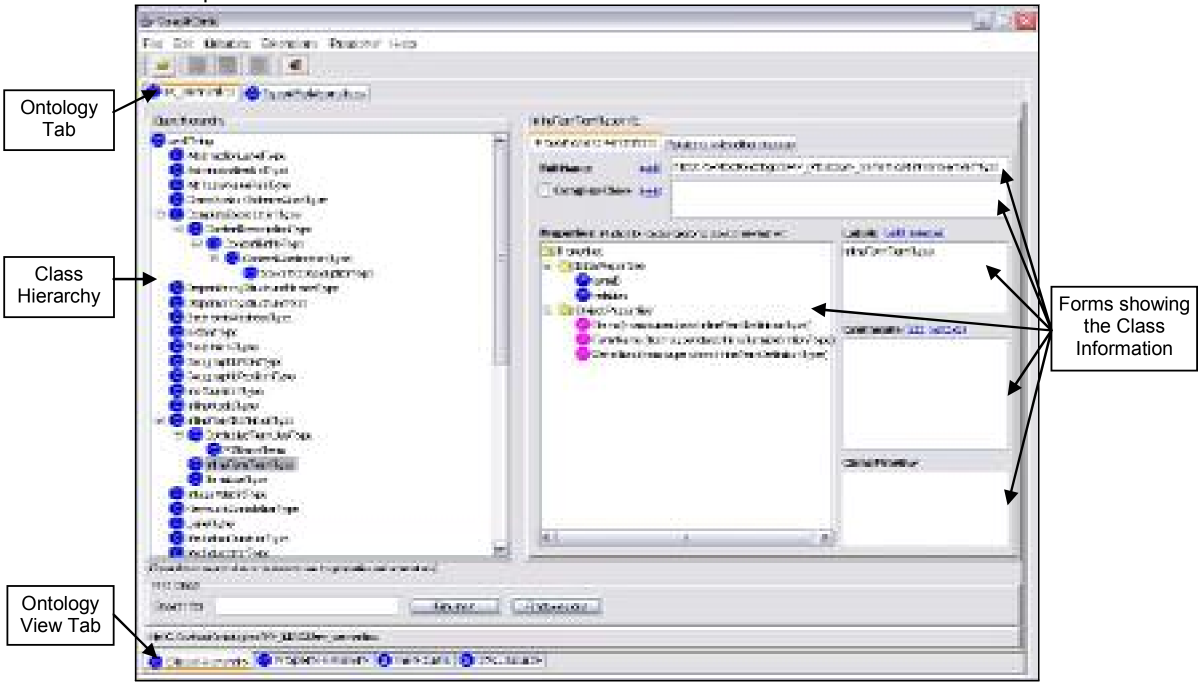

Figure 3: GraphOnto User Interface

- The **WonderWeb OWL API:** It is a public Java API that provides Java classes having the appropriate attributes and methods for reading and parsing OWL ontologies available in the web or stored in local files.
- The **GraphOnto API:** It is a Java API that provides Java classes having the appropriate attributes and methods for fully editing all the OWL constructs. It is based on the OWL API, whose classes are transformed to the corresponding GraphOnto API classes so as to support ontology editing and updating beyond the parsing functionality of the OWL API.
- The **OWL2MPEG7 Converter:** This component allows GraphOnto to be used not only as an ontology editor, but also as a semantic multimedia annotation component in the context of the DS-MIRF framework. Semantic Multimedia annotation is internally treated from an ontological point of view. However, the final output must be in MPEG-7 valid XML format. For this purpose, this component takes on to transform the OWL-based semantics to MPEG-7 based semantics and to store them either into MPEG-7 XML files in the local file system or in an MPEG-7 compliant XML repository.
- The **MPEG-7 Compliant XML Repository:** GraphOnto interoperates with an XML repository that supports the storage of and retrieval MPEG-7 compliant XML descriptions. This repository has been developed on top of the Berkley DB XML and is used both for semantic entity reuse during multimedia content annotation and for semantic description storage.

5. GraphOnto Functionality and Performance Evaluation

We present in this section the functionality of GraphOnto and its performance evaluation, measured against Protégé. The GraphOnto functionality can be viewed in two different contexts: a) as a pure OWL Ontology Editor (see subsection 5.1 for details); and b) as a Semantic Multimedia Annotation tool based on MPEG-7 (see subsection 5.2 for details). The performance evaluation results, presented in subsection 5.3, are clearly in favour of GraphOnto.

5.1. GraphOnto as a Pure OWL Ontology Editor

We present in this subsection GraphOnto as a pure OWL ontology editor. In this context, GraphOnto is a form-based OWL Ontology Editor and can be used to create or modify standalone OWL-DL ontologies. We decided to support OWL-DL, as it provides maximum expressiveness contrary to OWL-Lite and guarantees computational completeness and decidability of reasoning systems contrary to OWL-Full.

GraphOnto fully supports OWL-DL Ontology editing, that is to say the creation of new ontologies, the parsing of existing ones (available in the Web or in local files) and the manipulation (including definition and editing) of classes, properties (datatype and object) and individuals inside these ontologies. Multiple ontologies may be opened and concurrently edited. Browsing in ontologies is supported by four different views: a) The **Class View**, where a tree-like visualization is used to represent the class hierarchy of the currently edited ontology. The Class View is shown in Figure 3, where an OWL ontology capturing the Semantic DS of the MPEG-7 MDS is loaded; b) The **Property View**, where a tree-like visualization is used to represent the class hierarchy of the currently edited ontology. The Property View is shown in Figure 4, where the Semantic DS OWL ontology is loaded; c) The **Individual View**, where a tree-like visualization is used to represent the individuals of the currently edited ontology;

and d) The **OWL Source View**, where the OWL source of the currently edited ontology is shown. The OWL Source View is shown in Figure 5, where the Semantic DS OWL ontology is loaded.

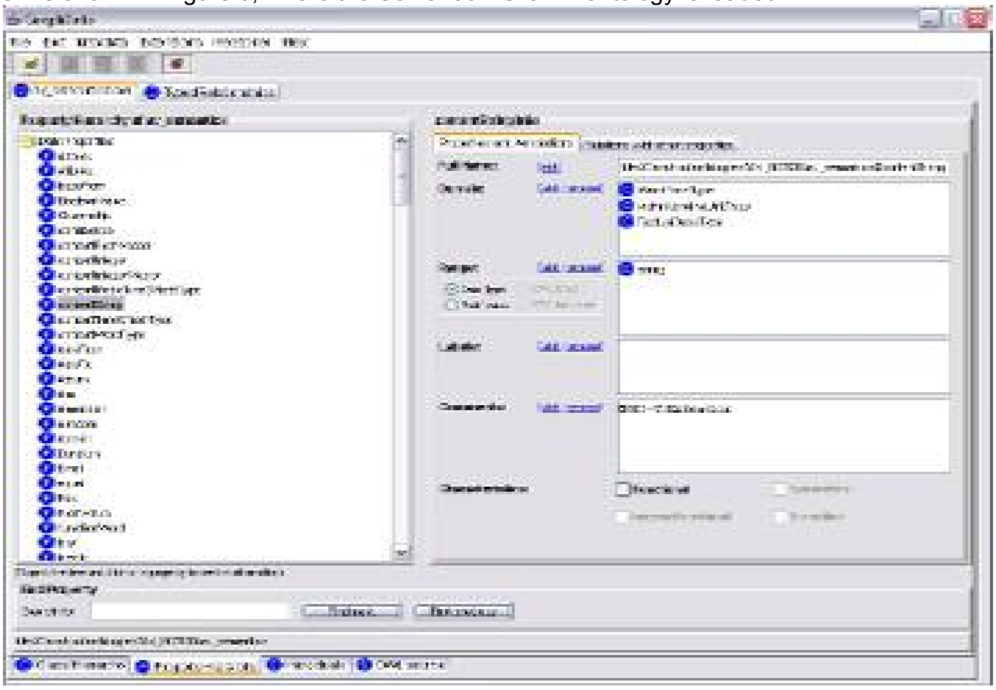

Figure 4: Property Hierarchy View

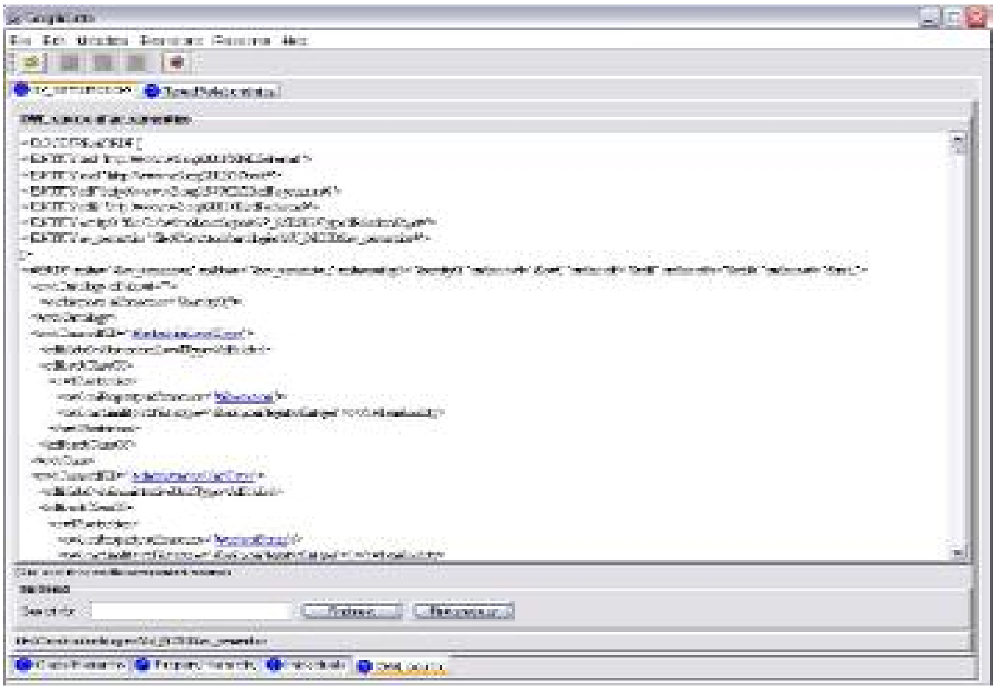

Figure 5: OWL Source View

GraphOnto supports all the functionality specified in OWL-DL on the following OWL constructs:

- **Class:** The user may define an OWL class, edit the rdf:ID of a class, add annotations (labels and comments) to a class, select the values of the (datatype and object) properties of a class, impose (cardinality and value) restrictions on class properties, define relations between classes (including subclass, super class, equivalent class, disjoint class) and define whether a class is complex (in the sense that the class is defined using the intersection, union, complement or enumeration constructs.
- **Datatype property:** The user may define a datatype property, edit the rdf:ID of a datatype property, add annotations to a datatype property, select the datatype property domain(s) among the available OWL classes and the datatype property range(s) (which may be specific datatypes or plain values), specify the datatype property characteristics (including functional) and define relations between datatype properties (including subproperty, super property and equivalent property).
- **Object property:** The user may define an object property, edit the rdf:ID of an object property, add annotations to an object property, select the property domain(s) among the available OWL classes and the property

range(s) (which may be OWL Classes or specific OWL Individuals), specify the property characteristics (including functional, inverse functional, transitive and symmetric) and define relations with other object properties (including subproperty, super property, equivalent property and inverse property).

- **Individual:** The user may define an individual, edit the rdf:ID of an individual, add annotations to an individual, select the individual type among the available OWL classes, define relations between individuals (including identity, difference) and add values to the individual properties.

Ontology consistency is checked during editing and the individuals' property values are automatically filled in case the OWL semantics impose them (for example, if a property is stated to have a fixed value). The edited ontologies may be stored into OWL files.

Semantic Annotation is supported through the definition of individuals outside the scope of an ontology. This is supported by the metadata definition interface of Figure 6, which is automatically generated, based on the ranges and the constraints holding for the properties of the class where the OWL Individual being defined belongs. These newly created OWL Individuals are stored in different OWL files. This feature can be completely personalized, as the user can select which of the classes and the class properties will be instantiated and define custom forms according to which the corresponding individuals are created. These custom forms can be saved as templates in OWL files and re-used in next definitions of individuals, thus allowing personalization of the GraphOnto interface on the application level. The templates available in a user's working environment may be specified in his GraphOnto configuration file.

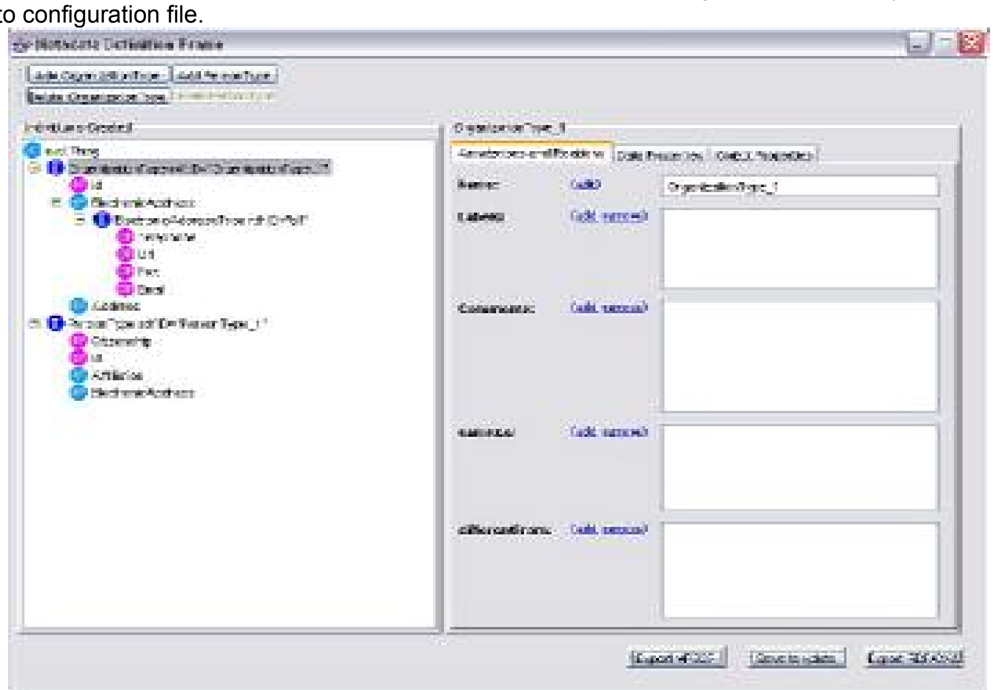

Figure 6: Metadata Definition Frame View

GraphOnto distinguishes Upper and Domain ontologies, which are treated in a different manner. Upper Ontologies are usually standard-based or represent well accepted models. They are parsed and opened only for browsing. Editing is allowed for domain ontologies only. For facilitating and personalizing the user's working environment, GraphOnto utilizes the GraphOnto Configuration files. They are XML files structured according to a simple XML Schema that allows the specification of the GraphOnto user working environment features and contains information about the upper and domain ontologies to be opened and parsed and the templates available to the user. The GraphOnto configuration files allow the users to open multiple ontologies with one click as well as to have available the templates they wish to use. The GraphOnto configuration files provide personalization support both at the user level (assuming the existence of different configuration files for different users) and the task level (the same user may create different configuration files for different tasks).

GraphOnto supports mappings between OWL Ontologies utilizing the following OWL constructs:

- equivalentClass, disjointWith, subClassOf for the definition of mappings between OWL classes
- equivalentProperty, subPropertyOf for the definition of mappings between OWL properties
- sameAs, differentFrom for the definition of mappings between OWL individuals.

The OWL-based mappings allow OWL Ontology personalization as the user can create his ontologies utilizing personalized names for classes, properties and individuals and map them with classes, properties and individuals of existing and well-known ontologies. The personalized ontologies may be utilized during ontology-based query definition: The user defines and uses a personalized ontology, where the individuals, the classes and the properties are mapped to the corresponding constructs of the ontology supported by a retrieval system and the user queries are translated according to the later ontology.

5.2. GraphOnto as a Semantic Multimedia Annotation Component

In addition to its pure OWL editor functionality, GraphOnto may be also used as a semantic multimedia annotation component in the context of the DS-MIRF Framework, as described in this subsection. The functionality of distinguishing Upper and Domain ontologies is applied in the DS-MIRF Ontological infrastructure which contains an Upper Ontology capturing the MPEG 7 MDS, an application ontology capturing the Typed Relations of MPEG-7 and domain ontologies describing the domains of soccer and F1. These ontologies are easily opened at once using the Multimedia GraphOnto Configuration File that contains the DS-MIRF working environment specification. GraphOnto is also used for the further extension of the DS-MIRF ontological infrastructure with additional domain ontologies.

GraphOnto implements the DS-MIRF transformation rules from OWL to MPEG-7 compliant XML syntax. These transformations may be applied to domain ontologies integrated in the DS-MIRF framework, extending this way the semantics of the Upper Ontology and the Typed Relations application ontology. In particular:

- OWL Domain Ontologies are transformed into MPEG-7 Abstract Semantic Entity Hierarchies. The OWL domain ontology classes and individuals are represented as MPEG-7 semantic elements of type “SemanticBaseType”. The “AbstractionLevel” element of the “SemanticBaseType” and the MPEG-7 semantic relationships are used to capture ontology semantics. “AbstractionLevel” has the “dimension” attribute, of non-negative integer type, which denotes if a semantic element is abstract and represents a class (when “AbstractionLevel.dimension”>0) or is non-abstract and describes an individual (when “AbstractionLevel” is absent or “AbstractionLevel.dimension”=0). An abstract semantic entity that represents a domain ontology class is related with each of its subclasses through a pair of “Relation” elements of type “generalizes”/“specializes”. In addition, an abstract semantic entity that represents a domain ontology class is related with each of the semantic entities representing the class individuals through pairs of “exemplifies”/“exemplifiedBy” relationships. The properties defined in the domain ontology classes are transformed into “Property” elements (if they are of simple type) or into pairs of “property”/“propertyOf” relationships that associate the semantic entities representing the classes and the property values (if they are of complex type).

  As an example, consider a OWL soccer ontology, where the “PlayerObject” class represents the soccer players and extends the MPEG-7 MDS class “AgentObject” (which represents persons, person groups and organizations). In this ontology, the goalkeepers are represented by the “GoalKeeperObject” class, subclass of “PlayerObject”, which is shown in Figure 7.

```
<owl:Class rdf:ID="GoalKeeperObject">
  <rdfs:subClassOf rdf:resource="#PlayerObject"/>
  <rdfs:label>GoalKeeper Object</rdfs:label>
</owl:Class>
```

Figure 7: OWL Definition of the “GoalKeeperObject” Class

  If the DS-MIRF transformation rules are applied in the above-referred soccer ontology, the “GoalKeeperObject” class of Figure 7, is transformed into the MPEG-7 abstract semantic entity shown in Figure 8.

```
<SemanticBase xsi:type="AgentObjectType" id="GoalKeeperObject ">
  <AbstractionLevel dimension="1"/>
  <Label><Name>GoalKeeper Object</Name></Label>
  <Relation type="specializes" source="#GoalKeeperObject" target="#PlayerObject"/>
  <Relation type="generalizes" source="#PlayerObject" target="#GoalKeeperObject"/>
</SemanticBase>
```

Figure 8: MPEG-7 Abstract Semantic Entity resulting from the transformation of the “GoalKeeperObject” Class of Figure 7

- Individuals defined outside the scope of an ontology are transformed into MPEG-7 metadata descriptions that can be used for annotate multimedia content. The individuals representing MPEG-7 constructs are transformed into XML elements, the object properties of which are transformed into elements and the datatype properties are transformed into the constructs they represent (attributes, elements or simple values).

During the transformation, in order to produce valid MPEG-7 descriptions, information regarding the MPEG-7 XML element order and the MPEG-7 datatype property representation is needed. This information is kept in a transformation rule ontology and is utilized during both ontology and individual transformations.

The produced MPEG-7 Descriptions may be stored either in MPEG-7 compliant XML files or into the DS-MIRF MPEG-7 Repository. The MPEG-7 repository may also be used for retrieving existing MPEG-7 semantic entity descriptions during annotation.

The transformations from OWL/RDF to MPEG-7 allow full compatibility with MPEG-7 so that all tools and applications that use MPEG-7/21 still work transparently with the produced MPEG-7 metadata.

5.3. Performance of GraphOnto vs. Protégé

According to subsection 5.1, GraphOnto is a complete graphical OWL ontology editor that provides functionality also supported by existing ontology editors, among which Protégé is the most advanced, as well as additional important functionality. In order to compare the performance of GraphOnto with the one of Protégé, we performed a set of experiments on the most time-consuming operation: The parsing of existing ontologies.

Our evaluation metrics are the parsing time and the amount of main memory used during parsing. We have selected three OWL ontologies for experimentation:

- A travel ontology, comprised of 6 classes, 13 datatype properties, 8 object properties and 11 individuals.
- An ontology capturing the Semantic DS of the MPEG-7 MDS, comprised of 97 classes, 79 datatype properties, 126 object properties and 0 individuals
- A food ontology, comprised of 63 classes, 0 datatype properties, 4 object properties and 45 individuals.

The evaluation took place in a Pentium IV at 2,8 GHz, with 512 MB of main memory, under the MS Windows XP operating system. The evaluation results, shown in Table 1, show that GraphOnto outperforms Protégé in all the three ontologies and for both the evaluation metrics: GraphOnto uses almost 4/7 of the memory used by Protégé for the same ontology, and the parsing takes always less than half the time taken by Protégé (especially for the travel ontology, the parsing time is less than 1/3 of the time taken by Protégé). The main reason for these results is that Protégé has been developed around a frame-based ontology representation approach, and the OWL ontology manipulation is performed by the OWL plug-in of Protégé, while the GraphOnto development has been based on OWL.

Table 1: Performance Evaluation Results between GraphOnto and Protégé in Ontology Parsing

Ontology Name	GraphOnto Parsing Time	Protégé Parsing Time	GraphOnto Memory Usage	Protégé Memory Usage
Travel	1,47 seconds	6,80 seconds	40240 Kbytes	71540 Kbytes
AVSemantics	3,29 seconds	7,88 seconds	45384 Kbytes	72984 Kbytes
Food	3,33 seconds	8,35 seconds	44388 Kbytes	76104 Kbytes

6. Conclusions – Future Work

We have presented GraphOnto, a software component that facilitates the definition and editing of both standard-based and domain ontologies and their use in multimedia information system components. It is based on OWL, the dominant ontology definition language, and can be viewed as both a complete and generic graphical OWL ontology editor and as a semantic annotation component for multimedia content.

As a pure OWL ontology editor, GraphOnto provides functionality supported by existing ontology editors, among which Protégé is the most advanced, as well as several important additional features: Multiple ontology management, OWL Ontology mappings that allow the definition of personalized ontologies expressed using OWL constructs, and personalization of the GraphOnto user interface on the application, task and user levels. In addition, GraphOnto has been shown to have, in terms of main memory utilization and parsing time, better performance than Protégé in OWL ontology parsing, which is the most time-consuming ontology-related operation.

As a semantic annotation component for multimedia content, GraphOnto is used in the context of the DS-MIRF framework, utilizing and extending the DS-MIRF ontological infrastructure. In particular, GraphOnto allows the semantic annotation of multimedia objects utilizing the DS-MIRF upper ontology, which fully captures the MPEG-7 MDS. In addition, GraphOnto allows the utilization and definition of OWL domain ontologies integrated in the DS-MIRF framework, which extend the upper ontology with domain knowledge. Both the domain ontologies and the semantic multimedia annotations defined using them are transformed, using the DS-MIRF transformation rules, to semantic MPEG-7 multimedia content descriptions. The semantic MPEG-7 descriptions may be stored either in MPEG-7 compliant XML files or in the DS-MIRF MPEG-7 Repository. In addition, the GraphOnto API is used for the development of ontology-based retrieval and filtering application interfaces in the DS-MIRF framework.

Our future research includes the investigation of the use of GraphOnto in multi-user environments and the development of application interfaces in the DS-MIRF framework based on the GraphOnto API, like for example an Ontology-based Query Formulation Tool, as well as the integration of the GraphOnto ontology editor with semantic multimedia content annotation applications, like a multimedia content segmentation component.

7. References

1. D. Brickley and R. V. Guha (eds.) (2004). RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, 10 Feb. 2004. (http://www.w3.org/TR/rdf-schema).
2. S.F. Chang, T. Sikora, A. Puri (2001). Overview of the MPEG-7 standard. IEEE Transactions on Circuits and Systems for Video Technology 11(2001), pp. 688–695.
3. D. Fallside (2001). XML Schema Part 0: Primer. W3C Recommendation, 2001. (http://www.w3.org/TR/xmlschema-0/).
4. ISO/IEC: 15938-5:2003 (2003). Information Technology – Multimedia content description interface – Part 5: Multimedia description schemes. 2003, First Edition.
5. A. Karanastasi, S. Christodoulakis (2005). OntoNL: An Ontology-based Natural Language Interface Generator for Multimedia Repositories. AvivDiLib 2005, pp. 205-216.
6. F. Manola and Eric Milles (eds.) (2004). RDF Primer. W3C Recommendation. 10 Feb. 2004. (http://www.w3.org/TR/rdf-primer).
7. D. L. McGuinness and F. van Harmelen (eds.) (2004). OWL Web Ontology Language: Overview. W3C Recommendation, 10 Feb. 2004. (http://www.w3.org/TR/owl-features).
8. The Protege site, http://protege.stanford.edu/.
9. C. Tsinaraki, A. Perego, P. Polydoros, A. Syntzanaki, A. Martin, S. Christodoulakis (2006). Semantic, Constraint & Preference Based Authoring of Multi-topic Multimedia Presentations. Workshop on Multimedia Semantics (WMS), Chania, Crete, June 2006.
10. C. Tsinaraki, P. Polydoros, S. Christodoulakis (2004b). Integration of OWL ontologies in MPEG-7 and TV-Anytime compliant Semantic Indexing". 16th International Conference on Advanced Information Systems Engineering (CAISE), pp. 398-413, June 2004, Riga, Latvia.
11. C. Tsinaraki, P. Polydoros, S. Christodoulakis (2004a). Interoperability support for Ontology-based Video Retrieval Applications. Conference on Image and Video Retrieval (CIVR) 2004, pp. 582-591, July 2004, Dublin, Ireland.