Discovery-Driven Ontology Evolution

Silvana Castano, Alfio Ferrara, Guillermo N. Hess
Dipartamento di Informatica e Comunicazione
Università degli Studi di Milano
Milan, Italy 20135
Email: {castano,ferrara,hess}@dico.unimi.it
Telephone: (39) 0250316319
Fax: (39) 0250316229

***Abstract*— In this paper, we present a methodology for ontology evolution, by focusing on the specific case of multimedia ontology evolution. In particular, we discuss the situation where the ontology needs to be enriched because it does not contain any concept that could be used to explain a new multimedia resource. The paper shows how ontology matching techniques can be used to enforce the discovery of new relevant concepts by probing external knowledge sources using both the information available in the multimedia resource and the knowledge contained in the current version of the ontology.**

I. Introduction

Ontology evolution regards the capability of managing the modification of an ontology in a consistent way. An ontology may change because the domain or the user needs have changed [16] or simply because the shared conceptualization (i.e., the perspective) has been modified [13]. According to [11], we can define ontology evolution as the timely adaptation of the ontology to changing requirements and the consistent propagation of changes to the dependent artifacts. The problem of ontology evolution can be considered also as a special case of the more general and well studied problem of belief change as described in [5]. In this respect, some of the most important concepts of the belief change literature have been revised in order to apply them to the ontology evolution context. A six-phase evolution process for ontology evolution has been proposed in [15] which is generally recognized as a comprehensive reference methodology capable of handling the evolution of multiple ontologies.

In this paper, we focus on a specific case of evolution, which is multimedia ontology evolution. In our approach, the ontology is used to provide an interpretation for the elements extracted from multimedia resources, such as images, textual documents, video and audio. Our approach is defined in the framework of the BOEMIE EU Project [1], where the reader can find further details about the semantic information extraction from multimedia resources as well as the whole evolution methodology. In the paper, we focus on the use of ontology matching techniques to support ontology enrichment when new knowledge is required to find an interpretation for a new resource.

The paper is organized as follows. In Section II we describe the requirements for ontology evolution. The methodology we propose for multimedia ontology evolution is presented in Section III. A discussion of the activity of concept discovery, together with an example is provided in Section IV. Section V presents the related work in the field of ontology evolution. Conclusions and future work are discussed in Section VI.

II. Requirements for the ontology evolution

In this paper, we address ontology evolution in a scenario where:

- Multimedia resources are considered, which have been already submitted to a semantic extraction process for associating them with appropriate metadata that describe their contents. Consequently, the ontology evolution is triggered by incoming multimedia resources with associated metadata information.
- An initial domain ontology is available, providing a description of the domain of interest (in the worst case, also a high-level, poorly detailed description). Such an ontology is used also by the extraction process and will be continuously evolved so that the new ontology version is used as input for subsequent extraction and evolution activities, according to a bootstrapping process which is iterated until a satisfactory quality of multimedia resource classification is reached. In the paper, we will consider the Athletics domain and we will discuss ontology evolution by providing some examples taken from this domain.
- An open system approach is adopted, in that when background knowledge in the underlying ontology is not sufficient to interpret new incoming multimedia resources, a concept discovery activity is triggered to acquire new knowledge from other external knowledge sources. This way, we try to limit the human involvement of the ontology expert in the definition of new concepts, which is usually a manual and time consuming activity in ontology evolution [16].

Before introducing our evolution methodology, we introduce some basic definition used throughout the paper:

- A **multimedia resource** is any kind of multimedia object (e.g., image, video, sound) from which information is extracted.
- Each individual extracted element from a multimedia resource is called a **media object** (e.g., a portion of an image).

- With each media object some **metadata** are associated, which represent the semantic information that is extracted from the resource.
- A **simple concept** is a concept in the domain ontology which explains a media object in terms of metadata.
- An **aggregate concept** is a concept in the domain ontology which is defined as an aggregation of simple concepts, such as, for instance, an event. Furthermore, an aggregate concept is an explanation of the multimedia resource, that is an interpretation of the real event or object described in the resource.

To illustrate these concepts, we consider an evolution scenario in the Athletics domain characterized by the multimedia resource shown in Fig. 1. In this case, four media objects (marked in the figure) are identified during semantic extraction which have associated metadata stating that: (1) is a foam mattress, (2) is a pole, (3) is an horizontal bar, and (4) is an athlete, respectively. Moreover, with respect to the event (i.e., aggregate concept) described by the image, we can recognize that all these media objects are part of a bigger picture, which describes a pole vault event.

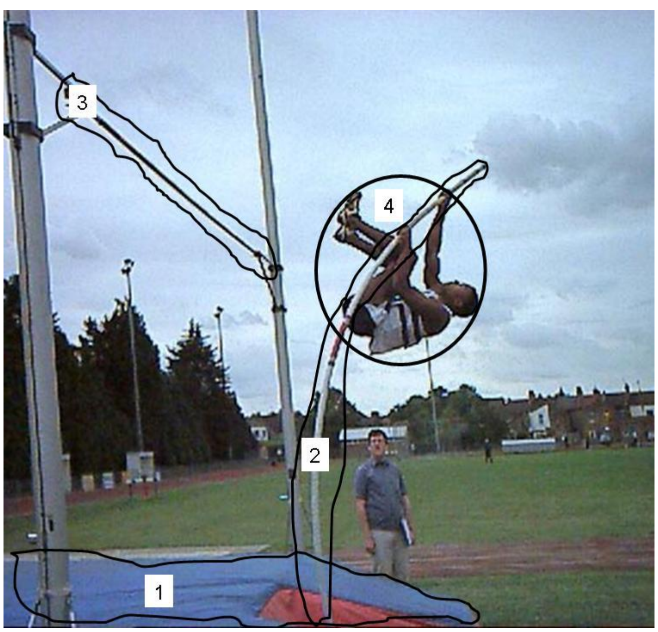


Fig. 1. An example of multimedia resource

The metadata associated with the media objects of Fig. 1 are shown in Table I.

TABLE I
METADATA ASSOCIATED WITH THE MEDIA OBJECTS OF THE MULTIMEDIA RESOURCE OF FIG. 1

Foam_mat	$foam\_mat_1$
Pole	$pole_1$
Horizontal_bar	$horizontal\_bar_1$
Athlete	$athlete_1$

With respect to this example, the domain ontology contains the aggregate concept Pole_vault, while Horizontal_bar, Foam_mat, Athlete and Pole are simple concepts. The activity of classifying a new multimedia resource with respect to the ontology is, in this context, the activity of finding an aggregate concept in the ontology capable of explaining the set of simple concepts detected in the multimedia resource semantic extraction. Such an activity is called *interpretation* of the multimedia resource, which can be performed by exploiting non-standard reasoning techniques [9]. However, not always the interpretation process succeeds in determining an explanation for the resource in the ontology. In particular, depending on the background ontology knowledge and on the metadata information for the incoming resource, four cases are possible: i) one single aggregate concept explaining the multimedia resource semantics is found in the ontology; ii) more than one aggregate concept explaining the multimedia resource semantics are found in the ontology; iii) no aggregate concepts are found in the ontology capable of explaining the semantics of the multimedia resource, but all the simple concepts extracted from the resource are already defined in the ontology; iv) no aggregate concepts are found in the ontology capable of explaining the semantics of the multimedia resource, and one or more simple concepts are missing in the ontology too.

These four scenarios produce two main classes of evolution patterns, namely population and enrichment, respectively (see Fig. 2). In the following we briefly describe these two classes of patterns.

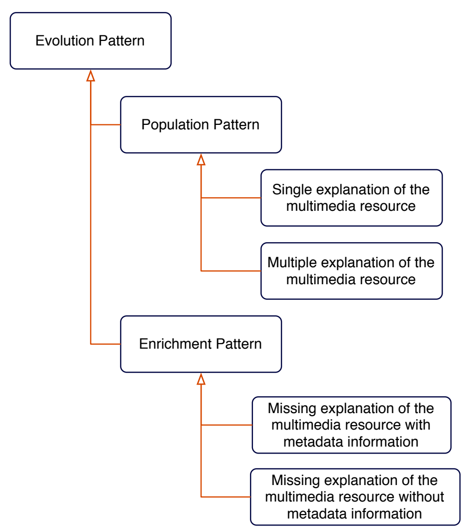


Fig. 2. The evolution patterns

- **Population patterns**: the domain ontology contains the aggregate concepts as well as all the simple concepts described by the media objects metadata extracted from the multimedia resource. In this case, only ontology population has to be performed. Ontology population is the activity of introducing new instances in the ontology to describe the multimedia resource with respect to the aggregate concepts already present in the ontology.

- **Enrichment patterns:** The ontology does not contain concepts that can be used to explain the multimedia resource. In this case, the ontology has to be enriched. Ontology enrichment is the activity of defining new concepts, properties, and/or semantic relations in the ontology. Basically, the idea behind our approach to ontology enrichment is to exploit the metadata available in the multimedia resource description in order to discover new aggregate concepts in external knowledge sources (e.g., other ontologies, web directories, lexical systems) that can be added to the domain ontology in order to explain the new multimedia resource. This activity is referred as *concept discovery* and it is executed by exploiting ontology matching techniques.

In this paper, we focus on the enrichment patterns and, in particular, we describe the case when all the simple concepts are already stored in the domain ontology, but the aggregate concept is missing, in order to show how the ontology can be enriched by the concept discovery activity.

III. Methodology for discovery-driven ontology evolution

In the evolution scenario discussed in the paper, the interpretation of a multimedia resource produces a description of the simple concept instances that are detected into the resource (e.g., the objects that appear into an image) but no aggregate concept (e.g., events) in the ontology can be found that explain the resource itself. In this section, we describe a methodology for ontology evolution in this scenario and the role of ontology matching techniques in supporting the identified evolution phases.

A. The evolution phases

The goal of ontology evolution is to augment the background knowledge in the existing domain ontology to better classify extracted object descriptions. The ontology is enriched by adding concepts, properties, and/or semantic relations or by modifying existing ones, in order to produce a new ontology version capable of providing interpretation of the new multimedia resource.

In order to detect the ontology modifications required for explaining the new resources, we exploit the description of the media objects extracted from the resources. This means that, different from the usual evolution scenarios, in our case we know from the interpretation phase that an aggregate concept is missing in the domain ontology, but we do not know exactly which one. We have the instance of the missing concept and have to create an aggregate concept which explains it.

The evolution process is articulated in three phases, which are explained in the following.

Temporary concept definition: the incoming instance and its associated metadata information is used to define a temporary concept c_t, which is matched against the domain ontology to detect semantically related concepts already present in the ontology. This internal concept detection activity is optional and aims at finding in the domain ontology some aggregate concepts which have semantic affinity with c_t to be used as the starting basis in the next phase.

Concept discovery: if internal concepts have been detected in the previous phase, for each retrieved matching concept c_i in the ontology (i.e., c_i is semantically similar to the temporary concept c_t) a *probe query* is created. Otherwise, the ontology expert has to define the probe query manually, by exploiting his own domain knowledge and the information available in the definition of the temporary concept c_t. A probe query is a description of the properties, the constraints, and the semantic relations that should be exhibited by the (actually missing) new concept in order to consider it as a candidate for ontology enrichment. The probe query is then matched against one or more external knowledge sources to discover some external concepts that could be useful to define the new concept to be inserted in the ontology.

Ontology enrichment: this phase comprises the definition and insertion of the new (aggregate) concept in the domain ontology and also the validation of the resulting ontology. The definition of the new concept is interactively performed by the ontology expert, by exploiting the definitions of the temporary concept c_t and of the external matching concepts retrieved during the concept discovery phase. Once the new aggregate concept has been defined, the ontology expert inserts it in a proper position in the ontology taxonomy. The task of finding the most suitable placement of the new concept in the ontology taxonomy is performed by combining ontology matching techniques and ontology reasoning, as described in [2]. Finally, the evolved ontology is validated by means of standard reasoning techniques [6].

B. Ontology matching for concept discovery

As described in the previous section, ontology matching techniques play an essential role in the ontology evolution process. In temporary concept definition, the temporary concept can be compared against the existing ontology to find internal matching concepts that can help the probe query formulation activity. In concept discovery, a probe query is compared against external knowledge sources with the goal of finding potentially relevant knowledge. To perform these concept comparisons, we rely on our semantic matchmaker H-Match developed in the framework of the Helios project for matching independent peer ontologies [3]. H-Match takes a target concept description c and an ontology O as input and returns the concepts in O which match c, namely the concepts with the same or the closest intended meaning of c. In H-Match we perform concept matching through affinity metrics by determining a measure of semantic affinity in the range $[0, 1]$. A threshold-based mechanism is enforced to set the minimum level of semantic affinity required to consider two concepts as matching concepts. Given two concepts c and c', H-Match calculates a semantic affinity value $SA(c, c')$ as the linear combination of a linguistic affinity value $LA(c, c')$ and a contextual affinity value $CA(c, c')$. The linguistic affinity function of H-MATCH provides a measure of similarity between two ontology concepts c and c' computed on the basis of their linguistic features (i.e., concept names). For the linguistic affinity evaluation, H-MATCH relies on a thesaurus of terms and terminological relationships automatically extracted from the WordNet lexical system [12]. The contextual affinity function of H-MATCH provides a measure of similarity by taking into account the contextual features of the ontology concepts c and c'. The context of a concept can include properties, semantic relations with other concepts, and property values. The context can be differently composed to consider different levels of semantic complexity, and four matching models, namely, *surface*, *shallow*, *deep*, and *intensive*, are defined to this end. In the surface matching, only the linguistic affinity between the concept names of c and c' is considered to determine concept similarity. In the shallow, deep, and intensive matching, also contextual affinity is taken into account to determine concept similarity. In particular, the shallow matching computes the contextual affinity by considering the context of c and c' as composed only by their properties. Deep and intensive matching extend the depth of concept context for the contextual affinity evaluation of c and c', by considering also semantic relations with other concepts (deep matching model) as well as property values (intensive matching model), respectively. The comprehensive semantic affinity $SA(c,c')$ is evaluated as the weighted sum of the Linguistic Affinity value and the Contextual Affinity value, that is:

$$SA(c,c') = W_{LA} \cdot LA(c,c') + (1 - W_{LA}) \cdot CA(c,c') \quad (1)$$

where W_{LA} is a weight expressing the relevance to be given for the linguistic affinity in the semantic affinity evaluation process. A detailed description of H-MATCH and related matching models is provided in [3].

IV. CONCEPT DISCOVERY AND ONTOLOGY ENRICHMENT

In this section, we describe in more detail how concept discovery and ontology enrichment are performed, with respect to the jumping example of Fig. 1.

A. Concept discovery

The concept discovery activity is articulated in the following steps:

1) Given a multimedia resource r and the set M of media objects that have been extracted from r, we build a temporary target concept c_t. The context of c_t (i.e., the set of properties, restrictions, and semantic relations featuring c_t) is composed by exploiting the metadata associated with M, that is the set of simple concepts S explaining objects in M, that is,

$$S = \{c_i \in O \mid \forall m_i \in M, m_i : c_i\}$$

where O denotes the ontology.

2) We use H-MATCH to match c_t against the existing domain ontology. Since c_t is not featured by a name, H-MATCH is executed by exploiting only the contextual affinity matching techniques. In fact, contextual matching considers only the contextual affinity between the concepts to be compared without considering their names. The goal of this activity is to exploit the concepts already inside the ontology which are similar to c_t, and which are featured by a name, to trigger the concept discovery phase. In other terms, we know that the missing concept is similar to something that is already known in the ontology, and we exploit this similarity between the ontology (what we know) and the external knowledge sources (what is known in other sources) in order to learn new concepts to be added to the ontology.

3) For each concept c_i whose affinity with c_t is higher than a threshold t, we build a probe query q_i. Each probe query q_i is matched against external knowledge sources using H-MATCH. The result is a set of external candidate concepts returned by the matching process whose semantic affinity is higher than or equal to a given matching threshold. Retrieved concepts are then made available to the ontology expert, who can exploit them to define the new concept to be added to the ontology.

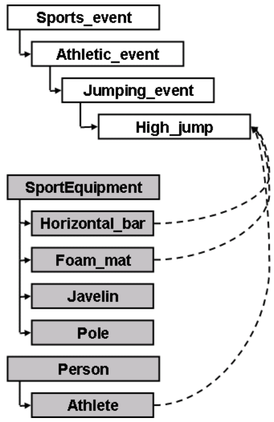
```
function conceptDiscovery(Extracted_Metadata){
  Vector probeQueries = null;
  Vector candidateConcepts = null;
  TemporaryConcept ct = createConcept(Extracted_Metadata);
  foreach Concept c in Ontology{
    if(hmatch.contextualMatch(ct,c) >= threshold){
      probeQueries.add(c);
    }
   foreach ProbeQuery q in probeQueries{
     foreach Concept e in ExternalKnowledgeSource{
       if(hmatch.match(q,e) >= threshold){
         candidateConcepts.add(e);
       }
     }
   }
  }
  return candidateConcepts;
}
```

Fig. 3. Concept discovery algorithm

In order to discuss an example of concept discovery, consider the multimedia resource shown in Fig. 1 and the media objects marked in the figure. Then suppose that the domain ontology contains information about jumping events, as defined in Fig. 4. The concepts graphically represented by a white box (i.e., Sport_event, Athletic_event, Jumping_event and High_jump) represent aggregate concepts, while gray boxes (e.g., Foam_mat, Pole, Horizontal_bar, Athlete) represent simple concepts. Composition dependencies among simple and aggregate concepts are represented by means of dashed arrows. In the right side of the figure, the corresponding Description Logics specification is reported.

The first step of concept discovery is the creation of the temporary concept c_t, starting from the incoming resource information, as shown in Fig. 5.

The definition of c_t is based on the analysis of the metadata associated with the multimedia resource, that are shown in



$$
\begin{array}{rcl}
Athlete & \equiv & Person \sqcap \exists hasProfession.Sport \\
Foam\_mat & \sqsubseteq & SportEquipment \\
Pole & \sqsubseteq & SportEquipment \\
Javelin & \sqsubseteq & SportEquipment \\
Horizontal\_bar & \sqsubseteq & SportEquipment \\
Jumping\_event & \sqsubseteq & Event \\
High\_Jump & \sqsubseteq & Jumping\_event \sqcap \\
 & & \exists hasPart.Jumper \sqcap \\
 & & \exists hasPart.Horizontal\_bar \sqcap \\
 & & \exists hasPart.Foam\_mat
\end{array}
$$

Fig. 4. The domain ontology before evolution

$$
\begin{array}{rcl}
c_t & \sqsubseteq & \exists hasPart.Jumper \sqcap \\
 & & \exists hasPart.Horizontal\_bar \sqcap \\
 & & \exists hasPart.Foam\_mat \sqcap \\
 & & \exists hasPart.Pole
\end{array}
$$

Fig. 5. The temporary concept c_t definition from the metadata of Table I

Table I. In particular, since the media object retrieved in the resource are considered as components of an unknown concepts that should be added in the ontology, we define on c_t a $hasPart$ restriction for each simple concept of Table I. The temporary concept c_t describes the features (in terms of restrictions) of the required target concept. However, we do not have neither a name for c_t nor any semantic relation among it and other concepts in the ontology. Because of this, if we match c_t against the external knowledge source, we could not find relevant candidate concepts. In order to address this problem, we enrich the definition of c_t by exploiting the name and the context of other concepts, if any, in the ontology that are semantically similar to c_t. Since a name is not available for c_t, only the contextual matching is performed. In the case of the example, H-MATCH evaluates the following semantic affinity values in the ontology of Fig. 4: $SA(\mathsf{Athlete}, c_t) = 0.45$, and $SA(\mathsf{High\_Jump}, c_t) = 0.97$. Using the H-MATCH default threshold of 0.5, only the High_jump concept is actually selected for probe query composition. Having the High_jump concept retrieved, we now create one probe query for High_jump (i.e., q_1) to be compared against external knowledge sources. In our example, we use the Google and the Yahoo sports directories as external knowledge sources. The probe query q_1 contains a description of High_jump, which is composed by the name of the concept and by its context, i.e., by the restrictions defined on it and by its super-classes and subclasses. The result of matching q_1 against the Google and Yahoo external sources is shown if Fig. 6, where matching concepts are shown together with corresponding semantic affinity values.

All the 11 candidates concepts shown in Fig. 6 are returned to the ontology expert. High_jump is discarded, since it is already present in the domain ontology.

B. Ontology enrichment

Based on the matching concepts definitions retrieved during the concept discovery phase, on the incoming temporary concept c_t and on the domain ontology, the ontology expert begins the concept definition activity for ontology enrichment. A new aggregate concept is defined using the simple concepts available in the domain ontology. If some simple concept is needed which is not yet described in the ontology, it is also defined. In the example, the ontology expert chooses to add the Pole_Vault concept, on the basis of the information provided by the two web sources and on the fact that Pole_Vault is retrieved in both the external sources. Concept definition is a manual activity, in that the ontology expert must specify all the properties and restrictions of the new concept. In our approach, the amount of manual activity is reduced in that the ontology expert works directly on the concepts specifications retrieved during the discovery activity, by properly integrating/merging those considered more relevant. Once the new concept is defined, the ontology expert inserts it into the ontology, which means that the appropriate location in the ontology hierarchy has to be found. To this end, the ontology expert may use H-MATCH to locate a possible insertion point for the new concept and a reasoner to validate the new resulting version of the ontology [2]. Fig. 7 show the definition of the new aggregate concept Pole_vault from our example.

V. RELATED WORK

Ontology evolution research work has mainly focused on the problem of evaluating the impact of requirement changes on the ontology contents [8]. In [10], the authors present a framework for ontology evolution and change management based on an ontology of change operations with the aim of providing a formal description of the ontology modifications required to perform a given evolution task. The ontology of change operations is defined for the OWL knowledge model and contains basic change operations and complex change operations. A basic operation describes the procedure of modifying only one specific feature of the OWL knowledge model (e.g., type and cardinality restriction change), while a complex operation describes an articulated change procedure and is

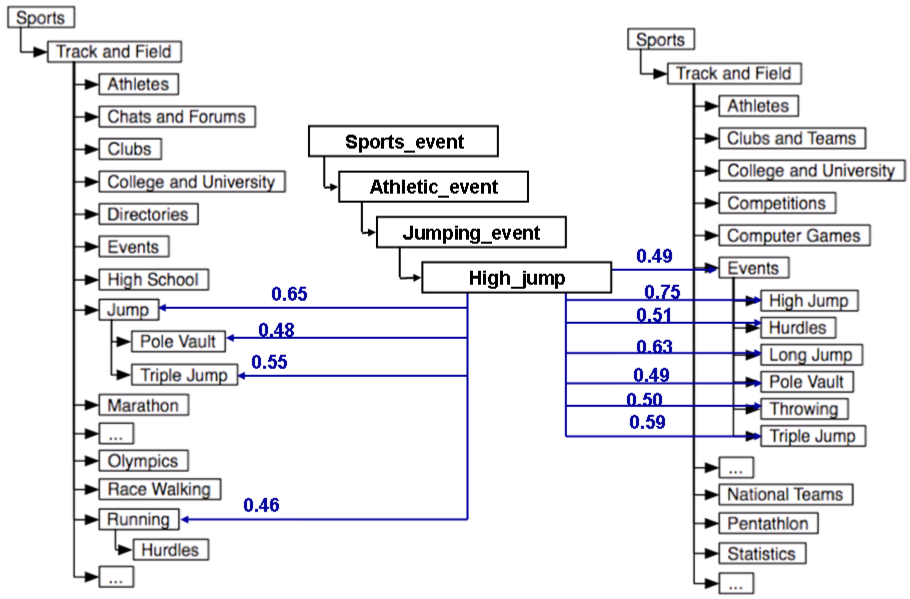
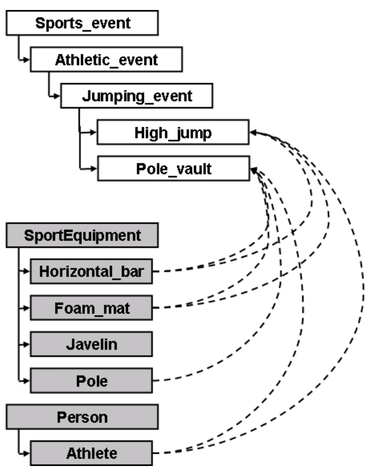


Fig. 6. Sample of probe query evaluation



$$
\begin{aligned}
Athlete &\equiv Person \sqcap \exists hasProfession.Sport \\
Foam\_mat &\sqsubseteq SportEquipment \\
Pole &\sqsubseteq SportEquipment \\
Javelin &\sqsubseteq SportEquipment \\
Horizontal\_bar &\sqsubseteq SportEquipment \\
Jumping\_event &\sqsubseteq Event \\
High\_Jump &\sqsubseteq Jumping\_event \sqcap \\
&\quad \exists hasPart.Jumper \sqcap \\
&\quad \exists hasPart.Horizontal\_bar \sqcap \\
&\quad \exists hasPart.Foam\_mat \\
Pole\_vault &\sqsubseteq Jumping\_event \sqcap \\
&\quad \exists hasPart.Jumper \sqcap \\
&\quad \exists hasPart.Horizontal\_bar \sqcap \\
&\quad \exists hasPart.Foam\_mat \\
&\quad \exists hasPart.Pole
\end{aligned}
$$

Fig. 7. The domain ontology after evolution

composed of multiple basic operations. With respect to these classification of changes, the requirements that an ontology management tool should address for ontology evolution are discussed in [17]. In particular, the authors emphasize that such a tool should provide a set of evolution operations according to the supported ontology models (functional requirement) and that changes should be discovered semi-automatically by analyzing user behavior (refinement requirement). During the evolution process, the tool has to reflect the user preferences (user supervision requirement) by providing advanced facilities, such as change-visualization and inconsistency detection (transparency and usability requirements). Moreover, history of changes needs to be supported to eventually undo any change applied to the ontology (auditing and reversibility requirements). The recent success of distributed and dynamic infrastructures for knowledge sharing has raised the need of semiautomatic/automatic ontology evolution strategies. An overview of some proposed approaches in this direction is presented in [4], even if limited concrete results have appeared in the literature. In most recent work, formal and logic-based approaches to ontology evolution are also being proposed. In [7], the authors provide a formal model for handling the semantics of change phase embedded in the evolution process of an OWL ontology. The proposed formalization allows to define and to preserve arbitrary consistency conditions (i.e., structural, logical, and user-defined).

A six-phase evolution methodology has been implemented within the KAON [14] infrastructure for business-oriented ontology management. The ontology evolution process starts with the capturing phase, that identifies the ontology modifications to apply either from the explicit business requirements or from the results of a change discovery activity. In the representation phase, the identified changes are described in a suitable format according to the specification language of the ontology to modify (e.g., OWL). The effects of the changes are evaluated in the semantics of change phase, where the ontology consistency check is also performed. Due to the fact that an ontology can reuse or extend other ontologies (e.g., through inclusion or mapping), the propagation phase ensures that any ontology change is propagated to the possible dependent artifacts in order to preserve the overall consistency. The subsequent implementation phase has the role to log all the performed changes in order to support the recovery facilities that in the final validation phase are provided to reverse an ontology change in case that an undesired effect occurs.

With respect to the state of the art on ontology evolution, the original contribution of the presented approach is an evolution methodology tailored to provide a semi-automated support to the new concept detection and definition activities, which are conventionally the most human-intensive activities. In particular, we address the problem of semi-automatically discovering the new concepts required in the ontology to describe new multimedia resources by exploiting ontology matching techniques.

VI. Conclusion

In this paper, we have presented a methodology for supporting the ontology enrichment activity in the context of multimedia ontology evolution. This problem and its peculiar requirements, is one of the main research issues addressed in the BOEMIE [1] project, where the discovery approach presented in the paper is studied. The original contribution of the work is focused on the role of ontology matching techniques with respect to the ontology evolution scenario. We have shown how ontology matching can support the ontology expert in retrieving new concepts to be added in the ontology. In the concept discovery activity, the use of matching techniques reduces the manual effort required to the ontology expert, by suggesting a set of alternatives (candidate concepts) for the definition of the new concept. Our future work will be devoted to: i) implement and test a software tool for supporting the whole discovery process; ii) study the problem of defining a new concept out of the specifications of retrieved matching concepts, by exploiting a combination of matching and reasoning techniques; iii) to investigate the problem of storing and maintaining the mappings retrieved among the ontology concept and the external knowledge sources, in order to reuse the results of an activity of concept discovery in subsequent stages of ontology enrichment.

Acknowledgment

This work is founded by the BOEMIE Project, FP6-027538 - 6th EU Framework Programme. The authors would like to thank the partners of the BOEMIE project for providing the pictures for the examples.

References

[1] BOEMIE Project, “Bootstrapping ontology evolution with multimedia information extraction,” IST-2004-2.4.7 Semantic-based Knowledge and Content Systems Proposal, 2004.

[2] S. Castano, A. Ferrara, and S. Montanelli, “Evolving open and independent ontologies,” *Journal of Metadata, Semantics and Ontologies*, 2006.

[3] ——, “Matching ontologies in open networked systems: Techniques and applications,” *Journal on Data Semantics (JoDS)*, vol. V, 2006.

[4] Y. Ding and S. Foo, “Ontology research and development. part 2 - a review of ontology mapping and evolving,” *Journal of Information Science*, vol. 28, no. 5, pp. 375–388, 2002.

[5] G. Flouris, D. Plexousakis, and G. Antoniou, “Evolving ontology evolution.” in *SOFSEM*, ser. Lecture Notes in Computer Science, J. Wiedermann, G. Tel, J. Pokorný, M. Bieliková, and J. Stuller, Eds., vol. 3831. Springer, 2006, pp. 14–29.

[6] V. Haarslev and R. Möller, “Description of the RACER system and its applications.” CEUR Electronic Workshop Proceedings, http://ceur-ws.org/Vol-49/, 2001, pp. 132–141.

[7] P. Haase and L. Stojanovic, “Consistent evolution of owl ontologies.” in *ESWC*, ser. Lecture Notes in Computer Science, A. Gómez-Pérez and J. Euzenat, Eds., vol. 3532. Springer, 2005, pp. 182–197.

[8] P. Haase and Y. Sure, “State-of-the-art on ontology evolution,” Institute AIFB, University of Karlsruhe, SEKT informal deliverable 3.1.1.b, 2004. [Online]. Available: http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/SEKT-D3.1.1.b.pdf

[9] J. R. Hobbs, M. Stickel, D. Appelt, and P. Martin, “Interpretation as abduction,” *Artificial Intelligence Journal*, vol. 63, pp. 69–142, 1993.

[10] M. Klein and N. Noy, “A component-based framework for ontology evolution,” 2003. [Online]. Available: citeseer.ist.psu.edu/article/klein03componentbased.html

[11] A. Maedche, B. Motik, L. Stojanovic, R. Studer, and R. Volz, “An infrastructure for searching, reusing and evolving distributed ontologies,” in *WWW*. ACM, 2003, pp. 439–448.

[12] G. A. Miller, “WordNet: A Lexical Database for English,” *Communications of the ACM (CACM)*, vol. 38, no. 11, pp. 39–41, 1995.

[13] N. F. Noy and M. Klein, “Ontology evolution: Not the same as schema evolution,” *Knowl. Inf. Syst.*, vol. 6, no. 4, pp. 428–440, 2004.

[14] D. Oberle, R. Volz, B. Motik, and S. Staab, “An extensible ontology software environment,” in *Handbook on Ontologies*, ser. International Handbooks on Information Systems, S. Staab and R. Studer, Eds. Springer, 2004, ch. III, pp. 311–333.

[15] L. Stojanovic, A. Madche, B. Motik, and N. Stojanovic, “User-driven ontology evolution management,” in *European Conf. Knowledge Eng. and Management (EKAW 2002)*. Springer-Verlag, 2002, p. 285300. [Online]. Available: http://wwwred.fzi.de/wim/publikationen.php?id=804

[16] L. Stojanovic, A. Maedche, N. Stojanovic, and R. Studer, “Ontology evolution as reconfiguration-design problem solving.” in *K-CAP*, J. H. Gennari, B. W. Porter, and Y. Gil, Eds. ACM, 2003, pp. 162–171.

[17] L. Stojanovic and B. Motik, “Ontology evolution within ontology editors,” in *OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002; Siguenza (Spain), 30th September 2002*, 2002. [Online]. Available: http://km.aifb.uni-karlsruhe.de/eon2002/program.html