

Enhancing Ontology Concept Design by Knowledge Discovery *

Silvana Castano and Alfio Ferrara
Università degli Studi di Milano
DICO - Via Comelico, 39, 20135 Milano - Italy
{castano,ferrara}@dico.unimi.it

Abstract

In this paper, we propose a knowledge discovery-based approach to ontology concept design. In our approach, concept design is a stepwise activity which exploits ontology matching techniques in order to retrieve useful external concepts semantically related to the design at hand. This way, the resulting ontology knowledge space is open towards external knowledge sources, by complementing the ontology expert knowledge with domain knowledge stored in other external sources, such as other domain ontologies, web directories, and, in general, the semantic web.

1. Introduction

Ontology design and ontology evolution are complex activities that require to define the most appropriate concepts for providing a shared conceptualization of a domain of interest. Concept definition is usually a manual, human-intensive activity, which requires the ontology expert to find the most appropriate specification of a missing concept in terms of concept name, properties, and semantic relations with already existing concepts [5, 10]. The definition of a conceptualization for a domain can be a simple task when the ontology expert has a deep knowhow of the domain and/or the domain is simple. But, in real applications, the domain to conceptualize is often complex and wide and the knowhow of the ontology expert could be limited. In such a scenario, the possibility to exploit as much as possible the knowledge provided by available conceptualizations of the same domain is important and tools are needed to support the expert in integrating this knowledge in the domain conceptualization. In this paper, we propose a knowledge discovery-based approach to concept design based on the idea of integrating/merging into a unified and consistent concept/ontology definition different knowledge fragments semantically related to the conceptualization at hand. This way, the resulting ontology knowledge space is open towards external knowledge sources, by complementing the

ontology expert knowledge with domain knowledge coming from other domain ontologies, web directories, and, in general, the semantic web. The paper is organized as follows. In Section 2, we give an overview of the proposed approach by introducing a running example. In Section 3, we describe external knowledge source probing activity, while, in Section 4, we present the activity of concept commitment. Finally, in Section 5, we give our concluding remarks.

2. Overview of the proposed concept design approach

Concept design is the activity of defining a new concept into an ontology, by setting a concept name, as well as properties and semantic relations that are required to frame the new concept in the ontology. Concept design is an important activity for both the creation of a new ontology from scratch and for the evolution of an existing ontology for adaptation to changed requirements. Both in ontology creation and in ontology evolution, domain conceptualization and concept design are usually manual activities. The idea behind our approach is to look at the semantic web as a repository of knowledge about the domain of interest and to harvest this knowledge to suggest a possible conceptualization of the domain at hand, by reducing the effort required to the ontology expert for concept definition. To this end, the input of concept design is constituted by i) a set of external knowledge sources (e.g., other domain ontologies, web directory taxonomies, lexical systems) and ii) an initial, also rough, set of concepts describing the domain at hand. The output will be a set of new concepts and axioms that are produced during the concept definition process.

The approach is articulated in two main phases: i) *probing external knowledge sources* and ii) *concept commitment*. In the probing phase, the initial concepts are used for searching semantically related concepts in the external knowledge sources, using so-called *probe queries* regarding an initial concept. The result is a set of mappings between the initial concepts and semantically related concepts in the external sources, detected by means of ontology matching techniques. Concept commitment is the activity of defining final ontology concepts by refining the initial concept definitions through the integration/merge of knowledge frag-

*This paper has been partially funded by BOEMIE, FP6-027538 - 6th EU Framework Programme and by ESTEEM MIUR PRIN project funded by the Italian Ministry of Education, University, and Research..

ments related to the retrieved matching concepts. The result of this activity is the addition of new ontology axioms in the current ontology in order to refine/enrich the current domain conceptualization.

Example. To illustrate the approach, we consider the case of creating an ontology to describe the organization of a Conference. We choose a set of domain ontologies¹ about conferences provided for the purpose of ontology matching evaluation [3]. For the example, we selected three semantic web ontologies, SOFSEM, CMT, and SIGKDD. We suppose to have the initial concept *Person* featured by *first_name* and *last_name* and we will show how to use the proposed approach to retrieve information about conference organizers from these three sample sources. We illustrate the approach working with a OWL *SHIF(D)* ontology; however, the approach can be easily extended to more expressive ontology languages.

3. Probing external knowledge sources

The goal of probing external knowledge sources is to find definitions of concepts semantically related to the initial concepts at hand. In order to evaluate the degree of semantic affinity between initial concepts and other external concepts, probe queries are defined and ontology matching techniques are exploited [2].

3.1. Composing probe queries

To the end of comparing an initial concept against external knowledge, we formulate a probe query specifying the information available about the characteristics of the initial concept to start the knowledge discovery activity.

Probe Query. Given a concept C , a probe query Q_C is defined as a 3-tuple of the form:

$$Q_C = \langle n(C), \mathcal{P}, \mathcal{A} \rangle$$

where, $n(C)$ denotes the name of C , \mathcal{P} denotes the set of properties featuring C , and \mathcal{A} denotes the set of concepts that are related to C , either because there is a semantic relation between them and C or because they are considered to be similar to C .

A probe query is defined for each available initial concept. Set \mathcal{P} is defined by collecting all the properties specified for C , while \mathcal{A} is defined by specifying all the concepts linked to C by a semantic relation in the initial concept.

3.2. Ontology matching and mappings

The goal of ontology matching is to evaluate the semantic affinity between the ontology elements (i.e., concept and properties) of different and independent ontologies [7, 8, 11]. In general, the result provided by ontology

¹The reader can find the ontologies and related metadata used in the example at <http://nb.vse.cz/svabo/oaiei2006/>.

matching tools is a set of mappings between the elements of a source ontology and the elements of a target ontology. A mapping is defined as a semantic correspondence between two ontology elements. It is associated with a mapping relation holding between the two elements and with a degree of confidence (usually in the range [0,1]) of such a relation. In our approach, we adopt our matching system HMatch [2]. HMatch is implemented as a plugin for the ontology editor Protégé² and can be easily used in combination with the editor for implementing the presented approach³. The result of the probing phase is a set of mappings among initial concepts and properties specified in the probe queries and the external matching ontology elements.

Ontology mapping. Given a pair of matching ontology elements E and E' , a mapping $\mathcal{M}(E, E')$ is defined as a 4-tuple of the form:

$$\mathcal{M}(E, E') = \langle E, E', \mathcal{R}, \mathcal{V} \rangle$$

where $\mathcal{R} \in \{ \overset{\equiv}{\Rightarrow}, \overset{\sqsubseteq}{\Rightarrow}, \overset{\supseteq}{\Rightarrow} \}$ is a mapping relation and $\mathcal{V} \in [0, 1]$ is a confidence value associated with \mathcal{R} .

The mapping relation provides information about the interpretation of the mapping holding between the matching elements E and E' . In particular, the relation $\overset{\equiv}{\Rightarrow}$ denotes the fact that E and E' are considered as equivalent, while $\overset{\sqsubseteq}{\Rightarrow}$ (resp., $\overset{\supseteq}{\Rightarrow}$) denotes that E is intended as a descendant (resp., ancestor) of E' .

Example. A simple probe query for the initial concept *Person* of our example is:

$$Q_{Person} = \langle Person, \{first_name, last_name\}, \{\} \rangle.$$

The probe query Q_{Person} is compared using HMatch against the three semantic web ontologies SOFSEM, CMT, and SIGKDD. For the sake of space, in Table 1, we report only the top-level portion of the results obtained against the three semantic web ontologies. Matching concepts discovered in this phase become *candidate concepts* for the subsequent activity of concept commitment.

4. Concept commitment

We define concept commitment as the activity of producing final concept definitions by refining the initial concept definitions through the integration/merging of the knowledge fragments (i.e., properties, axioms) associated with the external mapped concepts. Concept commitment relies on appropriate *unification rules* to handle the different situations and heterogeneities that can occur when integrating the various knowledge fragments related to different candidates concepts of a given probe query.

²<http://protege.stanford.edu/>

³Our approach to concept design is compatible with other matchmaking tools that provide mappings in this form, which is the case of the largest part of the state of the art tools.

Table 1. Results of the knowledge discovery

Source	Target	Relation	Confidence
Person	sofsem:Person	\equiv	1.0
Person	cmt:Person	\equiv	1.0
Person	sigkdd:Person	\equiv	1.0
Person	sofsem:Committee_Member	\supseteq	1.0
Person	sofsem:Chair	\supseteq	1.0
Person	cmt:ProgramCommitteeChair	\supseteq	1.0
Person	sigkdd:Author	\supseteq	1.0
Person	sigkdd:Author_of_Paper	\supseteq	1.0
first_name	cmt:name	\equiv	0.82
last_name	sofsem:has_the_last_name	\equiv	0.8
last_name	cmt:name	\equiv	0.82

4.1. Unification rules

Similarly to other approaches proposed for ontology evolution [6], we adopt a rule-based approach to concept commitment. In particular, we define unification rules to address all the possible cases of incompatibility/inconsistency among the knowledge specification of two candidate concepts retrieved for the same probe query. In particular, we have a rule (Rule 1) for the unification of names for both concepts and properties. Moreover, we have rules for the unification of property types (Rule 2), property domains (Rule 3), and property ranges (Rule 4), respectively. Finally, we have specific rules for the unification of property restrictions (Rule 5). Given two concepts C and C' where C is the initial ontology concept in the probe query and C' is a candidate concept, the unification rules are driven by the idea of preserving, as much as possible, the definition of the concept C , by automatically enriching it with all the compatible specifications of C' .

Rule 1: Name unification. Given a name n used as a name of concept or property, and the name n' of a candidate concept or property, the unified name \bar{n} is set by default to be equal to n .

Rule 2: Property type unification. Given the type t (datatype or object property) of an ontology property P and the type t' of a candidate property P' , the unified type \bar{t} is set to be equal to t .

Rule 3: Property domain unification. Given the domain D of an ontology property P and the domain D' of a candidate property P' , the unified domain \bar{D} is defined to be equal to D .

Rule 4: Property range unification. Given the range R of an ontology property P and the range R' of a candidate property P' , the definition of the unified range \bar{R} depends on the original types of P and P' . If the original types of P and P' are different (e.g., a datatype and an object property), \bar{R} is set to be equal to R . If both P and P' are datatype properties, \bar{R} coincides with the less restrictive between R and R' . For example, if R is the datatype `smallint` and R' is the datatype `integer`, we set \bar{R} to be equal to `integer` since

it is compatible with `smallint` data. For datatypes with a weak compatibility (e.g., `integer` and `string`), we set \bar{R} to be equal to R by default. The ontology expert can modify this behavior by choosing to define a new datatype as the union of the two original datatypes of P and P' . When P and P' are both object properties, we analyze their original ranges R and R' . If the two ranges contain matching concepts or R' is a subclass of R , \bar{R} is set to be equal to R . If R is a subclass of R' , \bar{R} is set to be equal to R' . Otherwise, \bar{R} is defined as the union of R and R' , such that $\bar{R} \equiv R \sqcup R'$.

Rule 5: Property restriction unification. To determine a unified property restriction \bar{PR} , we distinguish between *quantified restrictions* and *cardinality restrictions*. Quantified restrictions are of the form $\forall P.C$, $\exists P.C$ or $P : i$, where: $\forall P.C$ denotes that for each occurrence of the property P we have a range given by the concept C ; $\exists P.C$ denotes that it does exist an occurrence of the property P whose value is in the range given by the concept C ; $P : i$ denotes that the occurrence of the property P has the individual i as value. Cardinality restrictions are of the form (mc_P, MC_P) , where mc_P denotes the minimum number of occurrences of P while MC_P denotes the maximum number of occurrences of P . Different cases occur, as follows. i) *Unification of two quantified restrictions*: in this case, we apply the rules described in Table 2. The idea behind these rules is that we perform restriction unification by choosing the less restrictive option and by contemporary avoiding to introduce inconsistencies in the resulting concept definition. The original restriction PR is maintained otherwise. ii) *Unification of two cardinality restrictions*: given two cardinality restrictions (mc_P, MC_P) and $(mc_{P'}, MC_{P'})$, we define a new cardinality restriction $(mc_{\bar{P}}, MC_{\bar{P}})$ as the less restrictive cardinality, with $mc_{\bar{P}} = \min\{mc_P, mc_{P'}\}$ and $MC_{\bar{P}} = \max\{MC_P, MC_{P'}\}$. iii) *Unification of a quantified restriction and a cardinality restriction*: the general solution in this case is to modify the cardinality restriction in order to become compliant with the number of existential quantified restrictions to be unified. Given a number n of existential quantified restrictions of the form $\exists P.D$ and a cardinality restriction of the form $(mc_{P'}, MC_{P'})$, we modify $(mc_{P'}, MC_{P'})$ such that $mc_{P'} \leq n \wedge MC_{P'} \geq n$.

4.2. Concept commitment procedure

Given an initial concept C , the concept commitment procedure has the goal of refining its definition (i.e., the set of its properties and semantic relations) by integrating the candidate concepts selected during the probing phase. The concept commitment procedure works recursively on the set of mappings M_C established for C as the output of the probing phase (see Figure 1).

In the first step, the procedure is executed by taking into account the initial concept C and its mappings M_C . The procedure defines the sets $[C]^\equiv$, $[C]^\supseteq$, and $[C]^\sqsubseteq$, respectively. We cluster together the candidate concepts that result equivalent to C on the basis of the matching results (set

Table 2. Rules for the unification of quantified property restrictions

PR'/PR	$\forall P.C$	$\exists P.C$	$P : i$
$\forall P'.C'$	if $\exists M(C, C') : \forall \overline{P}.C$ otherwise: $\forall \overline{P}.(C \sqcup C')$	if $\exists M(C, C') : \forall \overline{P}.C, \exists \overline{P}.C$ otherwise: $\forall \overline{P}.(C \sqcup C'), \exists \overline{P}.(C \sqcup C')$	$\overline{P} : i$
$\exists P'.C'$	if $\exists M(C, C') : \forall \overline{P}.C, \exists \overline{P}.C$ otherwise: $\forall \overline{P}.(C \sqcup C'), \exists \overline{P}.(C \sqcup C')$	if $\exists M(C, C') : \exists \overline{P}.C$ otherwise: $\exists \overline{P}.(C \sqcup C')$	$\overline{P} : i$
$P' : i'$	$\forall \overline{P}.C$	$\exists \overline{P}.C$	$\overline{P} : i$

ConceptCommitment(C, M_C)
define:
 $[C]^{\equiv} = \{C_1, \dots, C_n\} \mid \forall C_i, \exists \langle C, C_i, \equiv, v \rangle \in M_C$
 $[C]^{\supseteq} = \{C_1, \dots, C_n\} \mid \forall C_i, \exists \langle C, C_i, \supseteq, v \rangle \in M_C$
 $[C]^{\sqsubseteq} = \{C_1, \dots, C_n\} \mid \forall C_i, \exists \langle C, C_i, \sqsubseteq, v \rangle \in M_C$
 $\forall C_i \in [C]^{\equiv},$
define:
 \overline{C} by unifying C with C_i using Rules 1-5
if $[C]^{\supseteq}$ and $[C]^{\sqsubseteq}$ are empty or equal to a previously defined set:
goto exit
else:
for each $C_i \in [C]^{\supseteq}$:
 $M_{C_i} = \text{matching}(C_i, [C]^{\supseteq})$
ConceptCommitment(C_i, M_{C_i})
for each $C_j \in [C]^{\sqsubseteq}$:
 $M_{C_j} = \text{matching}(C_j, [C]^{\sqsubseteq})$
ConceptCommitment(C_j, M_{C_j})
}

Figure 1. Concept commitment procedure

$[C]^{\equiv}$, the candidate concepts that are considered as candidate ancestors (set $[C]^{\supseteq}$), and descendant of C (set $[C]^{\sqsubseteq}$), respectively. Then, first, concepts in $[C]^{\equiv}$ are integrated with C into a unified concept \overline{C} , by applying Rules 1-5. Then, we take into account $[C]^{\supseteq}$ and $[C]^{\sqsubseteq}$ and, for each of them, we execute matching among the concepts contained in the set. The result of the matching procedure is a new set of mappings between these concepts. On the basis of new mappings, we execute again the first step of the procedure for each concept in $[C]^{\supseteq}$ and in $[C]^{\sqsubseteq}$. The concept commitment procedure is recursively iterated until $[C]^{\supseteq}$ and $[C]^{\sqsubseteq}$ are empty or a concept already defined in a previous step is defined. The final result of such a procedure is a concept taxonomy where each concept is obtained by the unification of a set of equivalent concepts. The final taxonomy is then submitted to the ontology expert for validation.

Example. In order to provide an example of the procedure, we take into account the initial concept *Person* and the mappings shown in Table 1. A graphical representation of the execution of the concept commitment procedure is shown in Figure 2. The procedure is iterated three times.

Iteration 1. The analysis of the mappings retrieved for *Person* leads to the definition of the following sets:

$[Person]^{\equiv} = \{sofsem : Person, cmt : Person, sigkdd : Person\},$
 $[Person]^{\supseteq} = \{Committee_Member, Author, Author_of_Paper, Chair, ProgrammCommitteeMember\},$
 $[Person]^{\sqsubseteq} = \emptyset$

Concepts in $[Person]^{\equiv}$ are unified into *Person* (Rule 1).

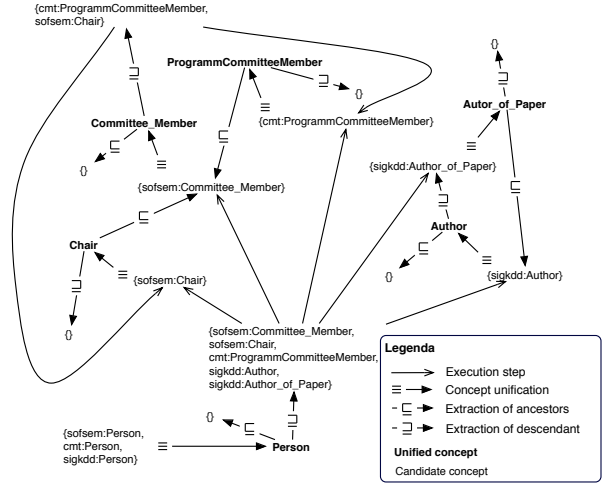


Figure 2. Example of concept commitment procedure execution for concept Person

Properties *first_name*, *sofsem : has_the_first_name*, and *cmt : name* are unified into *first_name*. Analogously, *last_name* is maintained in the new unified concept *Person* (Rules 1-5). Then, each of the five concepts in $[Person]^{\supseteq}$ is matched against all the concepts of $[Person]^{\supseteq}$, producing five concepts and related sets of mappings given as input to the concept commitment procedure again. In the following, we describe the next iteration for the concept *Committee_Member* with its associated set of mappings $M_{Committee_Member}$:

$\{(Committee_Member, Committee_Member, \equiv, 1.0),$
 $(Committee_Member, ProgrammCommitteeMember, \supseteq, 1.0),$
 $(Committee_Member, Chair, \sqsubseteq, 1.0)\}$

Iteration 2. The set of mappings $M_{Committee_Member}$ is leads to the definition of the sets:

$[Committee_Member]^{\equiv} = \{Committee_Member\},$
 $[Committee_Member]^{\supseteq} = \{ProgrammCommitteeMember, Chair\},$
 $[Committee_Member]^{\sqsubseteq} = \emptyset$

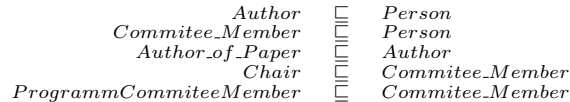
The unification of $[Committee_Member]^{\equiv}$ is trivial. The matching is executed for each concept in $[Committee_Member]^{\supseteq}$ (i.e., *ProgrammCommitteeMember*, *Chair*), and on each resulting concept and mapping set the concept commitment procedure is invoked again.

Iteration 3. Taking into account the concept *Chair*, its mapping set leads to:

$$\begin{aligned} [Chair]^{\equiv} &= \{Chair\}, \\ [Chair]^{\supseteq} &= \emptyset, \\ [Chair]^{\sqsubseteq} &= \{Committee_Member\} \end{aligned}$$

Since $[Chair]^{\supseteq}$ is empty and $[Chair]^{\sqsubseteq}$ contains a concept already defined, the procedure ends. The case of *ProgrammCommitteeMember* is analogous.

By analyzing the resulting graph, the ontology expert introduces an appropriate subclass relation into the ontology for each concept derived from the unification of an equivalence set, thus leading to the following taxonomy:



Considerations. During concept unification, new concepts could be introduced in the ontology, due to the introduction of new axioms. Once the concept commitment procedure is terminated, we start a new discovery phase for each newly concept introduced into the ontology. In particular, all matching concepts will be properly unified into a unique representation by applying the concept commitment procedure again. In our example, the new concept *ProgramCommitteeMember* is defined as follows:

$$ProgramCommitteeMember \sqsubseteq \exists memberOf. ProgramCommittee.$$

This causes the introduction of the property *memberOf* and the concept *ProgramCommittee* in the ontology. Once the design process ends, the expert will use conventional reasoning facilities to ensure that resulting ontology is consistent. When an inconsistency is detected, the ontology expert can rely on the reasoning output to find the most suitable solution for modifying the ontology.

The effectiveness of the approach in real-world applications involves the experimentation and evaluation in real cases of ontology design and evolution. This involves first of all the accuracy of the knowledge-discovery approach, and consequently the accuracy of the matching process. The HMatch system has been extensively evaluated on ontology matching benchmarks and experimental results concerning precision and recall have been analyzed [1]. Moreover, further evaluations will be performed concerning the quality of the resulting concepts using retrieved matching knowledge. On this issues we will work in the framework of the BOEMIE project working specifically on the ontology evolution case.

5. Concluding remarks

In this paper, we have presented an approach for supporting concept design based on knowledge discovery and matching techniques. The proposed approach can be adopted for supporting both the creation of a new ontology from scratch and the evolution of an existing ontology. In the first case, the input of concept definition is constituted by the results of the domain analysis activity,

which is usually performed as the initial activity in ontology creation [4, 5]. Instead, when an existing ontology is evolved, the input of concept definition is constituted by the initial version of the ontology and by the results of the change analysis phase [9, 12]. We argue that knowledge discovery could play an important role in supporting the expert in concept definition, because, on one hand, it can help the ontology expert in better understanding the domain and, on the other hand, it can simplify his work and reduce the manual effort required for concept design. This knowledge discovery-enabled approach is being developed in the framework of the EU project BOEMIE, for evolving multimedia ontologies. In such a context, our ongoing and future work is mainly devoted to the development of a comprehensive system capable of supporting the ontology expert in all the activities of ontology evolution and to the evaluation of the proposed techniques and of the resulting ontologies.

References

- [1] S. Castano, A. Ferrara, and G. Messa. Islab hmatch results for oaei 2006. In *Proc. of International Workshop on Ontology Matching, collocated with the 5th International Semantic Web Conference ISWC-2006*, Athens, Georgia, USA, November 2006.
- [2] S. Castano, A. Ferrara, and S. Montanelli. Matching Ontologies in Open Networked Systems: Techniques and Applications. *Journal on Data Semantics (JoDS)*, V:25–63, 2006.
- [3] J. Euzenat, M. Mochol, P. Shvaiko, H. Stuckenschmidt, O. Svab, V. Svatek, W. R. van Hage, and M. Yatskevich. First results of the ontology alignment evaluation initiative 2006. In *Proc. of International Workshop on Ontology Matching, collocated with ISWC-2006*, Athens, Georgia, USA, November 2006.
- [4] D. Fensel. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag, Berlin, 2000.
- [5] A. Gomez-Perez, M. Fernandez-Lopez, and O. Corcho. *Ontological Engineering*. Springer Verlag, 2003.
- [6] P. Haase and L. Stojanovic. Consistent Evolution of OWL Ontologies. In *Proc. Of the 2nd European Semantic Web Conference (ESWC 2005)*, pages 182–197, Heraklion, Crete, Greece, 2005.
- [7] Y. Kalfoglou and M. Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review Journal*, 18(1), 2003.
- [8] N. Noy. Semantic integration: a survey of ontology-based approaches. *SIGMOD Record Special Issue on Semantic Integration*, December 2004.
- [9] N. F. Noy and M. Klein. Ontology evolution: Not the same as schema evolution. *Knowledge and Information Systems*, 6(4):428–440, July 2004.
- [10] H. S. Pinto and J. P. Martins. A methodology for ontology integration. In *K-CAP '01: Proceedings of the 1st international conference on Knowledge capture*, pages 131–138, New York, NY, USA, 2001. ACM Press.
- [11] P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics (JoDS)*, 1, 2005.
- [12] L. Stojanovic and B. Motik. Ontology Evolution within Ontology Editors. In *Proc. of the EKAW Workshop on Evaluation of Ontology-based Tools (EON 2002)*, Siguenza, Spain, 2002.