



## International Journal of Web Information Systems

Adaptive ontology re-use: finding and re-using sub-ontologies  
Rodolfo Stecher, Claudia Niederée, Wolfgang Nejdl, Paolo Bouquet,

### Article information:

To cite this document:

Rodolfo Stecher, Claudia Niederée, Wolfgang Nejdl, Paolo Bouquet, (2008) "Adaptive ontology re-use: finding and re-using sub-ontologies", International Journal of Web Information Systems, Vol. 4 Issue: 2, pp.198-214, <https://doi.org/10.1108/17440080810882379>

Permanent link to this document:

<https://doi.org/10.1108/17440080810882379>

Downloaded on: 02 February 2018, At: 01:01 (PT)

References: this document contains references to 22 other documents.

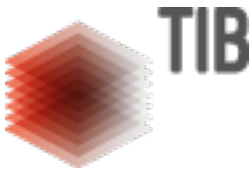
To copy this document: [permissions@emeraldinsight.com](mailto:permissions@emeraldinsight.com)

The fulltext of this document has been downloaded 315 times since 2008\*

### Users who downloaded this article also downloaded:

(2010), "Dynamic ontology as an ontological framework of anticipatory systems", foresight, Vol. 12 Iss 3 pp. 38-49 <<https://doi.org/10.1108/14636681011049866>><https://doi.org/10.1108/14636681011049866></a>

(2014), "Ontology in information studies: without, within, and withal knowledge management", Journal of Documentation, Vol. 70 Iss 3 pp. 425-442 <<https://doi.org/10.1108/JD-06-2012-0077>><https://doi.org/10.1108/JD-06-2012-0077></a>



Access to this document was granted through an Emerald subscription provided by emerald-srm:271967 []

### For Authors

If you would like to write for this, or any other Emerald publication, then please use our Emerald for Authors service information about how to choose which publication to write for and submission guidelines are available for all. Please visit [www.emeraldinsight.com/authors](http://www.emeraldinsight.com/authors) for more information.

### About Emerald [www.emeraldinsight.com](http://www.emeraldinsight.com)

Emerald is a global publisher linking research and practice to the benefit of society. The company manages a portfolio of more than 290 journals and over 2,350 books and book series volumes, as well as providing an extensive range of online products and additional customer resources and services.

Emerald is both COUNTER 4 and TRANSFER compliant. The organization is a partner of the Committee on Publication Ethics (COPE) and also works with Portico and the LOCKSS initiative for digital archive preservation.

\*Related content and download information correct at time of download.



# Adaptive ontology re-use: finding and re-using sub-ontologies

Rodolfo Stecher, Claudia Niederée and Wolfgang Nejdll  
*L3S Research Center/University of Hannover, Germany, and*  
Paolo Bouquet  
*Department of Information and Communication Technologies,  
University of Trento, Italy*

## Abstract

**Purpose** – The discovery of the “right” ontology or ontology part is a central ingredient for effective ontology re-use. The purpose of this paper is to present an approach for supporting a form of adaptive re-use of sub-ontologies, where the ontologies are deeply integrated beyond pure referencing.

**Design/methodology/approach** – Starting from an ontology draft which reflects the intended modeling perspective, the ontology engineer can be supported by suggesting similar already existing sub-ontologies and ways for integrating them with the existing draft ontology. This paper’s approach combines syntactic, linguistic, structural and logical methods into an innovative modeling-perspective aware solution for detecting matchings between concepts from different ontologies. This paper focuses on the discovery and matching phase of this re-use process.

**Findings** – Owing to the combination of techniques presented in this general approach, the work described performs in the general case as well as approaches tailored for a specific usage scenario.

**Research limitations/implications** – The methods used rely on lexical information obtained from the labels of the concepts and properties in the ontologies, which makes this approach appropriate in cases where this information is available. Also, this approach can handle some missing label information.

**Practical implications** – Ontology engineering tasks can take advantage from the proposed adaptive re-use approach in order to re-use existing ontologies or parts of them without introducing inconsistencies in the resulting ontology.

**Originality/value** – The adaptive re-use of ontologies by finding and partially re-using parts of existing ontological resources for building new ontologies is a new idea in the field, and the inclusion of the modeling perspective in the computation of the matches adds a new perspective that could also be exploited by other matching approaches.

**Keywords** Knowledge management systems, Computer software, Computer theory, Task specialization, Specifications

**Paper type** Research paper

## 1. Introduction

Ontology re-use is an agreed upon goal in ontology engineering. It reduces the cost of creating ontologies, improves the quality of the resulting ontologies, and eases later interaction between systems. The re-use of ontologies and of knowledge collected in the context of ontology creation comes in many flavors. Ontologies may be referenced, imported, taken as a starting point for extensions and revisions, or

The work described in this paper has been partly funded by the European Commission through grant to the Project Nepomuk under the Number IST-027705.



---

taken as templates for the development of similar ontologies in other domains or for other purposes. Considering this more systematically, we distinguish three types of ontology re-use:

- (1) With conservative re-use the re-used ontology stays unaffected. Concepts, properties or individuals are used in the way they are defined in the re-used ontology, e.g. for defining new subclasses. This type of re-use is, for example, reflected in the work of Grau *et al.* (2007).
- (2) In adaptive re-use, the re-used ontology provides a starting point for local definitions, possibly changing the way concepts and properties are defined to fit the own purposes.
- (3) In best practice re-use, the know-how, best practices, and experiences of how an ontology is constructed are re-used as in Uschold *et al.* (1998) and Rector (2003).

The “right” type of re-use depends on factors such as the type of the ontology to be constructed and of the ontology to be re-used (top-level vs application ontology), the availability of widely accepted ontologies and the purpose of, and the requirements toward the constructed ontology.

Conservative re-use is clearly most valuable in the sense of propagating ontologies as a shared conceptualization. However, in many situations – especially, when application-specific ontologies are built – there is a gap between available ontologies and the ontology required. Our work, therefore takes a closer look on the adaptive re-use. In more detail, we are developing a method for supporting adaptive ontology re-use, which takes into account the modeling perspective selected by the ontology engineer and supports her in finding and integrating useful parts of existing ontologies. This reflects the fact that a part of a domain can be modeled in many ways depending on the purpose, individual conceptualization, etc. – taking different modeling perspectives.

Our approach combines lexical, linguistic, structural and logic methods for finding matches between ontologies by taking into account the intended modeling perspective. A modeling perspective can be communicated by the engineer by a first ontology draft. Based on computed matches, we extract a module containing the matching elements and reuse it in the constructed ontology. Our work builds upon work done in the area of ontology matching, ontology integration and ontology modularization.

For example, we use ontology matching as a starting point for identifying similar ontologies and find overlaps between the ontology draft and the available ontologies, and ontology modularization to select reasonable ontology portions from the selected ontology. Finally, ontology integration is considered for the merging of the detected ontology portions with the draft or start ontology. In this work, we explore the space of ontology re-use that lies between conservative extensions (Grau *et al.*, 2007) and the pure ensuring of consistency of the resulting ontology. This results in the process shown in Figure 1.

This paper presents an overview of the entire process, a new set-based matching method and the details of a combination of matching approaches in order to find matching ontology concepts out of a pool of ontologies, under consideration of the modeling perspective, as well as the implementation and evaluation results.

The paper is structured as follows. First, Section 2 discusses some related work, Sections 3 and 4 describe our approach and the details of the matching part of it,

1. INPUT: start ontology draft  $s$  with concept set  $C_{sel}$  and set  $O$  of existing ontologies
2. Search for (possibly) related candidate ontologies  $co_i$  in  $O$ .
3. For each candidate ontology  $co_i$ :
  - (a) Find existing matching concepts between  $co_i$  and  $s$  (by considering  $C_{sel}$ ) taking into account the modeling perspective
  - (b) Compute the similarity between  $s$  and  $co_i$  based on the matching results
4. select the candidate ontology  $c_s$  with the best matching result
5. Compute the (minimal) module in the  $c_s$  that contains the matching elements
6. Analyze the integration/merging feasibility of the computed modules with the start ontology
7. OUTPUT: Suggestions for merged ontology  $o_3$  to the engineer to decide about the merging/integration.

**Figure 1.**  
General description  
of our approach

Section 5 presents an overview of our prototype, Section 6 provides results of the performed evaluations on the matching. The paper finishes in Section 7 with conclusions and future work.

## 2. Related approaches

Our approach is related to and builds upon work in the areas of ontology reuse, ontology modularization, and ontology matching which will be presented briefly.

The most recent overview and classification of work existing in the ontology matching can be found in Euzenat and Shvaiko (2007). This overview presents not only a variety of systems and their details, but also a comprehensive classification of all basic techniques currently used by the existing matching approaches.

Approaches such as iPrompt (Noy and Musen, 2003) rely on syntactical, lexical and structural information. Its tool AnchorPrompt produces a set of new pairs of semantically close terms by using structural similarity. AnchorPrompt has difficulties to detect similar concepts if the analyzed ontologies are structurally very different. MoA (Kim *et al.*, 2005) is an approach to merge and align OWL ontologies which uses linguistic methods to disambiguate the meaning of elements based on their local names as we do in our approach. It provides an algorithm to detect semantic equivalences (specified as a semantic bridge) of concepts and properties and a merging algorithm which uses this semantic bridge for ontology merging. Others like GLUE (Doan *et al.*, 2003) and OMEN (Mitra *et al.*, 2005) use in contrast mainly probabilistic approaches to derive matches. Furthermore, there are also Logical or SAT-based approaches. For example, the CTXMatch (Bouquet *et al.*, 2005) approach discovers semantic relations between nodes of different schemata by reasoning on the explicit representation of the meaning of each node. We extend this approach by combining it with our set-oriented and a structure-based approach.

Approaches in ontology modularization focus on properly structuring ontologies at construction time for better reusing them in the future, or on extracting parts or modules of existing ontologies while preserving the original semantics. In Rector (2003), for example, guidelines are given on how to modularize ontologies for latter easier module reuse including strategies of low coupling and high cohesion as known from software engineering. The second kind of modularization approaches, namely the detection or extraction of (semantic preserving) modules out of existing ontologies as

---

well as their merging and integration, is highly related to our work Grau *et al.* (2007) present an approach to extract modules from an ontology which is based on a definition of module that guarantees to completely capture the meaning of a given set of terms based on conservative extensions.

Recently, various viable approaches for ontology reuse have been proposed (Ding *et al.*, 2007; Alani, 2006; Bontas *et al.*, 2005). Our work is very similar to the one presented in Alani (2006), where existing methods and technologies are integrated to enable the (semi-)automatic reuse of ontologies or parts of them. Ding *et al.* (2007) present an approach for extracting parts of existing ontologies based on a corpus, so that at the end the corpus information can be represented with the obtained ontology (parts). (Bontas *et al.*, 2005) present studies on reusing ontologies, explaining where the major problems and costs of reuse are, which is an important aspect to be considered. The evidence found in these papers reinforces our belief that our approach is needed and would be of much help in ontology engineering activities.

### 3. Overview

Starting point of our approach for supporting ontology engineering is an ontology module  $m_1$  built from a draft start ontology  $s$  and a set of concepts  $C_{\text{sel}}$  selected from  $s$  that reflects a first idea of what the ontology engineer wants to build, and a set  $O$  of existing, partially overlapping candidate ontologies.

The goal is to build an ontology  $O$  that is constructed by extending  $m_1$  by re-using parts of ontologies in  $O$ . For this purpose, we first identify an ontology module from one of the ontologies in  $O$  with the following properties:

- $m_2$  covers the intended aspects of the domain;
- $m_2$  respects the modeling perspective communicated by the engineer in  $s$ ; and
- $m_2$  has the right size to be useful (ontology module).

Subsequently,  $m_1$  is extended by  $m_2$ , where a form of adaptive extension for re-use is applied. The complete process is shown in Figure 1.

Imagine a scenario where the ontology engineer sketches a start draft ontology as shown in Figure 2, selects some concepts of interest as shown in Figure 3, and starts a search for candidate ontologies. Let us consider that one of the found candidate ontology is the one shown in Figure 4. We want to find how much of the selected concepts of the start ontology is represented in this candidate ontology following a similar modeling perspective.

Section 4 presents the details of Step 3. Steps 5-7 will not be explained in detail in this paper but sketched out in Section 7 and will be discussed in more details in a following paper.

### 4. The match

Before introducing the steps of the matching method, some general definitions have to be presented. The match is computed between a selected set of concepts and its properties from a draft “start” ontology  $s$ , and all the concepts of a candidate ontology  $co$  as described in Figure 5. Output of the matching process is a set of relations between concepts and a measure that describes to which extent  $co$  overlaps-with or covers the concepts selected from the start ontology  $s$ .

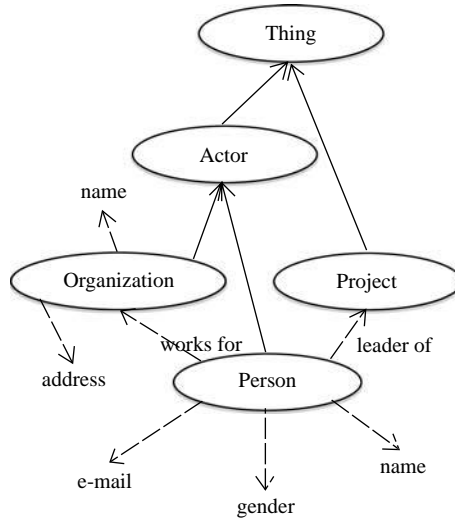


Figure 2.  
Start ontology

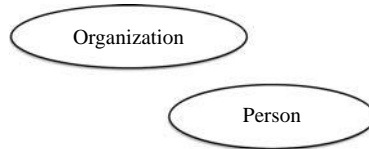


Figure 3.  
Selected concepts

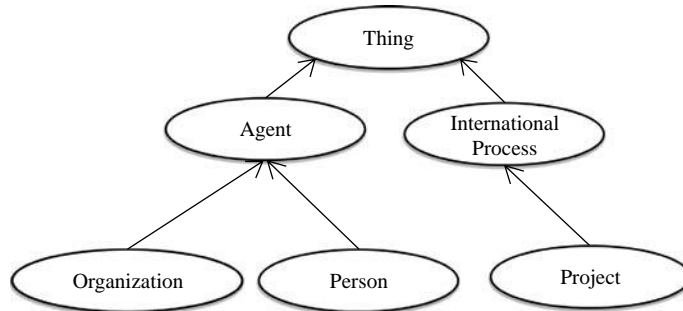


Figure 4.  
Example candidate  
ontology

for all selected concepts  $C_{sel}$  in  $s$  and the concepts in candidate ontology  $co$  do  
    Compute the similarity of concepts  
    Compute relations between concepts  
end for  
Compute the *coverage* between  $s$  and  $co$

Figure 5.  
General matching  
approach

The first step in our approach is to compute the context of the concepts. This is presented in the following section.

#### 4.1 Compute context of concepts

The context of a concept  $c$  is represented as a graph, which we call context graph  $cx$  – containing the elements “surrounding”  $c$  in the ontology. This context is defined with a radius  $r$ , so, the context graph with center element  $c$  and radius  $r$  is noted as  $cx(c, r)$ . Iteratively starting in  $c$ , the range/domain relationships and the sub/super hierarchies are traversed until path length  $r$  is reached ( $r$  limits the distance of the traversal). Nodes are added to the context graph for concepts and properties encountered on the path. Edges are added for the traversed relationships (domain/range, sub/super). Such a context graph is created for all concepts in  $C_{sel}$  and for those in  $c$ .

Each element  $e'$  in the context graph receives an element weight ( $wElement(e')$ ) and a distance weight ( $wDist$ ). The element weight ( $wElement(e')$ ) is assigned depending on the type of the considered element: concept, locally defined property, or inherited property. The distance weight ( $wDist$ ) depends on the distance to the center concept in the graph ( $dist(c, e')$ ) and is computed so that it decreases rapidly when the distance to the center element approaches the radius  $r$ , in order to give more weight to elements close to the center:

$$wDist(c; e^0) = \alpha \cdot 2^{\frac{\log_2(\alpha \cdot r / \alpha)}{r + 1}} \cdot dist(c; e^0) + (1/\alpha) \text{ having } \alpha > 1.$$

Our experiments have shown that choosing  $\alpha = 1.1$  give satisfactory results.

Owing to the fact that the properties are included in the context computation, the modeling perspective is captured and will influence all following computations. The context is used for disambiguating the meaning of the label of each center element (see below).

#### 4.2 Element meaning disambiguation

In many cases, the labels of the elements – property and concept names – in an ontology reflect part of the meaning of such elements. We extract the labels of the elements appearing in each computed context and retrieve from a lexical resource such as WordNet (Fellbaum, 1998) all the possible senses of the terms in the label.

The meaning of an element highly depends on the context where it is employed as for example the term “jaguar,” which might denote a brand or an animal. In general, only a subset of the found senses are meant by one concept. For removing irrelevant senses, we measure the relevance of each sense taking into account the context  $cx$ .

For the disambiguation of the most likely intended meanings of the center element of each computed context, we combine the work proposed in Hirst and St-Onge (1997), Silber and McCoy (2002) and Galley and McKeown (2003) and adapt it to our scenario by taking all senses of all words of the context’s center concept label (for simplicity “sense of the concept”), its synonyms, holonyms, hypernyms and the nouns appearing in the gloss, and compare each of them with each of the senses of the words of the element labels in the context.

If a sense of the context’s center concept appears among the senses of a context element, we compute a relation weight ( $wRel$ ) for this concept sense (Hirst and St-Onge, 1997), based on the relation found (synonym, hypernym, holonym or noun (Silber and



McCoy, 2002) in the gloss (Lesk, 1986)). wRel is combined with the corresponding wElement value of the context element and the wDist value between the context's center concept and the context element, and accumulated for each sense (relation of a sense with all senses in the context).

The normalized resulting value for each sense gives the disambiguated weight of the sense (dwSense). The senses whose dwSense value is below a sense relevance threshold value (in our current tests 0.05) are discarded and removed from the list of intended senses.

As a result, we have for every relevant word in the label of the context's center concept its relevant senses and the corresponding sense weights. The reader is reminded that such a context graph is created for all concepts in  $C_{sel}$  and for those in  $co$ , and for each center element the meaning is disambiguated. Next, we compute measures for context and concept similarity between concepts in  $C_{sel}$  and concepts of each  $co$ .

#### 4.3 Concept similarity computation

In this section, the different measures for the computation of the concept similarity will be presented.

*4.3.1 Set-based concept similarity measure.* The senses space of a concept is defined by all its senses. This senses space is treated as sets and the overlap of the different sets of two concepts is computed. The weight of the senses dwSense determines the relative size of the corresponding sets so that senses with higher weight have a corresponding set which is "larger" than senses with lower weight. The set overlap gives a measure of the concept similarity (cSim). The description of how this similarity measure is computed is shown in Figure 6. This is performed for every concept in  $C_{sel}$  compared with every concept in  $co$  so that at the end a measure of the similarity of every possible pair of concepts is available.

*4.3.2 Set-based context similarity measure.* The context similarity measure (ctxSim) is computed similarly to cSim, but is extended by considering all concepts and properties in the context and the overlap of the sets determined by the corresponding senses. The relative overlap is computed and accumulated which gives a measure for the context similarity (ctxSim). The steps of the computation of the context similarity are shown in Figure 7.

*4.3.3 Concept similarity measure.* The similarity (sim) is the similarity value between two concepts, computed by combining the local or concept similarity cSim and the global or context similarity ctxSim measures:

$$\text{sim}(c; c^0) = \min(\text{cSim}; \text{ct} \times \text{Sim}) + \frac{\text{jcSim}_i \text{ ct} \times \text{Sim}_j}{2}$$

#### 4.4 Concept relation computation

In this section, the computation of the logical relations holding between concepts in the two different ontologies will be presented (concepts in  $C_{sel}$  and in  $co$ ). A combination of different approaches is applied, one based on the set-based sense representation as presented in the previous section, the SAT-based approach CTXMatch (Bouquet *et al.*, 2005) and a structure-based approach. The results of all three approaches are then combined in order to decide the logical relation that holds between the analyzed concepts.

*4.4.1 Set-based relation discovery.* The approximation of the relation holding between two concepts is computed by analyzing:

```

for all concept csel in  $C_{sel}$  do
  for all concepts cco in the candidate ontology co do
    Compute the intersection of senses SINT between senses of csel and senses
    of cco
    overlap = 0:0
    for all sense sens in SINT do
      Compute  $wDif = \min(dwSense_{sens_{csel}}; dwSense_{sens_{cco}})$ 
      Accumulate the partial sense similarity of both concepts  $overlap =$ 
      overlap +  $wDif$ 
    end for
    synSim = hypSim = holSim = 0
    for all senses csel sense of csel do
      Compute the synonym set (synset), the hypernym set (hypset) and the
      holonym set (holset) of csel sense from the lexical resource
      for all senses ccosense of cco do
        if there is a common occurrence in synset and ccosense then
          Compute  $wDif = \min(dwSense_{csel\ sense}; dwSense_{ccosense})$ 
          Accumulate  $wDif$  weighted with a synonym relation factor  $synSim =$ 
           $synSim + wDif \times synFactor$ 
          Reduce the size of the sense's sets csel sense and ccosense by  $wDif$ 
        end if
        if there is a common occurrence in hypset and ccosense then
          Compute  $wDif = \min(dwSense_{csel\ sense}; dwSense_{ccosense})$ 
          Accumulate  $wDif$  weighted with a hypernym relation factor
           $hypSim = hypSim + wDif \times hypFactor$ 
          Reduce the size of the sense's sets csel sense and ccosense by  $wDif$ 
        end if
        if there is a common occurrence in holset and ccosense then
          Compute  $wDif = \min(dwSense_{csel\ sense}; dwSense_{ccosense})$ 
          Accumulate  $wDif$  weighted with a holonym relation factor  $holSim =$ 
           $holSim + wDif \times holFactor$ 
          Reduce the size of the sense's sets csel sense and ccosense by  $wDif$ 
        end if
      end for
    end for
    OUTPUT: the similarity between both concepts  $cSim =$ 
    overlap +  $synSim + hypSim + holSim$ 
  end for
end for

```

**Figure 6.**  
Set-based concept  
similarity measure

- (1) the relative overlap of the sets defined by the senses of the considered concepts (as already shown in Figure 6); and
- (2) the lexical relations existing between the senses of this concepts.

For (2), the lexical resource is inspected and synonyms, hypernyms and holonyms are investigated in order to find out what kind of (if any) lexical relations hold between the senses of the concepts being compared by considering its semantic neighborhood (Teich and Fankhauser, 2004).

The procedure for discovering the relations holding between concepts is shown in Figure 8.

*4.4.2 SAT-based relation computation.* All concept pairs from  $C_{sel}$  and  $co$  are fed into a reasoner in order to compute the logical relations holding between them.

```

for all concepts csel in Csel do
  Retrieve the context cxsel of csel (is the center element)
  for all concepts cco in co do
    Retrieve the context cxcco of cco
    for all element cxsele of context cxsel do
      ctx EleSim = 0
      for all element cxcco of context cxcco do
        Compute the similarity cSim between cxsele and cxcco using the ap-
        proach presented in Figure??
        Accumulate the weighted similarity (higher di@erence in distance
        from the center element, less similar perspective) by ctxEle Sim =
        ctxEleSim + cSim *  $\frac{1}{2^{\text{dist}(cxsele; csel) + \text{dist}(cxcco; cco)}}$ 
      end for
      Accumulate the mean value for ctxEleSim in ctxSim
    end for
    OUTPUT: the normalized context similarity ctxSim of the pair of concepts
    csel and cco
  end for
end for

```

**Figure 7.**  
Set-based context  
similarity measure

In order to do so, a logical expression of the concept is constructed by analyzing the corresponding labels.

The logical expression denoting the concept meaning is created based on the results obtained from a head-modifier tree which is built to identify the head word in the label and its modifiers, as proposed in Hovy *et al.* (2005). For this task, the parser presented in Koster (2003) is used. By traversing the head-modifier tree, a conjunction/disjunction expression of the different words in the label is built. The occurring words are then replaced by the conjunction of all corresponding senses, going in this way from the purely syntactic world to the semantic world and enabling the comparison of concepts with different labels but with possibly similar meaning. An example for the concept “Organization” in Figure 2 is:

Organization = ((organization#4[organization#5]actor#1[actor#2])

The logical expression of a concept does not only contain the senses of the current concept, but also considers the meaning of the superconcepts of it, taking the hierarchical information into account. Our tests showed that including this hierarchical information substantially increases the precision. Once the logical formulas describing each concept of  $C_{sel}$  and each concept in  $co$  were added to a reasoner, we query for the relations holding between each pair ( $C_{sel}$  concept,  $co$  concept).

The result, stored in a similarity object for each pair of concepts, is a relation specifying whether the two concepts are equivalent, more/less general, or their relationship is unknown.

*4.4.3 Structure-based relation deduction.* In a similar approach to the one presented in Noy and Musen (2003) or Mitra *et al.* (2005), earlier detected matches are used for deducing other matches by taking into account the structural information from the respective ontologies. If a  $C_{sel}$  concept without a match is detected in the “is-a”

---

```

for all concept csel in  $C_{sel}$  do
  for all concepts cco in the candidate ontology co do
    eq = 0, synSim = 0, hypSim = 0, holSim = 0
    Compute the intersection SINT between senses of csel and senses of cco
    for all sense sens in SINT do
      Compute wDif = min (dwSensesenscsel; dwSensesenscco)
      Accumulate wDif as the partial sense equality of both concept seq =
      eq + wDif
      Reduce the size of the sense's set sens by wDif in csel and co
    end for
    for all sense csel sense of csel do
      for all sense ccosense of cco do
        Compute the synonym set (synset), the hypernym set (hypset) and the
        holonym set (holset) of csel sense from the lexical resource
        if there is a common occurrence in synset and ccosense then
          Compute wDif = min (dwSensecsel sense; dwSenseccosense)
          Accumulate wDif weighted with a synonym relation factor synSim =
          synSim + wDif  $\times$  synFactor
          Reduce the size of the sense's sets csel sense and ccosense by wDif
        end if
        if there is a common occurrence in hypset and ccosense then
          Compute wDif = min (dwSensecsel sense; dwSenseccosense)
          Accumulate wDif weighted with a hypernym relation factor
          hypSim = hypSim + wDif  $\times$  hypFactor
          Reduce the size of the sense's sets csel sense and ccosense by wDif
        end if
        if there is a common occurrence in holset and ccosense then
          Compute wDif = min (dwSensecsel sense; dwSenseccosense)
          Accumulate wDif weighted with a holonym relation factor holSim =
          holSim + wDif  $\times$  holFactor
          Reduce the size of the sense's sets csel sense and ccosense by wDif
        end if
      end for
    end for
    Based on heuristics on value combinations of eq, synSim, hypSim and
    holSim decide the relation holding (superconcept, subconcept, equivalence or
    not related) between the concepts. If there is not enough evidence, only a
    related relation is set
    OUTPUT: is an approximation of the relation holding between the csel and
    cco and a confidence value
  end for
end for

```

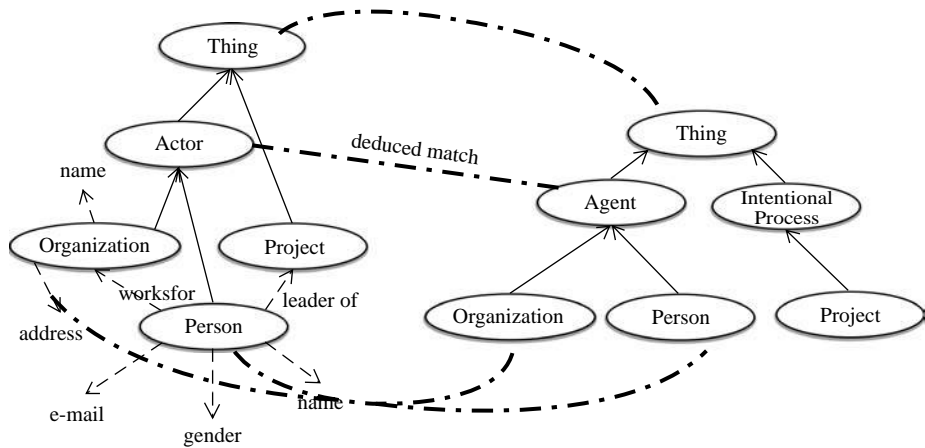
---

**Figure 8.**  
Set-based relation  
discovery

---

hierarchy between two other concepts in  $C_{sel}$  which do have a matching concept in co, and if there is a non-matched concept in co in the same relative hierarchical position, then we can deduce that there is likely to be a relation between this two concepts. Figure 9 shows an example where the actor-agent match is deduced. For these cases, we only state that there is evidence of a relation between these two concepts, but we do not specify which is the specific relation holding.

*4.4.4 Concept relations computation.* For computing the relation produced by our approach, we combine the relations obtained in the previously presented relation computations with the similarity measure. If relations coincide the result is trivial, if conflicts occur then depending on the combination of the similarity measure values we



**Figure 9.**  
Matching deduction  
example

decide heuristically if one of them should be favored. If there is not enough evidence to make a decision, we state that concepts are “related,” without any further explanation about the exact relation holding.

#### 4.5 Ontology coverage

Finally, we compute a measure of how much  $co$  matches the specified start ontology by measuring the similarity of each matching element over the total of expected matches:

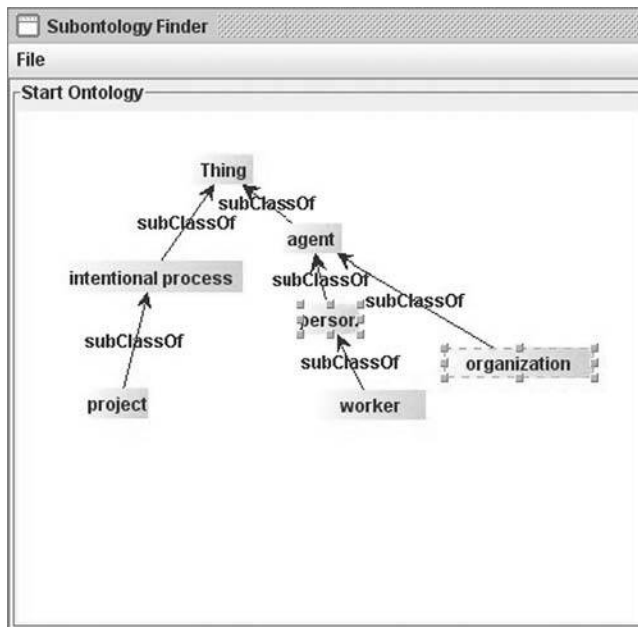
$$\text{coverage}(s; co) = \frac{\text{number of matches} \square \text{accumulated sim}}{jC_{sel}j}$$

### 5. Implementation

Currently, we have an implemented java prototype that allows performing Steps 1-4 from Figure 1. The prototype allows selecting an ontology from the local disk and displays it in a graph layout structure by using the JGraph ([www.jgraph.com](http://www.jgraph.com)) library as shown in Figure 10.

The engineer can then select a set of concepts  $C_{sel}$ . Once  $C_{sel}$  is specified, the labels of the concepts are extracted, tokenized and lemmatized and, by using WordNet (<http://wordnet.princeton.edu/>), the synonyms for them can be retrieved. Label words (and their synonyms if desired) will be used for a preselection of candidate ontologies. The pool of ontologies we are currently accessing for pre-selection of candidate ontologies is Swoogle ([swoogle.umbc.edu](http://swoogle.umbc.edu)). Ontologies having at least a (user defined) percentage of matching search terms will be retrieved for further analysis.

The results of the selection and of the search are presented (see screenshot in Figure 11). From the result list, the engineer can select the ones to be further analyzed. After this selection, the analysis process can be started, the selected ontologies are retrieved and, if accessible, parsed and the match, as presented in Section 4, is computed. The results are displayed ranked by coverage and the selection of any of them makes the tool display it in a graph layout view and highlights the matching concepts as can be seen in the screenshot shown in Figure 12. Additionally, our prototype also allows inspecting the matching details of any matching concept by double clicking on it.



**Figure 10.**  
Screenshot – start  
ontology selection

## 6. Evaluation

Since the first part of our solution aims to detect matching candidate ontologies, we employed the EON 2005 (Euzenat *et al.*, 2005) benchmark suite for evaluating this matching part. This benchmark is based on a reference ontology in the bibliography domain and a number of alternative ontologies of the same domain for which alignments are provided. The benchmark's tests are divided in groups as follows:

- (1) simple tests, compare the ontology to itself or to one from a different domain;
- (2) systematic tests, obtained by discarding some features from the reference ontology, e.g. names, hierarchy, relations, etc.; and
- (3) real life tests, including four ontologies about bibliographic references found on the web.

For the tests, we considered  $C_{sel}$  to contain all concepts in the start ontology (which is the reference ontology proposed in this benchmark suite) and ran it against all other benchmark candidate ontologies with a radius  $r = 2$ . Some preliminary tests showed that  $r > 2$  do not produce substantially better results, but this remains to be investigated more thoroughly.

Table I shows the precision, recall, fall out and  $f$ -measure values as known from information retrieval. These encouraging matching results were computed by comparing the results obtained by our approach with the golden standard as described in the evaluation benchmark suite guidelines. Considering and analyzing the characteristics of each ontology presented in Euzenat *et al.* (2005), the cases where labels or names do not carry meaningful English words are the ones where our approach has difficulties as can be seen in tests 201, 202, 248-266, or where only French

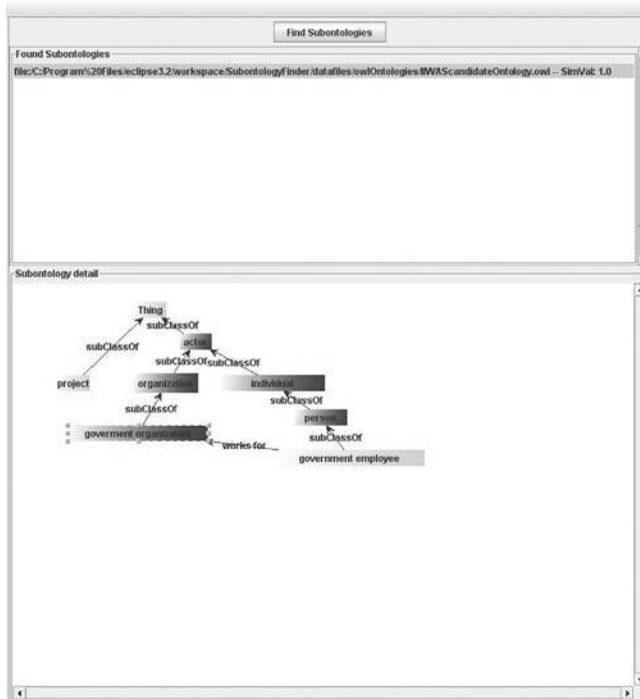


**Figure 11.**  
Screenshot – selected  
concepts and candidate  
ontologies

labels are used as 206, 207, and 210. This was expected as lexical information is one of the major criteria used for detecting matches. In other cases, with flattened hierarchies like in tests 221, 232, 241, etc. without properties attached to concepts as in tests 209, 228, 239, 246, etc. or with a different hierarchical structure as in tests 240, 247, etc. our approach still finds matches as expected. In cases where the domain is completely without overlap as in test 102, or with only partial overlap like in tests 205, 302, 304, etc. the precision and recall numbers show this. Misleading results as seen in test 103 occur in most tests due to the fact that we also search for matching concepts in the imported ontologies which is not considered in the provided golden standard.

An important factor to consider is that we do not only compute exact matches, but also others having a different logical relation as the equivalence, so the number of pairs our approach finds is higher than the ones presented in the golden standards. For the computation of this evaluation, we only took the equivalence matches and we disregarded the matching similarity values, we only computed matching evidence vs non-matching evidence cases.

Finally, results of test 101 (self-test) present some inaccuracies due to the fact that in our current implementation we employ a filtering procedure in order to reduce the number of needed pair-comparisons. We are confident this small deviation will not affect our later results.



**Figure 12.**  
Matching ontologies list  
and ontology details

The presented evaluation shows that our approach performs acceptably good in a variety of cases compared with the results of other approaches, some of them tailored to specific scenarios, available in Euzenat *et al.* (2005). Although there are specialized approaches with higher results in some specific cases, our general (mean) result show that our approach is performant and flexible enough to find the matches required in order to continue with the module extraction process of our approach.

## 7. Conclusions and future work

In this paper, we presented an approach for supporting adaptive ontology re-use starting from a drafted ontology. Our algorithms use a novel set-based approach combined with existing matching approaches by taking into account the modeling perspective of the drafted as well as of the analyzed existing ontologies. In this paper, we focus on the discovery and matching aspects of the presented approach.

Next steps in our planned work are to employ this approach for integrating datasources in the personal desktop, following the ideas presented in Halevy *et al.* (2006). Here, the aim is to first automatically propose an alignment of the ontologies describing the datasources in the desktop, so that the information contained in this datasources can, at least partially, be integrated. Then, based on different evidence such as user feedback and instances analysis, the alignments will be refined or corrected in a semi-automatic and iterative way so that at each iteration results will get more accurate increasing user satisfaction.



IJWIS  
4,2

212

Ndgo Test	Precision	Recall	Similarity finder <sup>a</sup> Fall out	F-score	Overall
101	0.868	1.000	0.132	0.930	0.848
102	1.000	1.000	0.000	1.000	1.000
103	0.868	1.000	0.132	0.930	0.848
104	0.861	0.939	0.139	0.899	0.788
201	0.400	0.061	0.600	0.105	0.000
202	0.000	0.000	1.000	0.000	0.000
203	0.829	0.879	0.171	0.853	0.697
204	0.875	0.848	0.125	0.862	0.727
205	0.556	0.152	0.444	0.238	0.030
206	0.400	0.083	0.600	0.138	0.000
207	0.400	0.083	0.600	0.138	0.000
208	0.828	0.727	0.172	0.774	0.576
209	0.333	0.061	0.667	0.103	0.000
210	0.400	0.083	0.600	0.138	0.000
221	0.808	0.636	0.192	0.712	0.485
222	0.848	0.966	0.152	0.903	0.793
223	0.806	0.758	0.194	0.781	0.576
224	0.868	1.000	0.132	0.930	0.848
225	0.853	0.879	0.147	0.866	0.727
228	0.815	0.667	0.185	0.733	0.515
230	0.808	0.840	0.192	0.824	0.640
231	0.868	1.000	0.132	0.930	0.848
232	0.808	0.636	0.192	0.712	0.485
233	0.783	0.545	0.217	0.643	0.394
236	0.815	0.667	0.185	0.733	0.515
237	0.844	0.931	0.156	0.885	0.759
238	0.806	0.758	0.194	0.781	0.576
239	0.808	0.724	0.192	0.764	0.552
240	0.792	0.576	0.208	0.667	0.424
241	0.783	0.545	0.217	0.643	0.394
246	0.808	0.724	0.192	0.764	0.552
247	0.792	0.576	0.208	0.667	0.424
248	0.000	0.000	1.000	0.000	0.000
249	0.000	0.000	1.000	0.000	0.000
250	0.000	0.000	1.000	0.000	0.000
251	0.000	0.000	1.000	0.000	0.000
252	0.000	0.000	1.000	0.000	0.000
253	0.000	0.000	1.000	0.000	0.000
254	0.000	0.000	1.000	0.000	0.000
257	0.000	0.000	1.000	0.000	0.000
258	0.000	0.000	1.000	0.000	0.000
259	0.000	0.000	1.000	0.000	0.000
260	0.000	0.000	1.000	0.000	0.000
261	0.000	0.000	1.000	0.000	0.000
262	0.000	0.000	1.000	0.000	0.000

Table I.  
Evaluation results*(continued)*

Table I.

Ndgo Test	Precision	Recall	Similarity finder <sup>a</sup>		Overall
			Fall out	F-score	
265	0.000	0.000	1.000	0.000	0.000
266	0.000	0.000	1.000	0.000	0.000
301	0.857	0.273	0.143	0.414	0.227
302	0.667	0.174	0.333	0.276	0.087
303	0.556	0.278	0.444	0.370	0.056
304	0.714	0.500	0.286	0.588	0.300
<i>H</i> -mean	0.761	0.418	–	–	–

**Note:** <sup>a</sup>Without comparing the measure values

In another line of work, we are also evaluating the inclusion of other lexical resources like FrameNet (framenet.icsi.berkeley.edu/), and expand our available test sets of ontological resources and repositories as well on improving and further testing our presented match approach.

## References

- Alani, H. (2006), "Position paper: ontology construction from online ontologies", *WWW'06: Proceedings of the 15th International Conference on World Wide Web*, ACM Press, New York, NY, pp. 491-5.
- Bontas, E.P., Mochol, M. and Tolksdorf, R. (2005), "Case studies on ontology reuse", paper presented at 5th International Conference on Knowledge Management (IKnow05), Graz.
- Bouquet, P., Serafini, L. and Zanobini, S. (2005), "Semantic coordination of heterogeneous classification schemas", in Staab, S. and Stuckenschmidt, H. (Eds), *Peer-to-peer and Semantic Web*, Springer, Heidelberg.
- Ding, Y., Lonsdale, D., Embley, D.W., Hepp, M. and Xu, L. (2007), "Generating ontologies via language components and ontology reuse", *Proceedings of the 12th International Conference on Applications of Natural Language to Information Systems (NLDB07)*, Springer, Paris.
- Doan, A., Domingos, P. and Halevy, A. (2003), "Learning to match the schemas of data sources: a multistrategy approach", *Mach. Learn.*, Vol. 50 No. 3, pp. 279-301.
- Euzenat, J. and Shvaiko, P. (2007), *Ontology Matching*, Springer, Heidelberg (DE).
- Euzenat, J., Stuckenschmidt, H. and Yatskevich, M. (2005), "Introduction to the ontology alignment evaluation 2005", in Ashpole, B., Ehrig, M., Euzenat, J. and Stuckenschmidt, H. (Eds), *Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies, CEUR, Banff, Canada*, Vol. 156.
- Fellbaum, C. (1998), *Wordnet: An Electronic Lexical Database*, MIT Press, Cambridge, MA.
- Galley, M. and McKeown, K. (2003), "Improving word sense disambiguation in lexical chaining", in Gottlob, G. and Walsh, T. (Eds), *IJCAI*, Morgan Kaufmann, San Francisco, CA, pp. 1486-8.
- Grau, B.C., Horrocks, I., Kazakov, Y. and Sattler, U. (2007), "Just the right amount: extracting modules from ontologies", *WWW'07: Proceedings of the 16th International Conference on World Wide Web*, ACM Press, New York, NY, pp. 717-26.

- Halevy, A., Franklin, M. and Maier, D. (2006), "Principles of dataspace systems", *PODS'06: Proceedings of the Twenty-fifth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ACM, New York, NY, pp. 1-9.
- Hirst, G. and St-Onge, D. (1997), "Lexical chains as representation of context for the detection and correction of malapropisms", *WordNet: An Electronic Lexical Database and Some of Its Applications*, The MIT Press, Cambridge, MA.
- Hovy, E., Lin, C.-Y. and Zhou, L. (2005), "A be-based multi-document summarizer with sentence compression", *Proceedings of Multilingual Summarization Evaluation (ACL 2005 Workshop)*, Ann Arbor, MI.
- Kim, J., Jang, M., Ha, Y.-G., Sohn, J.-C. and Lee, S.J. (2005), "Moa: owl ontology merging and alignment tool for the semantic web", *Innovations in Applied Artificial Intelligence: 18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2005, Bari, Italy, June 22-24*, Springer, Heidelberg, p. 722.
- Koster, C.H.A. (2003), "Head/modifier pairs for everyone", *SIGIR'03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, New York, NY, p. 466.
- Lesk, M. (1986), "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone", *SIGDOC'86: Proceedings of the 5th Annual International Conference on Systems Documentation*, ACM Press, New York, NY, pp. 24-6.
- Mitra, P., Noy, N.F. and Jaiswal, A. (2005), "Omen: a probabilistic ontology mapping tool", in Gil, Y., Motta, E., Benjamins, V.R. and Musen, M.A. (Eds), *International Semantic Web Conference*, Lecture Notes in Computer Science, Vol. 3729, Springer, Heidelberg, pp. 537-47.
- Noy, N.F. and Musen, M.A. (2003), "The prompt suite: interactive tools for ontology merging and mapping", *Int. J. Hum.-Comput. Stud.*, Vol. 59 No. 6, pp. 983-1024.
- Rector, A.L. (2003), "Modularisation of domain ontologies implemented in description logics and related formalisms including OWL", *K-CAP'03: Proceedings of the 2nd International Conference on Knowledge Capture*, ACM Press, New York, NY, pp. 121-8.
- Silber, H.G. and McCoy, K.F. (2002), "Efficiently computed lexical chains as an intermediate representation for automatic text summarization", *Computational Linguistics*, Vol. 28 No. 4, pp. 487-96.
- Teich, E. and Fankhauser, P. (2004), "WordNet for lexical cohesion analysis", in Sojka, P., Pala, K., Smrz, P., Fellbaum, C. and Vossen, P. (Eds), *Proceedings of the 2nd Global WordNet Conference, Brno, Czech Republic*.
- Uschold, M., Healy, M., Williamson, K., Clark, P. and Woods, S. (1998), "Ontology reuse and application", in Guarino, N. (Ed.), *Proc. of the Int. Conf on Formal Ontology in Information Systems – FOIS'9*, IOS Press, Amsterdam, pp. 179-92.

### Corresponding author

Rodolfo Stecher can be contacted at: [stecher@L3s.de](mailto:stecher@L3s.de)

**This article has been cited by:**

1. G. López, V. Custodio, J.I. Moreno, M. Sikora, P. Moura, N. Fernández. 2015. Modeling Smart Grid neighborhoods with the ENERsip ontology. *Computers in Industry* **70**, 168-182. [[CrossRef](#)]