Towards Automatic Merging of Domain Ontologies: The HCONE-merge approach

Konstantinos Kotis, George A. Vouros, Konstantinos Stergiou

Department of Information & Communications Systems Engineering,
University of the Aegean,
Karlovassi, Samos,
83200, Greece
{kkot, georgev, konsterg}@aegean.gr

Abstract

Latest research efforts on the semi-automatic coordination of ontologies "touch" on the mapping /merging of ontologies using the whole breadth of available knowledge. Addressing this issue, this paper presents the HCONE-merge approach, which is further extended towards automating the merging process. HCONE-merge makes use of the intended informal meaning of concepts by mapping them to WordNet senses using the Latent Semantic Indexing (LSI) method. Based on these mappings and using the reasoning services of Description Logics, HCONE-merge automatically aligns and then merges ontologies. Since the mapping of concepts to their intended meaning is an essential step of the HCONE-merge approach, this paper explores the level of human involvement required for mapping concepts of the source ontologies to their intended meanings. We propose a series of methods for ontology mapping (towards merging) with varying degrees of human involvement and evaluate them experimentally. We conclude that, although an effective fully automated process is not attainable, we can reach a point where ontology merging can be carried out efficiently with minimum human involvement.

Keywords: Ontology mapping, Ontology merging, Ontology coordination, Latent Semantic Indexing (LSI), WordNet

1. Introduction

Ontologies have been realized as the key technology to shaping and exploiting information for the effective management of knowledge and for the evolution of the Semantic Web and its applications. In such a distributed setting, ontologies establish a common vocabulary for community members to interlink, combine, and communicate knowledge shaped through practice and interaction, binding the knowledge processes of creating, importing, capturing, retrieving, and using knowledge. However, it seems that there will always be more than one ontology even for the same domain [22], [23]. In such a setting, where different conceptualizations of the same domain exist, information services must effectively answer queries, bridging the gaps between conceptualisations of the same domain. Towards this target, networks of semantically related information must be created at-request. Therefore, coordination (i.e. mapping, alignment, merging) of ontologies is a major challenge for bridging the gaps between agents (software and human) with different conceptualizations.

There are many research works on the mapping and merging of ontologies. These works exploit lexical (or syntactic), structural, semantic (or domain) knowledge and matching heuristics. Recent approaches aim to exploit all these types of knowledge and further capture the intended meanings of terms by means of heuristic rules (e.g. [5], [7], and [12]). The HCONE-merge approach to ontology merging [8] [9] exploits all the above-mentioned types of knowledge. This approach gives much emphasis on "uncovering" the intended informal meaning of concepts specified in an ontology by mapping them to WordNet senses. WordNet senses realize the informal, human-oriented intended meaning of the corresponding concepts. To compute these mappings, HCONE-merge uses the Latent Semantic Indexing (LSI) method. Exploiting the mappings proposed by LSI, the merging method we introduce translates concept definitions of the source ontologies to a common vocabulary and finally, merges the translated definitions by means of specific merging rules and description logics' reasoning services.

The HCONE-merge approach requires humans to validate the computed intended meaning of every concept in the ontology. Since this process is quite frustrating and error-prone, even for small ontologies, we need to investigate the required human involvement for mapping concepts to their intended meaning efficiently. The ultimate achievement would be to fully automate the mapping of concepts to their intended meaning, and consequently to fully automate merging. Towards this goal, the paper investigates a series of novel techniques and heuristics for ontology mapping and merging, with varying human involvement. The paper concludes that a fully automated ontology merging process is far from realistic, since there must always be a minimum set of human decisions present, something that has been also suggested by other lines of research (e.g. [16] [23]).

This paper is structured as follows: Section 2 describes the ontology merging problem. Section 3 provides background information concerning LSI and WordNet and section 4 describes the HCONE-merge approach. Section 5 reports on related work, identifies criteria and compares HCONE-merge with other approaches in terms of the identified criteria. Section 6 describes methods towards automating the process of mapping and merging, and finally, section 7 provides an evaluation of the proposed methods using real-life ontologies and ontologies that have been used by closely related approaches and tools. Section 8 discusses the mapping of domain relations and the influence of these mappings on the computation of concept's mapping. Section 9 concludes the paper and points out the advantages and disadvantages of the HCONE-merge approach.

2. The Ontology Merging Problem (OMP)

In order to have a common reference to other approaches, we formulate the ontology merging problem by means of definitions and terms used in [7].

An ontology is considered to be a pair $O=(S, A)$, where S is the ontological signature describing the vocabulary (i.e. the terms that lexicalize concepts and relations between concepts) and A is a set of ontological axioms, restricting the intended meaning of the terms included in the signature. In other words, A includes the formal definitions of concepts and relations that are also lexicalized by natural language terms in S. This is a slight variation of the definition given in [7], where S is also equipped with a partial order, based on the inclusion relation between concepts. In our definition, conforming to description logics' terminological axioms, inclusion relations are ontological axioms included in A. It must be noticed that in this paper we only deal with inclusion and equivalence relations among concepts.

Ontology mapping from ontology $O_1 = (S_1, A_1)$ to $O_2 = (S_2, A_2)$ is considered to be a morphism $f{:}S_1 \rightarrow S_2$ of ontological signatures such that $A_2 \models f(A_1)$, i.e. all interpretations that satisfy O_2's axioms also satisfy O_1's translated axioms. Consider for instance the ontologies depicted in **Fig. 1**: Given the morphism *f* such that $f(O_1\text{-}Infrastructure) = O_2\text{-}Facility$ and $f(O_1\text{-}Transportation) = O_2\text{-}Transportation\ System$, it is true that $A_2 \models \{f(O_1\text{-}Transportation) \sqsubseteq f(O_1\text{-}Infrastructure)\}$, therefore *f* is a mapping. Given the morphism *f'*, such that $f'(O_1\text{-}Infrastructure) = O_2\text{-}Transportation\ System$ and $f'(O_1\text{-}Transportation) = O_2\text{-}Transportation\ Means$, it is not true that $A_2 \models \{f'(O_1\text{-}Transportation) \sqsubseteq f'(O_1\text{-}Infrastructure)\}$, therefore *f'* is not a mapping.

Fig. 1. Example Ontologies

However, instead of a function, we may articulate a set of binary relations between the ontological signatures. Such relations can be the inclusion ($\sqsubseteq$) and the equivalence ($\equiv$) relations. For instance, given the ontologies in **Fig. 1**, we can say that $O_1\text{-}Transportation \equiv O_2\text{-}Transportation\ System$, $O_1\text{-}Installation \equiv O_2\text{-}Facility$ and $O_1\text{-}Infrastructure \sqsubseteq O_2\text{-}Facility$. Then we have indicated an alignment of the two ontologies and we can merge them. Based on the alignment, the merged ontology will be ontology O_3 in **Fig. 1**. It holds that $A_3 \models A_2$ and $A_3 \models A_1$.

Looking at **Fig. 1** in an other way, we can consider O_3 to be part of a larger *intermediate ontology* and define the alignment of ontologies O_1 and O_2 by means of morphisms $f_1: S_1 \rightarrow S_3$ and $f_2: S_2 \rightarrow S_3$, i.e. by means of their mapping to the intermediate ontology. Then, the merging of the two ontologies [5] is the minimal union of ontological vocabularies and axioms with respect to the intermediate ontology where ontologies have been mapped.

Therefore, the ontologies merging problem (OMP) can be stated as follows: Given two source ontologies O_1 and O_2 find an alignment between them by mapping them to an intermediate ontology, and then, get the minimal union of their (translated) vocabularies and axioms with respect to their alignment.

3. WordNet and Latent Semantic Indexing

Before proceeding to the description of the HCONE-merge method, let us give a brief overview of WordNet and LSI, which are key-technologies for the realization of the HCONE-merge approach.

3.1. WordNet

WordNet [13] is a lexicon based on psycho-linguistic theories. It contains information about nouns, verbs, adverbs, and adjectives, organized around the notion of a synset. A synset is a set of words with the same part-of-speech that can be interchanged in a certain context. For example {*facility; installation*} form a synset because they can be used to refer to the same concept. A synset is often further described by a gloss: e.g. "*created to provide a particular service*". Semantic relations among synsets include among others the synonymy, hyper(hyp)onymy, meronymy and antonymy relations. WordNet (version 1.4) contains more than 83.800 words, 63.300 synsets and 87.600 links between concepts.

Fig. 2. WordNet information for concept *Facility*

As **Fig. 2** shows**,** WordNet provides lexical and semantic information concerning a word. Specifically, concerning the word *Facility,* **Fig. 2** shows the 5 WordNet synsets (senses) of

Facility, and the hyperonyms of the terms that lexicalize each sense. For instance, the first sense of *Facility* is lexicalized by the synonyms *Facility* and *Installation,* and is defined to be something that has been "*created to provide a particular service*". Furthermore, the hyperonyms of the terms that lexicalize this sense are: "*artifact, artifact*", "*object, physical object*", "*entity, something*".

3.2. LSI

Latent Semantic Indexing (LSI) [2] is a vector space technique originally proposed for information retrieval and indexing. It assumes that there is an underlying latent semantic space that it estimates by means of statistical techniques using an association matrix ($n \times m$) of term-document data. Latent Semantic Analysis (LSA) computes the arrangement of a k-dimensional semantic space to reflect the major associative patterns in the data. This is done by deriving a set of k uncorrelated indexing factors. These factors may be thought of as artificial concepts whose lexicalization is not important for LSI. Each term and document is represented by its vector of factor values, indicating its strength of association with each of these underlying concepts. In other words, the meaning of each term or document is expressed by k factor values, or equivalently, by the location of a vector in the k-space defined by the factors. Then, a document is the (weighted) sum of its component term vectors. The similarity between two objects (e.g. between two documents) is computed by means of the dot product between the corresponding representation vectors.

For the computation of the k factors LSI employs a two-mode factor analysis by decomposing the original association matrix into three other matrices of a very similar form. This is done by a process called "singular value decomposition (SVD)". This results in a breakdown of the original term-document relationships into linearly independent factors. Some of these factors are not significant and are ignored. The resulting k factors specify the dimensionality of the semantic space.

By virtue of dimension reduction from the N terms space to the k factors space, where k<N, terms that did not actually appear in a document may still end up close to the document, if this is consistent with the major patterns of association in the data.

When one searches an LSI-indexed database of documents, it provides a query (i.e. a pseudo-document), which is a list of terms. As already pointed, a document is represented by the weighted sum of its component term vectors. The similarity between two documents is computed by means of the dot product between the corresponding representation vectors. Doing so, LSI returns a set of graded documents, according to their similarity to the query.

For instance, let us consider a set of 5 documents (described here only by their titles) referring to the baking of bread and pastries.

D1:How to <u>Bake</u> <u>Bread</u> Without <u>Recipes</u>
D2:The Classic Art of Viennese <u>Pastry</u>
D3:Numerical <u>Recipes</u>: The Art of Scientific Computing
D4:<u>Breads</u>, <u>Pastries</u>, <u>Pies</u> and <u>Cakes</u>: Quantity <u>Baking</u> <u>Recipes</u>
D5:<u>Pastry</u>: A Book of Best French <u>Recipes</u>

To compute the semantic space, the method builds the association matrix that associates terms that occur in documents with the documents themselves. This can be done by considering only the stems of those terms that do not belong in a stop-words list. Doing so, terms such as *bake* and *baking* should be considered as the term *bake*. Given the documents above, this step will produce the following list of terms:

T1: bak(e, ing)
T2: recipes
T3: bread
T4: cake
T5: pastr(y, ies)
T6: pie

Having obtained the list of terms, the next step involves the construction of the association matrix *A1* (*Terms*×*Documents*) that contains the frequency of term occurrences within each document. The value of an entry of matrix *A1* in **Fig.3** specifies the frequency occurrence of a term in a document. In case this value is 0, then the term does not appear in the corresponding document. For instance, the first line of the matrix *A1* in **Fig. 3** represents the fact that that the term *T1 (bak(e, ing))* occurs in documents *D1* and *D4*.

Fig. 3. A matrix in LSI method: 2 phases of computation

Having constructed the matrix, one may query using the single keyword "*baking"*. The query is written in the form of a vector *q = (1, 0, 0, 0, 0, 0)*, where the value *1* in the first position of the vector represents the term *baking* from the list of the *6* terms identified.

LSI, via SVD, computes the decomposition of the association matrix and "identifies" the *k* significant factors for the representation of terms and documents. Using this vector representation of terms and documents in terms of the *k*-factors, one can produce the matrix *A2* shown in **Fig. 3** that "approximates" *A1*: *A2* associates terms with documents semantically. The production of such an approximation is the major strength of LSI, since it is believed that the original term space is unreliable. The approximation expresses what is reliable and important in the underlying use of terms as document referents [2].

As it is depicted in **Fig. 3**, there are no zero values in matrix *A2* indicating that we can obtain a similarity value for every term in a query. More interesting is the fact that some values are negative, indicating that there is a very large distance between a term and a document. Using matrix *A2* to answer the query "*baking"*, the grades returned are: *0.5181, -0.0332, 0.0233, 0.5064, -0.0069.* I.e., both documents *D1* and *D4* have been rated with a high grade.

4. The HCONE-merge method of solving the OMP

Given two source ontologies, the HCONE-merge method finds a semantic morphism between each of these two ontologies and the so-called "hidden intermediate" ontology. As **Fig.4** shows, WordNet plays the role of an "intermediate" where concepts of the source ontologies are being mapped through the semantic morphism (*s-morphism*, symbolized by f_s). This morphism is computed by the Latent Semantic Indexing (LSI) method and associates ontology concepts with WordNet senses.

Fig. 4.The HCONE approach towards the OMP

It must be noticed that we do not consider WordNet to include any intermediate ontology, as this would be very restrictive for the specification of the original ontologies (i.e. the method would work only for those ontologies that preserve the inclusion relations among WordNet senses). Actually, HCONE-merge assumes that the intermediate ontology is "somewhere there" and constructs this ontology while mapping concepts to the WordNet senses.

In fact, WordNet can be replaced by any other lexicon, thesaurus or even a collection of documents that describe concepts of the source ontologies. We have used WordNet since it is a well-thought and widely available lexical resource with a large number of entries and semantic relations. We conjecture that any lexical resource that provides concepts' lexicalisations together with their informal intended meanings can be used as well.

The hidden intermediate ontology that it constructed during the mapping includes (a) a vocabulary with the lexicalizations of the specific senses of WordNet synsets corresponding to the ontologies' concepts, and (b) axioms that are the translated axioms of the source ontologies. As **Fig. 4** shows, having found the mappings to the hidden intermediate ontology, and having translated the source ontologies, these have been aligned and are then merged by means of the rename, merge, and classify actions.

4.1. Computing the s-morphism

As we have already mentioned, the first step in our approach is the mapping of ontologies O_1 and O_2 to the "*hidden intermediate*" ontology by means of semantic morphisms (*s-morphisms*) $f_1: S_1 \rightarrow S_3$ and $f_2: S_2 \rightarrow S_3$.

To compute the *s-morphism,* HCONE-merge uses the LSI method. In our case, the $n \times m$ association matrix comprises the *n* more frequently occurring terms of the *m* WordNet senses that the algorithm focuses on (what constitutes the focus of the algorithm is explained in **Fig. 5**).

The steps of the algorithm for finding the semantic morphism are shown in **Fig. 5**:

Fig. 5. The algorithm for computing the *s-morphism*

For example, **Fig. 6** shows the major steps for the computation of the *s-morphism* for the concept "*Facility*" of ontology O_2 depicted in **Fig. 1**. Concept "*Facility*" is associated with the five WordNet senses whose meanings range from "*something created to provide a service*" to "*a room equipped with washing and toilet facilities*" (see Step 1,2,3). These senses constitute the focus of the algorithm for the concept "*Facility*". The terms-senses association matrix (according to the Step 4 of the algorithm) for this concept is a 93×5 matrix (93 terms were found after applying filtering techniques to the 5 senses) containing values that correspond to the frequency of the terms' occurrence is each of the 5 senses within algorithm's focus. The query string is constructed by the query terms (Step 5 of the algorithm), assigning the value "*1*" (one) in the corresponding positions of a 93-positions vector that corresponds to the 93 terms of the association matrix. Having all the necessary data, LSI returns (see Step 6) the graded senses in the algorithm's focus. In this case, as **Fig.6** shows, the first sense has the largest grade and is hypothesized to be the one expressing the intended meaning of the concept "*Facility*" of ontology O_2.

Fig. 6. A running case for computing the mapping of the concept "*Facility*"

It must be emphasized that although LSI exploits structural information of ontologies and WordNet, it ends up with semantic associations between terms.

The algorithm is based on assumptions that influence the mappings produced:

- Currently, concept names lemmatization and morphological analysis is not sophisticated. The algorithm finds a lexical entry that matches a slight variation of the given concept name. However, in another line of research we produce methods for matching concept names based on a 'core set' of characters [24]. It must be noted that for ontologies concerning fine-grained domains, some of the high-technical concepts contained in these ontologies may not have a lexical entry in WordNet. In such a case, as already pointed out, one may use other lexicons, thesauruses, or lexical resources instead of, or in conjunction to WordNet. HCONE-merge is not restricted to the mandatory use of WordNet.
- Most multi-word terms have no senses in WordNet, thus we can only compute the intended meaning for each component-word of the term. This gives a partial indication of the intended meaning of the whole term. Currently, we assume that a multi-word term lexicalizes a concept that is related to the concepts that correspond to the words comprising it. In general, in case a multi-word term C has no lexical entry in WordNet, then the term C is associated with concepts H_n, $n=1,2\ldots$ corresponding to the single words comprising it. Then, C is considered to be mapped to a virtual concept Cw of the hidden intermediate ontology, and each concept H_n is considered to be included in the ontological signature of the intermediate ontology. For instance, the concept lexicalized by "*Transportation Means*" is considered to be related to the concepts lexicalized by "*Transportation*" and "*Means*". Specifically, it is assumed that the concept lexicalised by the right-most word of a multi-word term subsumes the concept lexicalised by the multi-word term [19]. I.e., the concept "*Means*" subsumes the concept "*Transportation Means*". The concepts that are lexicalized by the rest of the words of a multi-word term can be related with this term by any domain relation. For instance, given that *"Means"* subsumes *"TransportationMeans*", and that the domain relation to "*Transportation"* is "*function*", the following axiom holds: $TransportationMeans \sqsubseteq Means \sqcap function.Transportation$.

This treatment of multi-word terms is motivated by the need to reduce the problem of mapping these terms to the mapping of single terms. Doing so, we can exploit the translated formal definitions of multi-word terms by means of description logics reasoning services for testing equivalence and subsumption relations between the concepts' definitions during the mapping and merging of ontologies.

In related lines of research, the mapping of multiword terms is addressed mainly by syntactic methods [18] [10]. In [10] for instance, words that lexicalize concepts of the source ontology are matched to words of each term of the target ontology. Every matched pair has a score that represents the ratio of the number of the words matched with regard to the total number of words. Then, for each term, among all its pairs, only the highest graded pair is recorded as matched. Doing so, pairs such as "*meeting-place*" and "*place-of-meeting*", as well as "*written-by*" and "*wrote*" can be found. Closer to our work is an additional technique described in [10], described under the title "synset matching". According to this, the meanings of the words found in a multi-word term are represented by means of WordNet synsets. For each word in each term, if this word corresponds to a WordNet entry, then it must belong to one of the corresponding synsets. The two terms which have the largest number of common synsets are recorded as a matched pair. For instance, terms such as "*auto-care*" and "*car-maintenance*" can be matched.

In contrast to the above-mentioned approaches that involve estimating the similarity among labels using mainly syntactic similarity measures, the treatment of multi-word terms in HCONE-merge involves not only syntactic, but also (and mainly) semantic knowledge. Semantic knowledge is captured by means of the *s-morphism* computed for the component words of a multi-word term, as well as by means of the description logics axioms produced.

- The performance of the algorithm is related to assumptions concerning the information that has to be used for the computation of (a) the semantic space, and (b) the query terms. This is thoroughly examined in the paragraphs that follow.
- The implementation of LSI that we are currently using, as pointed out by the developers[1], works correctly when the $n \times m$ matrix utilized has more than 4 and less than 100 senses (i.e. $4 \leq m \leq 100$). In case there are fewer than 4 senses, we extend the semantic space with additional information.

The semantic space is constructed by terms in the vicinity of the senses $S_1, S_2, \ldots S_m$ that are in the focus of the algorithm for a concept *C*. Therefore, we have to decide what constitutes the vicinity of a sense for the calculation of the semantic space. In an analogous way we have to decide what constitutes the vicinity of an ontology concept for the calculation of the query string.

Terms that can be included in the semantic space include:

Sp1. The term *C'* that corresponds to *C*. *C'* is a lexical entry in WordNet that is a linguistic variation of *C* (as described in **Fig. 5**).
Sp2. Terms that appear in *C'* WordNet senses $S_1, S_2, \ldots S_m$.
Sp3. Terms that constitute hyperonyms / hyponyms of each *C'* sense.
Sp4. Terms that appear in hyper(hyp)onyms of *C'* senses.

Terms that can be included in the query string include:

Q1. The primitive super-concepts of concept C.
Q2. Concepts that subsume *C* and are immediate super-concepts of *C*.
Q3. Concepts that are immediate sub-concepts of *C*.
Q4. Concepts that are related to *C* via domain specific relations.
Q5. The most frequent terms in WordNet senses that have been associated with the concepts directly related to *C* via inclusion and equivalence relations.

The goal is to specify the vicinity of a concept and the vicinity of each sense in a generic and domain-independent way so as to compute valid mappings of concepts to WordNet senses without "distracting" LSI with information that is comprised by terms that are not in the domain of the ontology. Experiments imply that the *s-morphism* computation algorithm must consider senses and terms that are "close" to the intended meaning of the concepts in the hidden intermediate ontology, otherwise what we may call "semantic noise" can distract computations. Specifically, given terms that are not relevant to the domain of an ontology SVD may compute factors whose meaning do not represent the meaning of terms and documents adequately. However, since SVD computes what is reliable and important in the underlying use of terms as document referents, there must be a large percentage of terms irrelevant to the given ontology. Experiments showed that by reducing the amount of irrelevant information in the semantic space we actually achieved to get more hits. This

[1] KnownSpace Hydrogen License: This product includes software developed by the Know Space Group for use in the KnownSpace Project (http://www.knownspace.org)

happens when the semantic space includes *Sp4*, the query string includes *Q5* and the WordNet senses that have been associated with the concepts directly related to *C* do not represent the intended meanings of these concepts. Experiments using various ontologies have shown that we can achieve approximately 70% precision in mapping concepts to WordNet senses, if the vicinity of the senses that are in the focus of the algorithm include information *Sp1, Sp2, Sp3,* specified above, and the vicinity of the ontology concepts include information *Q1, Q2, Q3,* and *Q4*. *Q5* can further increase the precision of the method, if the WordNet senses that have been associated with the concepts directly related to *C* do represent the intended meanings of these concepts.

Using the algorithm described in **Fig. 5**, each ontology concept is associated with a set of graded WordNet senses. The highest graded sense expresses the most possible informal meaning of the corresponding concept. This sense is assumed to express the intended meaning of the concept specification and can be further validated by a human. In case a human indicates a sense to be the most preferable, then this sense is considered to capture the informal intended meaning of the formal ontology concept. Otherwise, the method considers the highest graded sense as the concept's intended interpretation.

4.2. Translation

Using the intended informal meanings of concepts, the proposed method of mapping/merging ontologies translates the formal definitions of concepts in a common vocabulary and merges the translated definitions using description logics reasoning services.

Given all the preferred mappings of concepts to WordNet senses, we have captured the intended meaning of ontology concepts. Using the intended meanings of the formal concepts, we construct an ontology $O^n=(S^n, A^n)$, $n=1,2$, where, S^n includes the lexicalizations of the senses associated to the concepts of the ontology $O_n=(S_n, A_n)$, $n=1,2$, and A^n contains the translated inclusion and equivalence relations between the corresponding concepts. Then, it holds that $A^n \models f_s(A_n)$ and the ontology $O^n=(S^n, A^n)$ with the corresponding associations from O_n *to* O^n, is a model of $O_n=(S_n, A_n)$, $n=1,2$. These associations define a mapping from O_n to O^n.

4.3. Merging of ontologies

Having discovered the associations between the ontology concepts and WordNet senses, the algorithm has found a semantic morphism between each of the source ontologies and the hidden intermediate ontology. Moreover, the source ontologies have been aligned. The actual construction of the intermediate ontology with the minimal set of axioms for both source ontologies results in the merging of these ontologies.

For instance, as shown in **Fig. 5**, given the morphisms produced, it holds that:

- For ontology O_1

$f_s(O_1\text{-}System) = System_1$,
$f_s(O_1\text{-}Installation) = Facility_1$,
$f_s(O_1\text{-}Infrastructure) = Infrastructure_1$, and
$f_s(O_1\text{-}Transportation) = TransportationSystem_1$

- For ontology O_2

$f_s(O_2\text{-}Facility) = Facility_1$,
$f_s(O_2\text{-}Transportation\ System) = TransportationSystem_1$, and

$f_s(O_2\text{-}Transportation\ Means) = TransporationMeans_w\ \{virtual\ concept\}$
$f_s(O_2\text{-}Means) = Means_1$
$f_s(O_2\text{-}Transportation) = Transportation_2$

The indices of the associated terms indicate the WordNet senses that provide the informal intended meanings of concepts. Notice that the intended meaning of concept Transportation in O_2 is different from the intended meaning of the homonym concept in O_1. Both ontologies are being translated using the corresponding WordNet senses' lexicalizations and are being merged taking into account the axioms of A^1 and A^2 (which are the translated axioms of A_1 and A_2 with respect to the computed *s-morphism*s).

The merging decisions are summarized in **Table 1**. We must emphasize that, as shown in **Table 1**, the semantic information concerning ontology concepts definitions is exploited by the description logics reasoner during merging.

Table 1. HCONE-Merge algorithm table summary

Concept & Relation Names[2]	**Concepts Mapping to WordNet Senses**[3]	**Action**
Match	No match	Rename concepts
Match	Match	Merge concept definitions
No match	Match	Merge concept definitions in a single concept named by the term lexicalizing their corresponding WordNet sense
No match	No match	Classify Concepts

The new ontology will incorporate the mappings of the original concepts and the translated axioms of O_1 and O_2, modulo the axioms of the intermediate ontology (see **Fig. 7**).

Fig. 7. *S-morphism* and the intermediate ontology

Therefore, the merged ontology is $O_m = (S_m, A_m)$, where:

$$S_m = \{System, facility, Means, Installation, Infrastructure, Transportation\ System, Transportation, Transportation\text{-}O_2, Transportation\ Means, exploit\},$$

$$\begin{aligned} A_m = \{ & Transportation \equiv TransportationSystem, \\ & Facility \equiv Installation,\ Infrastructure \sqsubseteq System \sqcap Facility, \\ & TransportationSystem \sqsubseteq Infrastructure, \\ & Means \sqsubseteq Facility, \\ & TransportationMeans \sqsubseteq Means \sqcap function.Transportation\text{-}O_2 \\ & \qquad \sqcap exploit.TransportationSystem\ \} \end{aligned}$$

It must be noticed that the concepts Transportation and Transportation System have the same intended meaning, and therefore are considered equivalent. According to **Table 1**, the merging of their formal definitions results to:

$$TransportationSystem \sqsubseteq Infrastructure \sqcap Facility$$

---

[2] Match in this case means linguistic match of the concept names from the two ontologies.
[3] Match means that both concepts have been mapped to the same WordNet sense

However, the description logics classification mechanism considers the axiom *TransportationSystem* $\sqsubseteq$ *Facility* to be redundant. Therefore O_3 contains only the axiom *TransportationSystem* $\sqsubseteq$ *Infrastructure*. By doing so, the merged ontology contains only the minimal set of axioms resulting from source ontologies mapping.

Furthermore, according to **Table 1**, the concept Transportation of O_2 will be renamed to Transportation-O_2 since it corresponds to a sense that is different to the sense of the homonym concept Transportation in O_1. This latter concept, based on the *s-morphism*, has been renamed to TransportationSystem.

An implementation of the merging method described so far is depicted in **Fig. 8**.

Fig. 8. The merging functionality integrated to HCONE. Merged concepts (e.g. FACILITY and INSTALLATION) are shown in the form Concept1+Concept2 (FACILITY+INSTALLATION) for presentation reasons

5. Merging methods related to HCONE-merge

As already explained in Section 4, ontology mapping has a close relation to the merging of ontologies. Mapping may utilize a reference ontology but it can also be point-to-point (non-mediated). In either case it must preserve the semantics of the mapped ontologies. The merging process takes into account the mapping results in order to resolve problems between the merged ontologies concerning name conflicts, taxonomy conflicts, etc.

To accomplish a mapping between two ontologies, an algorithm that will eventually discover the matching pairs of concepts is required. For instance, in HCONE, two concepts match if they have been mapped to the same sense of a WordNet synset. Matching can be distinguished in lexical, structural and semantic depending on the knowledge utilized and on the kind of similarity relation used [5]. Lexical matching involves the matching of ontology nodes' labels, estimating the similarity among nodes using syntactic similarity measures, as for instance in [11]. Minor name variations can lead the matching result astray. On the other hand, structural matching involves matching the neighbourhoods of ontology nodes, providing evidence for the similarity of the nodes themselves. Semantic matching explores the mapping between the meanings of concept specifications exploiting domain knowledge as well. Semantic matching specifies a similarity relation in the form of a semantic relation between the intensions of concepts [20]. Semantic matching may also rely on additional information from lexicons, thesaurus or reference ontologies incorporating semantic knowledge (mostly domain-dependent) into the process.

In contrast to techniques for merging non-populated ontologies, instance-based approaches exploit the set-theoretic semantics of concept definitions in order to uncover semantic relations among them. However, such approaches deal with specific (quite restricted) domains of discourse, rather than with the semantics of the statements themselves. Therefore, these approaches are useful in cases where information sources are rather stable (where the domain of discourse does not change frequently) or in cases where information is "representative" (e.g., as it is required in FCA-Merge [21]) for the concepts specified. Instance-based approaches can work complementary to techniques for matching concepts, thus, their combination with concept-based approaches could be very beneficial.

There is a variety of research efforts towards coordinating ontologies. According to [15] and [16] there is not a "best tool" or method, since there is not always the case that it will fit every user's or application's needs. To comment however on such efforts, we conjecture that specific criteria could be considered, such as:

a) The kind of mapping architecture they provide: (a) point-to-point mapping or mediated mapping, (b) top-down or bottom up mapping, considering techniques applied to the intensions of concepts (non-populated ontologies) or to the extensions of concepts (populated ontologies), respectively.
b) The kind of knowledge (lexical, structural, semantic) used for node matching, i.e. (a) techniques that are based on the syntax of labels of nodes and on syntactic similarity measures, (b) techniques that rely on structural information about ontologies, and (c) techniques that are based on the semantic relations of concepts and on semantic similarity measures.
c) The type of result produced: For instance, a mapping between two ontologies or/and a merged ontology.
d) Additional information sources consulted during the mapping/merging process, for instance, thesaurus or lexicons.
e) The level of user involvement: How and when the user is involved in the process.

Table 2 summarises the existing efforts to ontologies' coordination based on the above issues. The table has been produced based on the descriptions of the mentioned approaches in published articles. It must be mentioned that (a) some issues are not well defined (such as the use of the different types of knowledge and the exploitation of additional sources of information) and there can be objections on the characterization of methods based on them, and (b) the list is by no means exhaustive. However, this list provides a good starting point for discussing the major strengths of HCONE-merge in relation to other approaches.

Table 2. Issues concerning existing ontology mapping/merging tools

	Mapping Architecture	Kind of knowledge used	Type of result	Natural Language Information	User Involvement
ONIONS [4]	Mediated Bottom-up & Top-down	Lexical & Structural & Semantic	Mapping & Merging	Plain text descriptions	Semi-automatic
PROMPT [17]	Point-to-point Top-down	Lexical & Structural	Merging	No	Semi-automatic
FCA- Merge [21]	Point-to-point Bottom-up	Lexical & Structural	Merging	Natural Language Document	Semi-automatic
ONION [14]	Point-to-point Top-down	Lexical & Structural	Mapping & Merging	No	Semi-automatic
MOMIS [1]	Point-to-point Top-down	Lexical & Structural	Mapping	Thesaurus & WordNet	Semi-automatic
CUPID [11]	Point-to-point Top-down	Lexical & Structural	Mapping	Thesaurus	Automatic
IF-based [7]	Mediated Bottom-up	Lexical & Structural	Mapping	No	Automatic
GLUE [3]	Point-to-point Top-down	Lexical & Structural	Mapping	No	Semi-automatic
CTX- Match [20] S-Match [5]	Point-to-point Top-down	Lexical & Structural & Semantic	Mapping	WordNet	Automatic

A careful examination of the table shows that each research effort focuses on certain important issues. The HCONE-merge method, borrowing from the reported efforts, focuses on all of the issues mentioned above. In particular, we have realised that efforts conforming to mediated mapping and merging (such as [4] [7]) will possibly not work, since a reference ontology (that preserves the axioms of the source ontologies) may not be always available or may be hard to be constructed (especially in the “real world” of the Semantic Web). On the other hand, point-to-point efforts are missing the valuable knowledge (structural and domain) that a reference ontology can provide in respect to the semantic similarity relations between concepts. The HCONE merging process assumes that there is a hidden intermediate reference ontology that is built “on the fly” using WordNet senses that express the intended meaning of ontologies’ concepts and user-specified semantic relations among concepts.

Since bottom-up approaches [4] [7] [21] rely on strong assumptions concerning the population of ontologies, they have a higher grade of precision in their matching techniques since instances provide a better representation of concepts’ meaning in a domain. Using WordNet senses we provide an informal representation of concepts’ intensions (i.e. of the conditions for an entity to belong in the denotation of a concept, rather than the entities themselves).

More importantly, we have identified that apart from the efforts described in [3] [5] [20], most efforts do not consult domain knowledge significantly. As already pointed out, WordNet, as well as thesauruses and machine exploitable lexicons, are potential sources of such information. However, we must be careful in the way we exploit such sources of information. Utilizing for instance WordNet, in the way [5] and [17] do, implies that the domain ontologies must be consistent to the semantic relations between WordNet senses, which in our opinion is a very restrictive (if not prohibiting) condition for the alignment of the source ontologies. However, it must be noticed that when dealing with categories of documents in widely used search engines such as Google and Yahoo, the requirement for ontologies to be consistent with the inclusion relations of a generic lexicon such as WordNet may not be very restrictive, as it can be for ontologies in very specialized domains.

HCONE-merge exploits WordNet, which is an external (to the source ontologies) natural language information source. The proposed HCONE method consults WordNet for lexical information only, exploiting also structural information between senses in order to obtain the meaning of concepts (i.e. the informal human oriented semantics of defined terms). Other efforts such as [1] [11] [21] have used additional information sources but to our knowledge only efforts described in [5] and [20] have used WordNet for lexical and domain knowledge.

The basic aim of the research presented in this paper is to investigate the human involvement required during the process of ontology mapping and merging. Since we conjecture that in real environments such as the Semantic Web the humans’ intended meaning of concepts must always be captured, the question is where to place this involvement. Existing efforts [3] [4] [21], place this involvement after the mapping between source ontologies has been computed, as well as during, or at the end of the merging process. The user is usually asked to decide upon merging strategies, or to guide the process in case of inconsistency. Some other efforts head towards automated mapping techniques but they have not shown that a consistent and automatic merging will follow [5] [7] [11] [20].

The HCONE-merge approach places human involvement at the early stages of the merging process. If this involvement leads to capturing the intended meaning of conceptualisations, then the merging process follows, the results of which are subject to further human evaluation.

The method described up to this point (subsequently called the “user-validated” HCONE-merge method), requires users to be involved in every concept mapping in order to validate the LSI-hypothesized WordNet sense. For a small ontology, let us say a 10-concept ontology, this may not be considered a major hindrance. However, in real environments with hundreds of concepts, one can guess the frustration when validating the suggested mappings. This problem has led us to investigate the amount in which ontology mapping can be automated by exploiting the mapping algorithm presented in **Fig. 5**. In other words, the question to be answered concerns "*how much, if any, human involvement is required for ontology mapping and merging and in which stages of the merging process*". The rest of the paper presents methods that we have been experimenting with, and concludes with a suggested method for semi-automated mapping that has been tested and evaluated with real-life ontologies against manually created gold-standard ontologies [15].

6. Automating the HCONE-merge method

Given the crucial role of uncovering the intended meaning of concepts to the merging process, given two ontologies O_1 and O_2 to be merged, we aim at automating the mapping of O_1 and O_2 to the WordNet senses. As already pointed out, these mappings determine the alignment of the source ontologies. Then, the merging process can proceed as it has been explained in Section 3.

6.1. On automating the computation of the s-morphism

The following paragraphs present three methods towards automating the mapping process of the HCONE-merge method. All the experiments we have conducted involve ontologies of more than 10 and less than 100 concepts. For simplicity and presentation reasons, we will discuss here the results of experiments conducted with a 10-concept ontology taken from the Transportation domain:

O_1= ({*Airplane, Boat, Car, Craft, Motor Vehicle, Ship, Transport, Truck, Vehicle, Vessel*},
{*Vehicle*$\sqsubseteq$ *Transport, Motor Vehicle*$\sqsubseteq$ *Vehicle, Craft*$\sqsubseteq$ *Vehicle, Vessel*$\sqsubseteq$ *Vehicle, Car*$\sqsubseteq$ *Motor Vehicle, Truck*$\sqsubseteq$ *Motor Vehicle, Airplane*$\sqsubseteq$ *Craft, Boat*$\sqsubseteq$ *Vessel, Ship*$\sqsubseteq$ *Vessel*}).

This small ontology allows us to better inspect and criticize the results. Similar results however have been obtained in experiments with larger ontologies. Section 6 presents such experiments and their results.

6.1.1. Fully automated mapping

Fully automated mapping of ontology concepts to WordNet senses is achieved by running the mapping algorithm described in **Fig. 5** without any human intervention. That is, the algorithm simply maps each concept of the given ontology to the best-ranked sense returned by the algorithm. This method of computing the *s-morphism* for each ontology, gives the results shown in the "Automated 1st iteration" column of **Table 3**.

A mapping is considered to be "*correct*" if and only if the WordNet sense with which a concept is associated expresses the meaning intended by the ontology developer. To compute the mappings, the semantic space and the query string are constructed as specified in Section 3.

In order to increase the mapping precision by taking advantage of the correct mappings produced, we investigated the following method: Given the computed mappings of concepts to WordNet senses, the algorithm re-computes the *s-morphism*. Although the semantic space is computed in the same way as in the first iteration, the query string is constructed by taking into account the most frequent terms in the mappings produced during this iteration: If the mapping of a concept *C* has been changed during the 2nd iteration, then the new associated sense will be used for the computation of the query string for every concept that is related to *C* via an inclusion relation. Concerning our example, the mapping of the concept "*Vehicle*" has been changed in the second iteration of the mapping algorithm, since the query string for this concept has been changed due to the corrections made in the mappings of the concepts "*Craft*" and "*Car*". The latter concepts are related to the concept "*Vehicle*" via inclusion relations.

One cannot, of course, expect always a higher percentage of correct mappings after the second run. Due to the changes in the mappings, some correct mappings from the first run may change to wrong mappings and some others to correct ones, as the "Automated 2nd iteration" column of **Table 3** shows. So, even if the precision of the mapping seems to improve, the problem is the computation of wrong mappings for concepts whose mappings where computed correctly in the first run. This means that we cannot guarantee a higher precision after the second run.

Table 3. Results of the proposed methods for mapping ontologies to WordNet senses for the Transportation ontology.

Concept	Automated 1st iteration	Automated 2nd iteration	User-based	Semi-Automated
Airplane	✓	✓	✓	✓
Boat	✓	✓	✓	✓
Car	X	✓	✓	✓
Craft	X	✓	✓	✓
Motor Vehicle	✓	✓	✓	✓
Ship	✓	✓	✓	✓
Transport	✓	✓	✓	✓
Truck	X	X	X	X
Vehicle	✓	X	✓	✓
Vessel	✓	✓	✓	✓
✓ = Correct mapping X = Incorrect mapping				

Moreover, despite the second run, some concepts that were wrongly mapped in the first run remain wrongly mapped. The inability to correct the mappings of these concepts is due to the fact that the mappings of their related concepts have not been changed. Concerning our example, the inability to correct the mapping of the concept "*Truck*" is due to the fact that the query string remains the same for this concept, since it is computed by considering only the concept "*Motor Vehicle*", whose mapping has not changed.

6.1.2. User-based mapping

To overcome the problem of producing wrong mappings for those concepts whose mappings were correct in the first run of the algorithm, we can insist that the "correct" mappings of the first run are preserved. We can achieve this by requesting users' feedback on the results of the first run. The user is provided with a list of all the concepts of the ontology, and he/she can choose the concepts that are not mapped to their correct senses (**Fig. 9**). Doing so in the example ontology, one can choose to improve the mappings of the three concepts: "*Car*", "*Craft*", and "*Truck*". The mapping of the concept "*Truck*" remains unchanged (for the reasons described before), but the mappings of the other two concepts are corrected, as the "User-based" column of **Table 3** demonstrates.

Although this method produces more mappings that are "correct", it has two disadvantages: The first is that the user must check all the returned mappings one-by-one and validate them manually against the intended concept meaning. Thus, due to the overhead concerning the validation of the mappings, we are simply back where we started i.e. to the "*user-validated*" HCONE-merge method with a high user-involvement in the process of concept mapping. The second disadvantage is that, even if the user chooses a set of concepts whose mappings need to be corrected, the computation of the *s-morphism* is not guaranteed to improve the mappings for this set of concepts. This is due to the order in which concepts are considered: The algorithm does not produce any suggestions to which concept mappings must re-compute in the first place so as to further improve the mappings of the concepts in their vicinity. Therefore, in the worst case there may not be any improvement.

6.1.3. Semi-automated mapping

To minimize the time spent for the validation of mappings, to minimize user involvement and further guide the *s-morphism* computations to take advantage of the improvements made in the vicinity of concepts, we were motivated towards exploring methods for the automatic computation of the set of mappings that may need user validation. Towards this objective we implemented a method that locates *inconsistencies* between the translated axioms A_1 of ontology O_1 and WordNet inclusion relations.

An inconsistency occurs when concepts related via an inclusion relation are associated to WordNet senses via the *s-morphism* and these associations do not preserve the inclusion relations of the source ontology (i.e. these associations do not constitute a mapping to WordNet). It must be noted that although we do not insist that mappings of original ontologies must preserve inclusion relations between WordNet senses[4], the consistency-checking heuristic rule provides useful suggestions. A similar technique is used in the mapping algorithm proposed in [5].

Concerning our example case, the consistency-checking method suggested 4 concept pairs that produce such inconsistencies (**Fig. 9**). For example, the inconsistency for the pair of concepts "*Craft/Vehicle*" occurs, because (a) "*Craft*" is associated to the sense "*Craft, craftsmanship, workmanship -- (skill in an occupation or trade)*", (b) "*Vehicle*" is associated to the sense "*Vehicle -- (a conveyance that transports people or objects)*" and (c) these associations to the WordNet senses do not preserve their inclusion relation specified in the ontology.

For each suggested pair the user must identify which of the two concepts causes the inconsistency. In our example it is the "*Craft*" concept whose intended meaning does not match with the one computed by the s-morphsim. By running the *s-morphism* computation algorithm for this concept only, the inconsistency for the pair "*Craft/Vehicle*" is resolved by taking into account the correct mapping of the concept "*Vehicle*". However, it must be noted that making associations consistent does not ensure that mappings become "correct", since a consistent mapping does not necessarily reflect the intended meaning of the concept.

To improve the performance of the method, we have employed the following heuristic: In case an association is still wrong (i.e. the corresponding WordNet sense is not the intended one) or produces an inconsistency, then the mappings of the concepts that are semantically

[4] In the previous methods described, WordNet inclusion relations were not taken into account, since the axioms set of the hidden intermediate ontology to which concepts are mapped includes the translated axioms of the source ontology.

related to the suggested concepts are further validated. For instance, concerning the pair "*Craft/Vehicle*", in case the inconsistency can not be resolved, then the mappings of their related concepts must be validated. For instance, in case the mapping of "*Transport*" is wrong, the user runs the *s-morphism* calculation algorithm again only for the concept "*Transport*". Having a new mapping for the concept "*Transport*", the user re-runs the mapping algorithm for "*Craft*", resulting in a correct mapping. This heuristic provides further guidance for locating the concept(s) whose mappings distract (as this has been explained in section 4.1) the *s-morphism* from computing the correct senses of concepts in their vicinity.

6.1.4. Comparison of the methods

Based on the basic algorithm for computing the *s-morphism*, we have shaped an approach to ontology mapping, where human inspection and validation has been reduced down to the number of algorithm runs needed to correct the concept pairs whose associations produce inconsistencies with respect to the WordNet inclusion relations.

Table 4. Comparison of the proposed methods

	Fully-Automated (2^{nd} iteration)	User-based	Semi-Automated
Percentage of concepts validated by the user	0%	≤100%	≥0%

Table 4 summarizes the proposed methods according to the amount of the automation they achieve. The "*fully automated*" method requires the minimum number of user actions, but at the same time, as our experiments have shown, it achieves the lowest percentage of correct mappings. On the other hand, the "*user-based*" method achieves higher percentage of correct mappings, but the actions that are required by the user imply considerable effort, since the user has to validate the mapping of each ontology concept. It must be noted that this case requires also a considerable number of additional algorithm runs, equal to the percentage of wrong mappings. The "*semi-automated*" method, however, can significantly reduce the number of concepts that need validation by the user. In the worst case, where each concept is involved in at least one inconsistency, validation of all concepts is required.

6.2. Mapping of Ontology O_2 to Ontology O_1

Having reached the point where O_1 and O_2 have been mapped to the hidden intermediate ontology, we add one more step prior to merging: The mapping of unmatched concepts of O_2 to unmatched concepts of O_1. The motivation is to increase the mapping efficiency (i.e. the number of concept matches). This additional step is sketched as follows: *Find the set of non-matched concept pairs of O_1 and O_2, and re-compute the mappings by using only the matched pairs that have derived from the initial mapping of O_2 and O_1 to WordNet.*

"*Non-matched*" concept pairs include those whose mappings either:

(a) Have a different lexical entry (C_1' and C_2') in WordNet (e.g. C_1:*Facility*/C_2:*Installation*) and either $f_s(C_1)$ or $f_s(C_2)$ belong to C_1's or C_2's WordNet sysnset (e.g. $f_s(C_1)$: "*facility, installation -- something created to provide a particular service; "the assembly plant is an enormous facility*" is identical with sense 3 of C_2 synset), or

(b) Have the same WordNet lexical entry (e.g. $C_{1:}$*Facility/*$C_{2:}$*Facility*) but their associated senses in the related synset are not identical.

The mapping of an unmatched concept C_2 in O_2 can be conducted in a semantic space that is constructed by (a) those senses of C_2 that have been computed by mapping O_2 to WordNet and (b) the terms in the vicinity of the already-computed $f_s(C_2)$ sense. The query string is constructed by the most frequent terms of the highest graded WordNet senses of every concept C_R that is related to C_2 via an inclusion relation. C_R has to match with a concept of O_1. The mapping method is outlined as follows:

For each concept C_1 of O_1 and each concept C_2 of O_2 for which either (a) or (b) happens:

(a) Both $f_s(C_1)$ and $f_s(C_2)$ correspond to the same WordNet lexical entry and belong to the same synset

(b) C_1 and C_2 correspond to a different WordNet lexical entry and either $f_s(C_1)$ or $f_s(C_2)$ belong to C_1's or C_2's corresponding WordNet synset,

run the mapping algorithm for the concept C_2:

- The query string is constructed by taking into account the most frequent terms of $f_s(C_R)$, for every concept C_R that matches to a concept of O_1 and is related to C_2 via an inclusion relation.
- The semantic space is constructed by taking into account the senses of C_2 that have been computed by mapping O_2 to WordNet.

This additional heuristic can produce new mappings i.e. mappings between two concepts C_1 and C_2 that have an identical meaning. Although experiments we conducted have shown that additional mappings have being identified through this technique (e.g. between concepts "O_2-*Facility*" and "O_1-*Installation*", as it shown Fig. 7), further investigation is needed in order to specify the amount of additional information that is necessary to improve the initial mappings.

Fig. 9.Computing a mapping for the concept "Facility"

7. Evaluation of HCONE-merge methods with real-life ontologies

To further support the work presented in this paper, we have run experiments with ontologies from the DAML ontology library[5] and from the library of the ACCORD project[6] [5]. The descriptions of the example source ontologies taken from the DAML library are summarized in **Table 5**. In addition, an outline of the ontologies and their mapping can be found in **Fig. 11**.

[5] DAML. DAML ontology library 2004, www.daml.org

[6] The ACCORD project – Experiments**,** http://dit.unitn.it/~accord/

Table 5. Details of the experiment's source ontologies

Ontology O_1 (29 concepts)	Ontology O_2 (43 concepts and 5 relations)
Academic Positions	Academic Departments
By Terry R. Payne of Carnegie Mellon University http://www.daml.ri.cmu.edu/ont/homework/cmu-ri-employmenttypes-ont.daml	By Jeff Heflin of Univ. of Maryland http://www.cs.umd.edu/projects/plus/DAML/onts/cs1.0.daml
Ontology describing employment hierarchy based on many of the positions available at the Robotics Institute, CMU	Ontology for computer science academic departments. This is the DAML version of a SHOE ontology

According to N. Noy and M. Musen [15] [16], a "good" merging tool is a tool that produces "good" results when measuring the "distance" between the ontology produced and a manually created gold-standard ontology. We call "gold-standard" the ontology that results from experts' merging of the two source ontologies. In our experiments with DAML ontologies, gold-standard ontologies have been produced by experts in the corresponding domains using any kind of available knowledge (lexical, structural, domain): Structural knowledge concerns the equivalence and inclusion relations between concepts. Such knowledge constrains the meaning of concepts. Domain knowledge concerns the meaning of domain terms and their interrelations. For the ACCORD ontologies, we have used the expert mappings provided with the ontologies [5]. In both cases the mappings of ontology concepts to their informal intended meanings have been specified by domain experts so as to measure the recall and precision of the mappings to WordNet. Although domain experts may not agree on the gold-standard ontologies, these provide the standard for measuring the performance of the methods proposed.

The *distance* from the gold-standard ontology is being measured by means of the number of concept pairs that a method fails to identify. Furthermore, the *recall* of mappings is defined as the fraction of correct mappings to WordNet senses that the algorithm identifies. The *precision* is defined as the fraction of correct mappings to WordNet senses among the mappings that the method computes. The distance between the ontologies, as well as the recall and precision of mappings have been measured by inspecting the source ontologies, the suggested merging actions of the experts and the intended meanings of ontology concepts.

7.1. Measuring precision and recall of mapping to WordNet

Mapping ontology concepts to WordNet senses is critical to the success of our merging approaches. In this section we present and compare the results of the mappings to WordNet senses produced with the HCONE-merge "*user-validated*", "*user-based*" and "*semi-automated*" methods using the domain ontologies described in **Table 5**.

Table 6. Recall and Precision measures of mapping concepts to WordNet senses

	"User validated" method	**"User-based" method**	**"Semi-Automated" method**
Recall (in the fist iteration of the *s-morphism* computation algorithm)	**69%**		
Recall	**69%**	**73%**	**80%**
Precision	**97%**	**79%**	**89%**

As shown in **Table 6**, the percentage of correct mappings that are identified in the first iteration of the *s-morphism* is 69%. This is further increased by the involvement of the user and the heuristics in the "user-based" and the "semi-automated' method.

This percentage is also due to the fact that WordNet misses lexicalisations of some ontology concepts. Although WordNet is a general lexicon, some technical or very domain specific terms are not present. Future versions of WordNet (1.7.1 and 2.0) will be integrated in our approach in order to increase recall. However, as we have observed in experiments with ontologies including technical concepts, given that concepts have enough information in their vicinity for the computation of the *s-morphism*, the presence of technical terms do not influence their mapping. This is true, if the number of the technical terms in the association matrix of the LSI is limited, and thus, these terms are considered irrelevant.

The precision of the "*user-validated*" method, which is the basic HCONE-merge method that we have described in Section 4, is 97% (and not 100%) because the user may not be able to choose the intended meaning of a concept from the list of WordNet senses that the algorithm focuses on.

The "*User-based*" method provides the opportunity for the user to validate all the mappings one-by-one. This method achieves a 79% precision in the experiments conducted. The percentage is lower than the percentage of the "*user-validated*" process, since in this case the user does not indicate the intended meaning. The user points to the concepts whose mapping need to be re-examined, but the algorithm may not find the intended meaning of these concepts when it is rerun.

Finally, the "*Semi-automated*" mapping incorporates no additional techniques concerning the computation of the *s-morphism*. However, the use of the heuristic that suggests pairs with inconsistent associations to WordNet senses, together with the fact that the algorithm (during the construction of the query) considers the correct mappings of the concepts in the vicinity of the queried concept, raises the precision of this method to 89%.

Although the "User-based" and "Semi-automated" methods are both based on re-computing the *s-morphism* for a specific set of concepts, their precision may differ because of the second method's heuristic techniques. These techniques exploit semantic information (WordNet semantic relations between synsets) for checking the validity of the suggested mappings and, as already pointed, they further guide re-computations of the *s-morphism* ensuring that inconsistent to WordNet mappings will "have their chance" to be corrected.

7.2. Measuring distance to the gold-standard ontology: Comparison to other tools and methods

As already pointed out by other authors in [15], [16] and [22], we also support that there is not a "best" tool, since a tool cannot satisfy all user and application requirements. In our point of view, a "better" tool is a tool that will provide "better" recall and precision results in uncovering the intended meaning of concepts, and that will produce ontologies that are very close to the gold-standard ontologies suggested by domain experts. Therefore, it is rather difficult to compare recall and precision percentages of HCONE-merge methods with the results of other approaches such as GLUE (66-97% precision [3], i.e. 3-34% distance from the gold-standard ontology), C_{TX}-MATCH (60-76% precision [20], i.e. 24-40% distance from the gold-standard ontology), or S-Match (90% precision [5], i.e. 10% distance from the gold-standard ontology). The trade-off between precision percentages, time and human involvement spent during mapping must be carefully examined when doing such comparisons, as well as the input requirements (kind of knowledge used) of each approach i.e. the use of instances, or the use of additional information sources such as lexicons or corpora. To our knowledge, the only mapping approach which is close to HCONE-merge is S-Match [5]. S-Match computes concept matches semi-automatically with a high overall estimation, using semantic similarity measures. However, this method seems to work only for those ontologies that are categories hierarchies and preserve the inclusion relations among WordNet senses.

To evaluate our methods we have conducted experiments with the HCONE-merge "semi-automated" method on several ontologies found in the ACCORD project web site, and compared the results against the expert mappings provided by ACCORD.

For the "Simple Catalogs" ontologies that have been matched and evaluated by S-Match, we found a complete similarity with the expert mappings. For the "Parts of Google" and "Yahoo web directories" ontologies that have been matched and evaluated by CTX-Match, we found a complete similarity with the expert mappings also. The CTX-Match precision and recall measures for this case are not given in [5]. For the "Company profiles (mini)" ontologies that have been matched and evaluated by S-match, we found a distance of 4 from their expert mappings (four mappings were missed) [5]. However, we have to notice that although expert mappings provide a standard for measuring the performance of merging methods, they can be further refined. For instance, experts have mapped the concept "*Oil_Well_Services_Equipment*" with both concepts "*Oil_Gas_Equipment*" and "*Oil_and_Gas_Services*". A mapping between these concepts could not be achieved if the intended meaning of "*Oil_Well_Services_Equipment*" is "the equipment which is used to get an oil well serviced". This intended meaning can support a mapping with "*Oil_Gas_Equipment*", but it is not clear how this can be done with the "*Oil_and_Gas_Services*". Apart from that, due to the treatment of multi word terms proposed in Section 3, the translation mechanism and the description logics reasoner used in HCONE-merge, can provide a classification of "*Oil_Well_Services_Equipment*" with respect to these concepts as shown in **Fig. 10**.

Fig. 10. Description Logics reasoner classification of "*Oil_Well_Services_Equipment"* concept

To further evaluate HCONE-merge we have conducted experiments with the ontologies *Academic Positions* and *Academic Departments* shown in **Fig. 11.** These ontologies have also been merged using PROMPT, a mapping/merging method of Protégé-2.0 tool suite (PromptTab plug-in). **Table 7** shows pairs of concepts that PROMPT and HCONE-merge suggested for merging for these ontologies. As shown in **Table 7**, an expert suggested 9 pairs with matching concepts. These matching pairs have been used for the production of the gold-standard ontology. In order to find the matching pairs 1 to 6, lexical and semantic matching has been performed. Pairs 7 to 9 have been semantically examined since there is no lexical similarity. Merging of pairs 7 to 9 has been decided based on the agreed intended meaning of the concepts of both ontologies.

Closer to the gold-standard ontology are the suggestions that the HCONE-merge method produces using the "*user-validated*" mapping method. Baring in mind that this is a step-by-step mapping method, the users have produced the same mappings to WordNet senses for concept pairs 1 to 6 as well as for concept pairs 7 and 8. This has resulted in 8 concept pairs suggested for merging. Therefore, a distance of 1 to the gold-standard ontology occurs since pair 9 of the gold-standard has not been suggested by this method. For pairs 7 and 8 the *s-morphism* resulted in a complete match of WordNet senses.

Table 7. Concept pairs suggested for merging

1. Gold-standard merging pairs	**2. "User-validated" HCONE-merge method**
1. FACULTY, FACULTY 2. PROFESSOR, PROFESSOR 3. ADMINISTRATIVE STAFF, ADMINISTRATIVE STAFF 4. DIRECTOR, DIRECTOR 5. STUDENT, STUDENT 6. ASSISTANT, ASSISTANT 7. MANAGER, DIRECTOR 8. POSTDOCTORAL FELLOW, POST DOCTOR 9. STAFF, WORKER	1. FACULTY, FACULTY 2. PROFESSOR, PROFESSOR 3. ADMINISTRATIVE STAFF, ADMINISTRATIVE STAFF 4. DIRECTOR, DIRECTOR 5. STUDENT, STUDENT 6. ASSISTANT, ASSISTANT **7. MANAGER, DIRECTOR** **8. POSTDOCTORAL FELLOW, POST DOCTOR**
3. "Semi-automatic" HCONE-merge method	**4. PROMPT**
1. FACULTY, FACULTY 2. PROFESSOR, PROFESSOR 3. ADMINISTRATIVE STAFF, ADMINISTRATIVE STAFF 4. DIRECTOR, DIRECTOR 5. STUDENT, STUDENT 6. ASSISTANT, ASSISTANT **7. MANAGER, DIRECTOR**	1. FACULTY, FACULTY 2. PROFESSOR, PROFESSOR 3. ADMINISTRATIVE STAFF, ADMINISTRATIVE STAFF 4. DIRECTOR, DIRECTOR 5. STUDENT, STUDENT 6. ASSISTANT, ASSISTANT
Pairs in Bold =	Pairs with different concept names but with same meaning which have been discovered using semantic matching.

Using the "*Semi-automatic*" mapping method, the distance to the gold-standard ontology is getting even greater. Since now the user is not provided with a mechanism to assign WordNet senses manually, the algorithm automatically finds the mappings. The suggested pairs in this case are 7, since:

a) *O_1-Manager* and *O_2-Director* match due to the fact that their associated WordNet senses are identical,

b) *O_1-PostDoctoral Fellow* and *O_2-Post Doctor* do not match, since their associated senses are different and the components of their translated definitions cannot be mapped. Specifically, given that:

$POSTDOCTORAL\_FELLOW \sqsubseteq FELLOW \sqcap position.POSTDOCTORAL$,

and $POST\_DOCTOR \sqsubseteq DOCTOR \sqcap position.POST$,

since $f_s(FELLOW)$ is different from $f_s(DOCTOR)$, then the concept *POSTDOCTORAL_FELLOW* does not match with the concept *POST_DOCTOR*

c) Concepts *O_1-Staff* and *O_2-Worker* do not match, since their associated senses are different and in different WordNet synsets.

Concerning PROMPT, the distance to the gold-standard ontology is getting even greater, since PROMPT fails to discover semantic similarity for pairs 7 to 9.

To further validate the precision results of concept mappings to WordNet senses as well as the low distance from the "gold-standard" ontology, we also experimented with "bibliographic ontologies" taken from the EON-2004 Ontology Alignment Contest[7]. The results were very encouraging, reinforcing the efficiency of our approach towards the (semi)automated merging of ontologies.

Although the experiments conducted so far have been rather encouraging, our approach deserves further exploitation and study with more test cases. Larger and more technical ontologies should be tested in the near future.

8. Mapping using domain relations

In real-life ontologies, it is usually the case that domain relations other than *inclusion* and *equivalence* will be present. In the HCONE-merge methods, relations can be used to increase the precision of the *s-morphism* computation algorithm, thus the precision of uncovering the informal intended meaning of concepts. For example, the source ontology *Academic Departments* (**Fig. 11**) uses a number of domain relations that describe in more detail some of the concepts. The experiments described in Section 5 were conducted without including these relations. When these relations were involved in the mapping process - as terms in the semantic space and the query- an increase of the precision was observed. For instance, when the relations (*teacher* and *masters degree*) were included in the mapping process, the concept "*Dean"* defined as $Dean \sqsubseteq Professor \sqcap AdministrativeStaff \sqcap teacherOf.Course \sqcap mastersDegreeFrom.University$ (which was initially mapped to the sense: "*DEAN = dean, doyen -- (the senior member of a group;)*" was now mapped to its intended meaning (i.e. to the sense: "*DEAN = dean -- (an administrator in charge of a division of a university or college)*". Although in some cases, such as in the concept "*Post Doctor*", the use of relations and related concepts changed the mapping to a consistent but not correct sense, the plethora of cases (more than 70%) result in a correction.

The above technique may involve any relation between concepts. The incorporation of the terms that lexicalize a relation, as well as the incorporation of the terms that lexicalize a related concept, adds valuable domain knowledge to the query since relations and related concepts apply certain domain specific distinctions to the senses of concepts. For instance, the relation "*teacher*" and the related concept "*course*" together with the relation "*mastersDegreeFrom*" and the related concept "*University*" clearly distinguish the senses of the concept "*Dean*" mentioned above. This influences LSI to compute the sense: "*dean -- (an administrator in charge of a division of a university or college)*" as the most relevant sense.

It must be noticed that this technique is used only to improve the mapping of concepts to WordNet senses, and does not address the general issue of mapping concepts using domain relations to capture domain knowledge as it has been proposed in other approaches (MOMIS [1], CUPID [3], CTX-Match [20], S-Match [5]). For HCONE-merge, domain knowledge is captured by uncovering the intended meanings of concepts through the mapping of ontology concepts to WordNet senses. In MOMIS and CUPID, domain knowledge is captured by the use of semantic relations (synonymy, hypernymy, and relationship) found in thesauruses. These relations, in conjunction to concept names and structure, are exploited for the computation of affinity coefficients between two concepts. In S-Match and CTX-Match, relations between WordNet senses are used to check the validity of domain relations between

[7] http://co4.inrialpes.fr/align/Contest/

concepts of two source ontologies. However, the above mentioned approaches are very restrictive due to the fact that domain relations must be available in an external source (WordNet, thesaurus) in order to be used for matching.

Fig. 11. Source ontologies and their mapping

9. Concluding Remarks

In this paper we presented a number of methods dealing with the mapping of concepts to their intended meaning. Our aim was to identify the minimum user involvement required during the merging of ontologies. We presented an experimental evaluation of the proposed methods on various ontologies. Furthermore, we compared our methods to other approaches and tools with very promising results.

With respect to automating the mapping and merging processes we conjecture that in real environments such as the Semantic Web, the humans' intended meaning of concepts must always be captured. The aim of our research is to reduce human involvement for capturing the intended meaning of concepts. The HCONE-merge methods place human involvement at the early stages of the merging process. If this involvement leads to capturing the intended meaning of conceptualisations, then the rest is a consistent, error-free merging process, whose results are subject to further human evaluation. The new methods proposed in this paper show that human involvement is necessary to produce valid mappings between ontologies, however this involvement can be reduced significantly.

Major points that differentiate HCONE-merge from other approaches are the following:

- It supports the mapping/merging of ontologies in absence of a reference ontology, which seems to be hard to find, especially in a dynamic environment such as the Semantic Web.
- It supports the mapping/merging of ontologies that are not populated by instances, a very usual case again in the Semantic Web.
- It incorporates semantic knowledge into the mapping/merging of ontologies, using additional natural language sources, without requiring the existence of specific domain relations within these sources.
- Human involvement is required at the early stages of the process, where humans must validate the intended informal meanings assigned to ontology concepts. This makes the mapping/merging process to be seamlessly integrated in the ontology development lifecycle, avoiding difficult decisions that require ontology engineering skills.
- It limits human involvement during the mapping/merging method down to a small number of validations of mappings that HCONE-merge techniques suggest.

On the other hand, the current implementation of the HCONE-merge can not be considered for use with high technical domain ontologies: Highly technical terms do not have an entry in WordNet resulting in poor performance of the method.

Future work concerns additional experiments with real life ontologies. More importantly, the incorporation of other natural language sources instead of, or in conjunction to, WordNet is being investigated. Furthermore, additional heuristics are currently added in the experiments in order to investigate alternative methods of minimizing user involvement in the coordination process.

Acknowledgements

We would like to thank the anonymous reviewers of this paper for their very constructive comments and suggestions, which helped to improve earlier versions of this paper greatly.

References

[1] D. Beneventano, Bergamaschi, S., Guerra, F., Vincini, M. The MOMIS approach to Information Integration. IEEE and AAAI International Conference on Enterprise Information Systems, Setŭbal, Portugal (2001)
[2] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman. Indexing by Latent Semantic Analysis. Journal of the American Society of Information Science (1990)
[3] A. Doan, J.Madhavan, P. Domingos, and A. Halvey. Learning to map between ontologies on the semantic web. In Proceedings of WWW-02, 11th International WWW Conf., Hawaii (2002)
[4] A. Gangemi, D. M. Pisanelli, G. Steve. An Overview of the ONIONS Project: Applying Ontologies to the Integration of Medical Terminologies. Data and Knowledge Engineering, 1999, Vol. 31 (1999) 183-220
[5] F. Giunchiglia, P. Shvaiko, M. Yatskevich. S–Match: An Algorithm and Implementation of Semantic Matching. The Semantic Web: Research and Applications, Lecture Notes in Computer Science, Vol. 3053, Springer-Verlag, (2004) 61-75
[6] N. Guarino and C. Welty. Ontological analysis of taxonomic relationships. In A. Laender and V Storey, editors, ER-2000: The 19th International Conference on Conceptual Modeling. Springer-Verlag (2000)
[7] Y. Kalfoglou and M. Schorlemmer. Ontology mapping: the state of the art. The Knowledge Engineering Review 18(1):1--31 (2003)
[8] K. Kotis and G. A. Vouros. HCONE-Merge approach to ontology merging. The Semantic Web: Research and Applications, Lecture Notes in Computer Science, Vol. 3053, Springer-Verlag (2004) 137-151
[9] K. Kotis, G. A. Vouros, K. Stergiou. Capturing Semantics towards Automatic Coordination of Domain Ontologies. To appear at the 11th International conference of Artificial Intelligence: Methodology, Systems, Architectures - Semantic Web Challenges - AIMSA 2004, Varna, (2004)
[10] J. Li. LOM: A Lexicon-based Ontology Mapping Tool. Techknowledge Co., http://reliant.teknowledge.com/DAML/I3con.pdf
[11] J. Madhavan, P. A. Bern-stein, and E. Rahm. Generic schema matching with Cupid. VLDB Journal (2001) 49-58
[12] J. Madhavan, P. A. Bern-stein, P. Domingos, and A. Y. Halevy. Representing and reasoning about mappings between domain models. Proc. of the 18th AAAI Canada (2002) 80–86
[13] G. Miller, R. Beckwith, D. Fellbaum, et al. Five papers on WordNet. CSL Report 43, Cognitive Science Laboratory, Princeton University, 1995.
[14] P. Mitra, Wiederhold, G., Decker, S.: A Scalable Framework for Interoperation of Information Sources. In Proceedings of the 1st International Semantic Web Working Symposium, Stanford University, Stanford, CA (2001)
[15] N. Noy and M. Musen. Evaluating Ontology-Mapping Tools: Requirements and Experience. EON2002 Evaluation of Ontology-based Tools, OntoWeb-SIG3 Workshop

at the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW2002 Siguenza, Spain (2002) 1-14

[16] N. Noy. What do we need for ontology integration on the Semantic Web. Proceedings of the Semantic Integration Workshop Collocated with the Second International Semantic Web Conference (ISWC-03) Sanibel Island, Florida, (2003)

[17] N. Noy and M. Musen, A. M. PROMPT: Algorithm and tool for automated ontology merging and alignment. In Proceedings of 7th National Conference on AI, Austin, TX (2000)

[18] T. O'Hara, N. Salay, M. Witbrock, et al. Inducing criteria for mass noun lexical mappings using the Cyc Knowledge Base and its Extension to WordNet. In Proceedings of the Fifth International Workshop on Computational Semantics, Tilburg, 2003.

[19] C. J. Sager. A practical course in terminology processing. John Benjamins, Amsterdam/Philadelphia (1990)

[20] L. Serafini, P. Bouquet, B. Magnini, and S. Zanobini. An Algorithm for Matching Contextualized Schemas via SAT. In Proceedings of CONTEX 03, (2003)

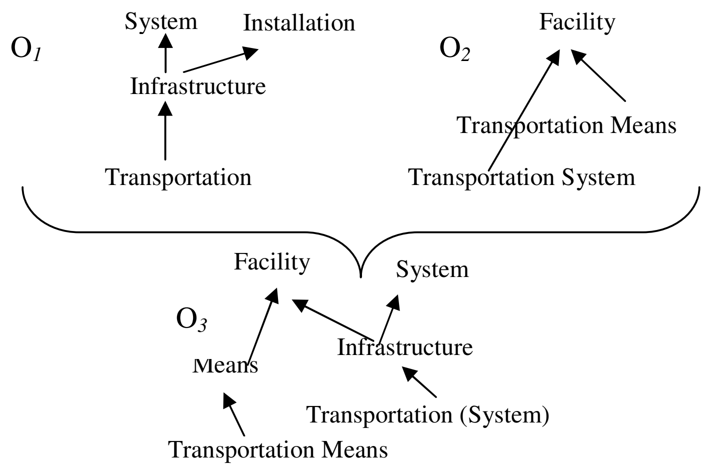
[21] G. Stumme, A. Mädche. FCA-Merge: Bottom-Up Merging of Ontologies. In: B. Nebel (Ed.): Proc. 17th Intl. Conf. on Artificial Intelligence (IJCAI '01). Seattle, WA, USA (2001) 225-230

[22] M. Uschold and M. Gruninger. Creating Semantically Integrated Communities on the World Wide Web. Invited Talk, Semantic Web Workshop, WWW 2002 Conference, Honolulu, H1. (2002)

[23] M. Uschold. Where are the Semantics in the Semantic Web? AI Magazine, Volume 24, No.3 (2003)

[24] A. Valarakos, A., Paliouras, G., Karkaletsis V., Vouros, G. A name-Matching Algorithm for supporting Ontology Enrichment. In Proceedings of SETN’04, 3rd Hellenic Conference on Artificial Intelligence, Samos, Greece (2004)

Figures



O_1= ({System, Infrastructure, Installation, Transportation},
{Transportation $\sqsubseteq$ Infrastructure, Infrastructure $\sqsubseteq$ Installation, Infrastructure $\sqsubseteq$ System})

O_2= ({Facility, Transportation System, Transportation Means, exploit},
{Transportation System $\sqsubseteq$ Facility,
Transportation Means $\sqsubseteq$ Facility $\sqcap$ exploit.TransportationSystem })

O_3= ({System, facility, Means, Installation, Infrastructure, Transportation System, Transportation, Transportation Means, exploit},
{Transportation $\equiv$ Transportation System,
Facility $\equiv$ Installation, Infrastructure $\sqsubseteq$ System $\sqcap$ Facility,
Transportation System $\sqsubseteq$ Infrastructure $\sqcap$ Facility,
Transportation Means $\sqsubseteq$ Means $\sqcap$ exploit.Transportation System , Means $\sqsubseteq$ Facility})

Fig. 1.Example Ontologies

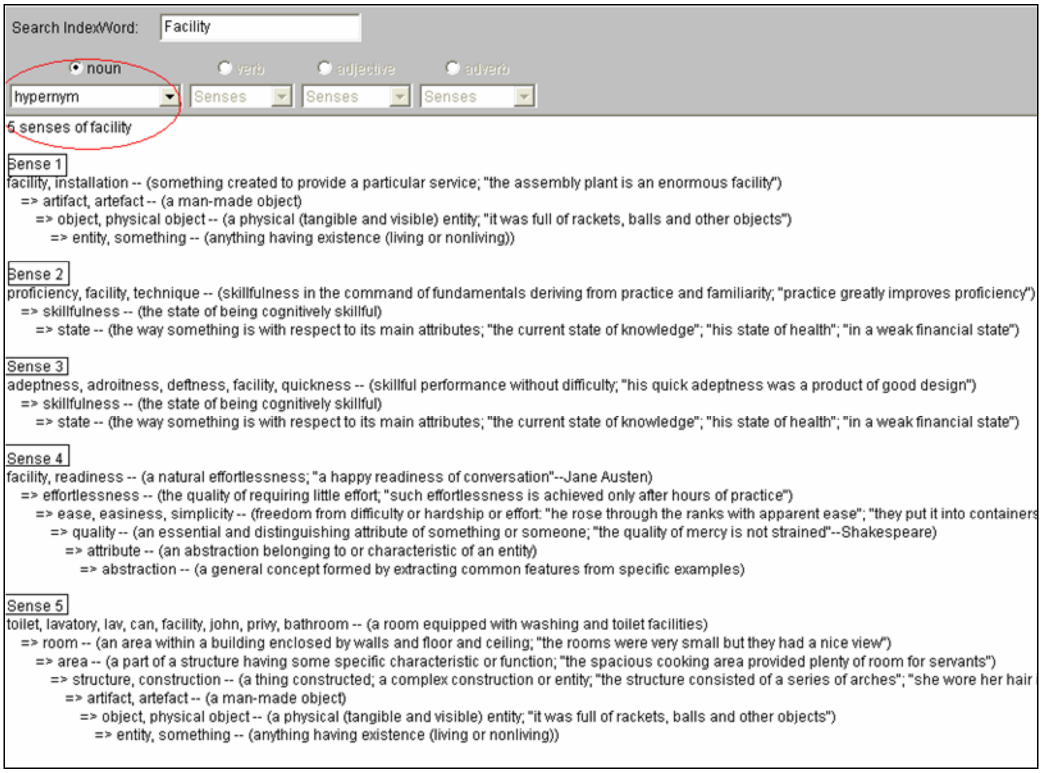


Fig. 2. WordNet information for concept *Facility*

$$\mathbf{A1} = \begin{pmatrix} 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0.5774 & 0 & 1.0000 & 0.4082 & 0.7071 \\ 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0 & 0 & 0 & 0.4082 & 0 \\ 0 & 1.0000 & 0 & 0.4082 & 0.7071 \\ 0 & 0 & 0 & 0.4082 & 0 \end{pmatrix}$$

$$\mathbf{A2} = \begin{pmatrix} 0.4971 & -0.0330 & 0.0232 & 0.4867 & -0.0069 \\ 0.6003 & 0.0094 & 0.9933 & 0.3858 & 0.7091 \\ 0.4971 & -0.0330 & 0.0232 & 0.4867 & -0.0069 \\ 0.1801 & 0.0740 & -0.0522 & 0.2320 & 0.0155 \\ -0.0326 & 0.9866 & 0.0094 & 0.4402 & 0.7043 \\ 0.1801 & 0.0740 & -0.0522 & 0.2320 & 0.0155 \end{pmatrix}$$

Fig. 3. A matrix in LSI method: 2 phases of computation

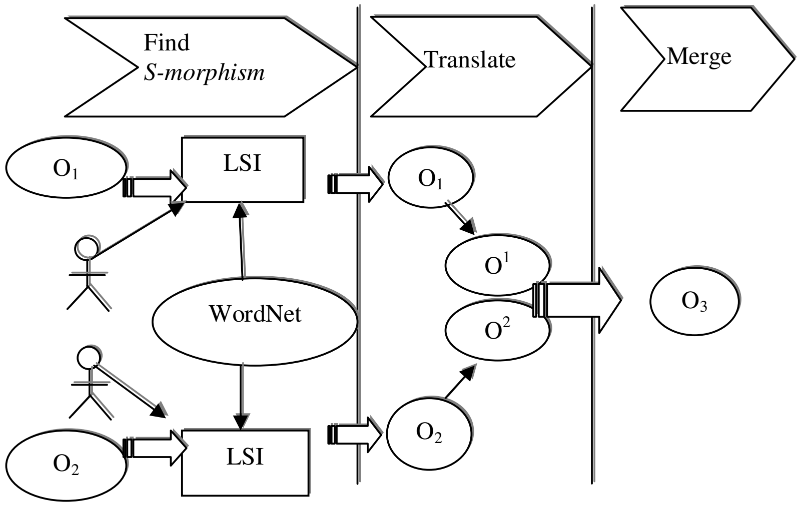


Fig. 4. The HCONE approach towards the OMP

1. Choose a concept from the ontology. Let *C* be the concept name.
2. Get all WordNet senses $S_1, S_2, \ldots S_m$, lexicalized by *C'*, where *C'* is a linguistic variation of *C*. These senses provide the *focus of the algorithm for* C.
3. Get the hyperonyms and hyponyms of all *C'* senses.
4. Build the "*association matrix*": An $n \times m$ matrix that comprises the *n* more frequently occurred terms in the *vicinity* of the *m* WordNet senses found in step 2.
5. Build a query string using the terms in the *vicinity* of *C*. The query string is a sequence of digits, each digit taking value 0 if a term in the *vicinity* of *C* does not exist in the set of *n*, and 1 if a query term exists in the set of *n*.
6. Find the ranked associations between *C* and *C'* senses by running the Latent Semantics Analysis (LSA) function and consider the association with the highest grade. LSA uses the query terms for constructing the query string and computes the similarities between the query and the senses in the focus of the algorithm.

Fig. 5. The algorithm for computing the *s-morphism*

<table>
<tr><th>S-morphism algorithm description step by step</th><th>Algorithm's Example Output</th></tr>
<tr><td>Step 1,2,3: The focus of the algorithm for concept "Facility"</td><td>
S_1: facility, installation -- (something created to provide a particular service; "the assembly plant is an enormous facility") => transportation system, transportation, transit…

S_2: proficiency, facility, technique -- (skillfulness in the command of fundamentals deriving from practice and familiarity; "practice greatly improves proficiency") => technique, skilfulness, state…

S_3: adeptness, adroitness, deftness, facility, quickness -- (skillful performance without difficulty; "his quick adeptness was a product of good design") => quickness, skilfulness, state…

S_4: facility, readiness -- (a natural effortlessness; "a happy readiness of conversation"--Jane Austen) => easiness, simplicity, quality…

S_5: toilet, lavatory, lav, can, facility, john, privy, bathroom -- (a room equipped with washing and toilet facilities) => structure, construction, artefact…
</td></tr>
<tr><td>Step 4: Part of the semantic space ($n \times m$ matrix) for concept "Facility"</td><td>
<table>
<tr><th>5 Senses (m)
93 Terms (n)</th><th>S_1</th><th>S_2</th><th>S_3</th><th>S_4</th><th>S_5</th></tr>
<tr><td>T_1: weight 3, POLICE</td><td>3</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>T_2: weight 2, LOUVRE</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>T_3: weight 2, LANDING</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>T_4: weight 5, INSTALLATION</td><td>5</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>T_5: weight 3, DEFTNESS</td><td>0</td><td>0</td><td>3</td><td>0</td><td>0</td></tr>
<tr><td>T_6: weight 2, NETWORK</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>T_7: weight 2, SWIMMING</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>T_8: weight 2, RAPID</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>T_9: weight 3, PUBLIC</td><td>1</td><td>0</td><td>0</td><td>0</td><td>2</td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
<tr><td>T_{35}: weight 15, TRANSPORTATION SYSTEM</td><td>3</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>Step 5: The query terms and string for concept "Facility"</td><td>
QueryTerm 0 :FACILITY

QueryTerm 2 :TRANSPORTATION-MEANS

QueryTerm 3 :TRANSPORTATION-SYSTEM

$\underline{T_1}$ $\underline{T_2}$ $\underline{T_3}$ $\underline{T_4}$ $\underline{T_5}$ $\underline{T_6}$ $\underline{T_7}$ $\underline{T_8}$ $\underline{T_9}$ …$\underline{T_{35}}$ …
Query String: 0 0 0 0 0 0 0 0 0 … 1…
</td></tr>
<tr><td>Step 6: The highest grade association is Sense 1 (S1)</td><td>
<u>S1</u> <u>S2</u> <u>S3</u> <u>S4</u> <u>S5</u>
0,032 0,013 0,02 -0,06 -0,052
</td></tr>
</table>

Fig. 6. A running case for computing the mapping of the concept "*Facility*"

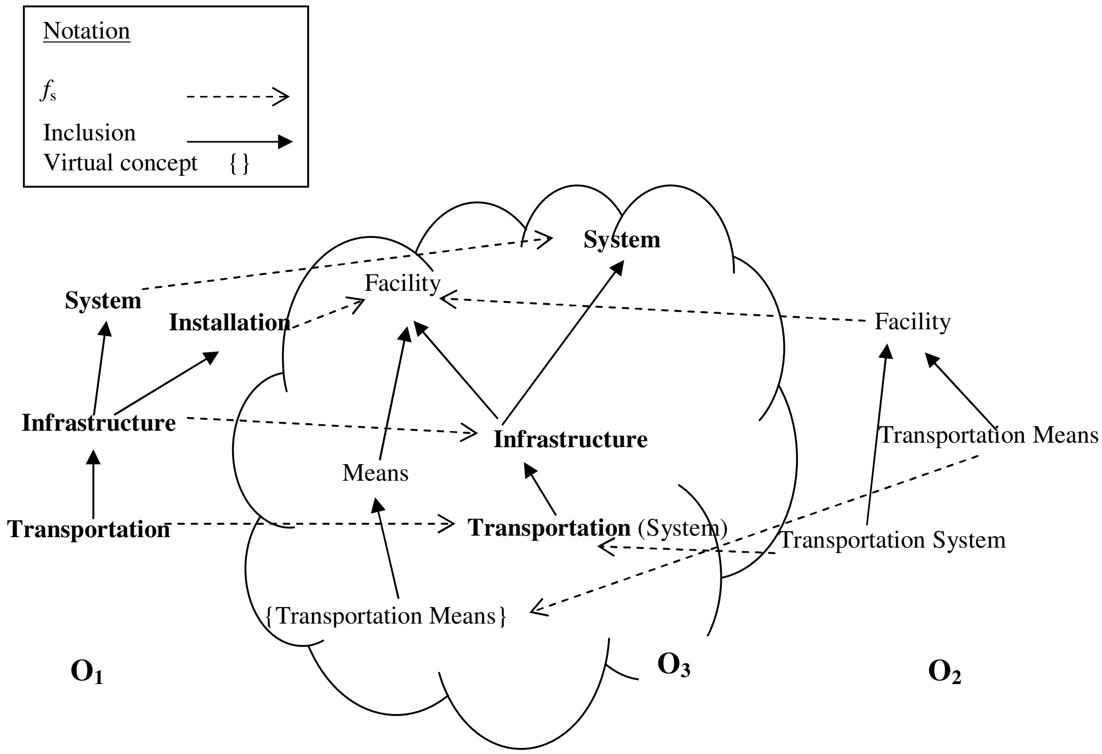


Fig. 7. *S-morphism* and the intermediate ontology

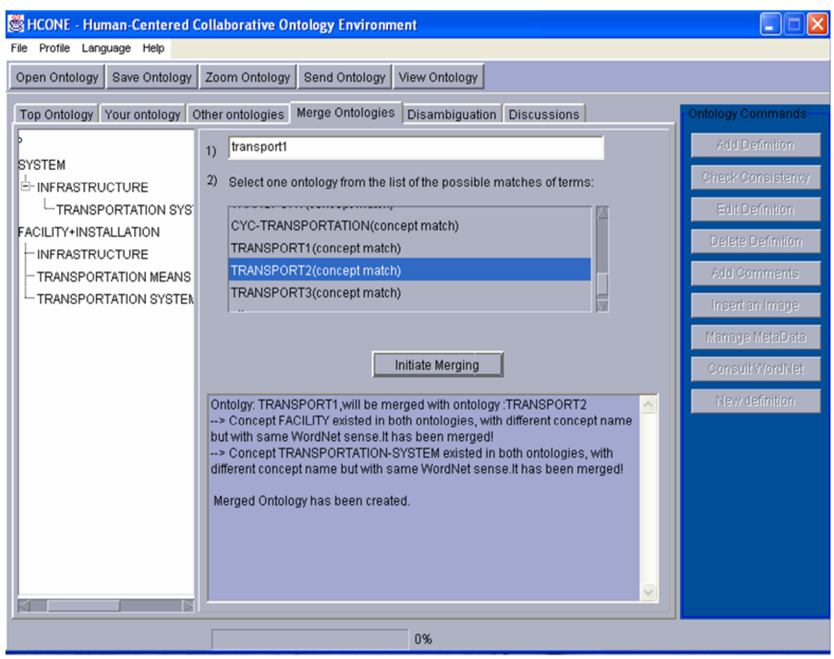


Fig. 8. The integrated to HCONE merge functionality. Merged concepts (e.g. FACILITY and INSTALLATION) are shown in the form Concept1+Concept2 (FACILITY+INSTALLATION) for presentation reasons

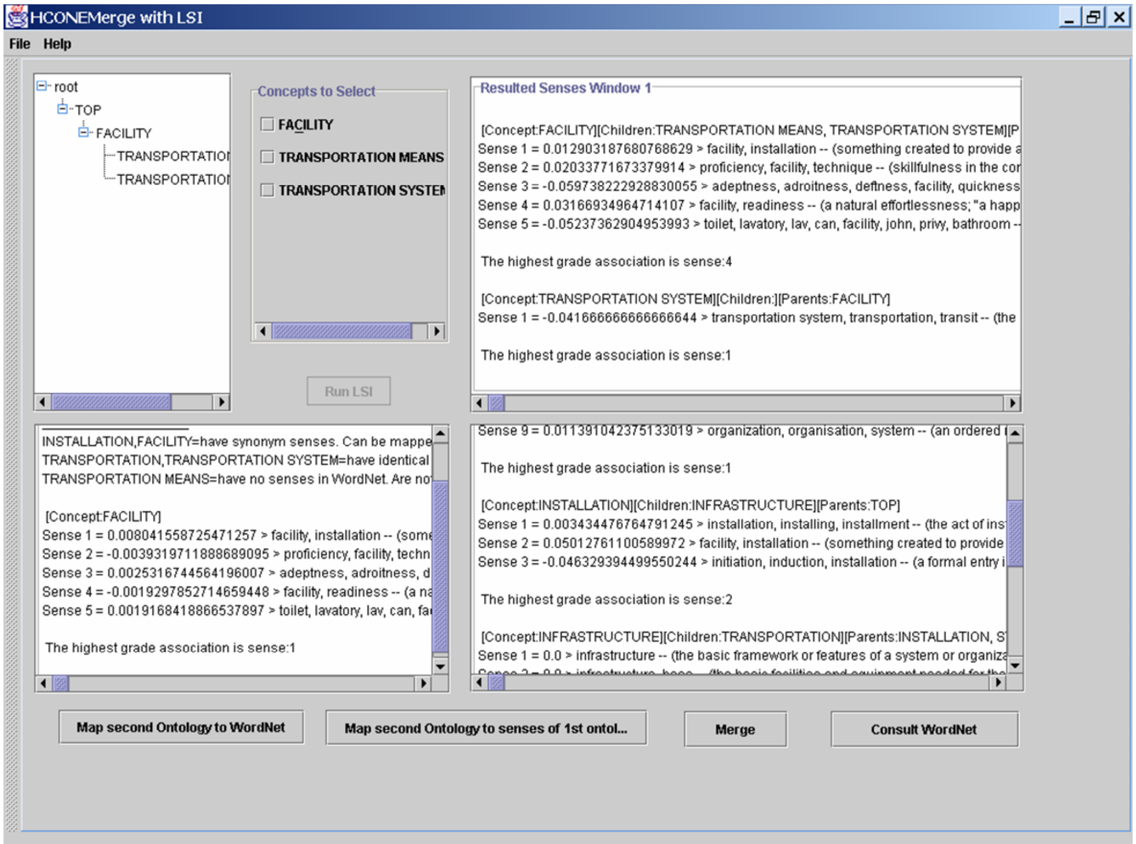
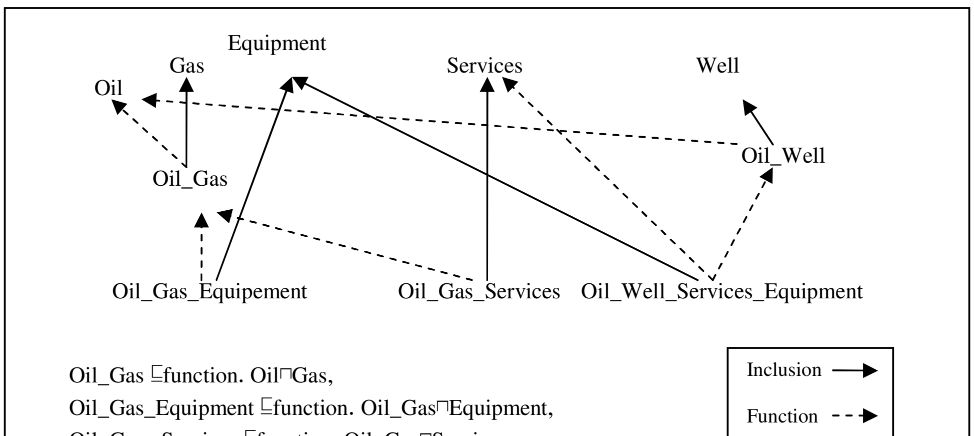


Fig. 9. Computing a mapping for the concept "Facility"



Oil_Gas $\sqsubseteq$function. Oil$\sqcap$Gas,

Oil_Gas_Equipment $\sqsubseteq$function. Oil_Gas$\sqcap$Equipment,

Oil_Gas_ Services $\sqsubseteq$function. Oil_Gas$\sqcap$Services,

Oil_Well $\sqsubseteq$function. Oil$\sqcap$Well,

Oil_Well_Services_Equipment$\sqsubseteq$function. Oil_Well$\sqcap$function. Services$\sqcap$Equipment

Fig. 10. Description Logic reasoner classification of "*Oil_Well_Services_Equipment"* concept

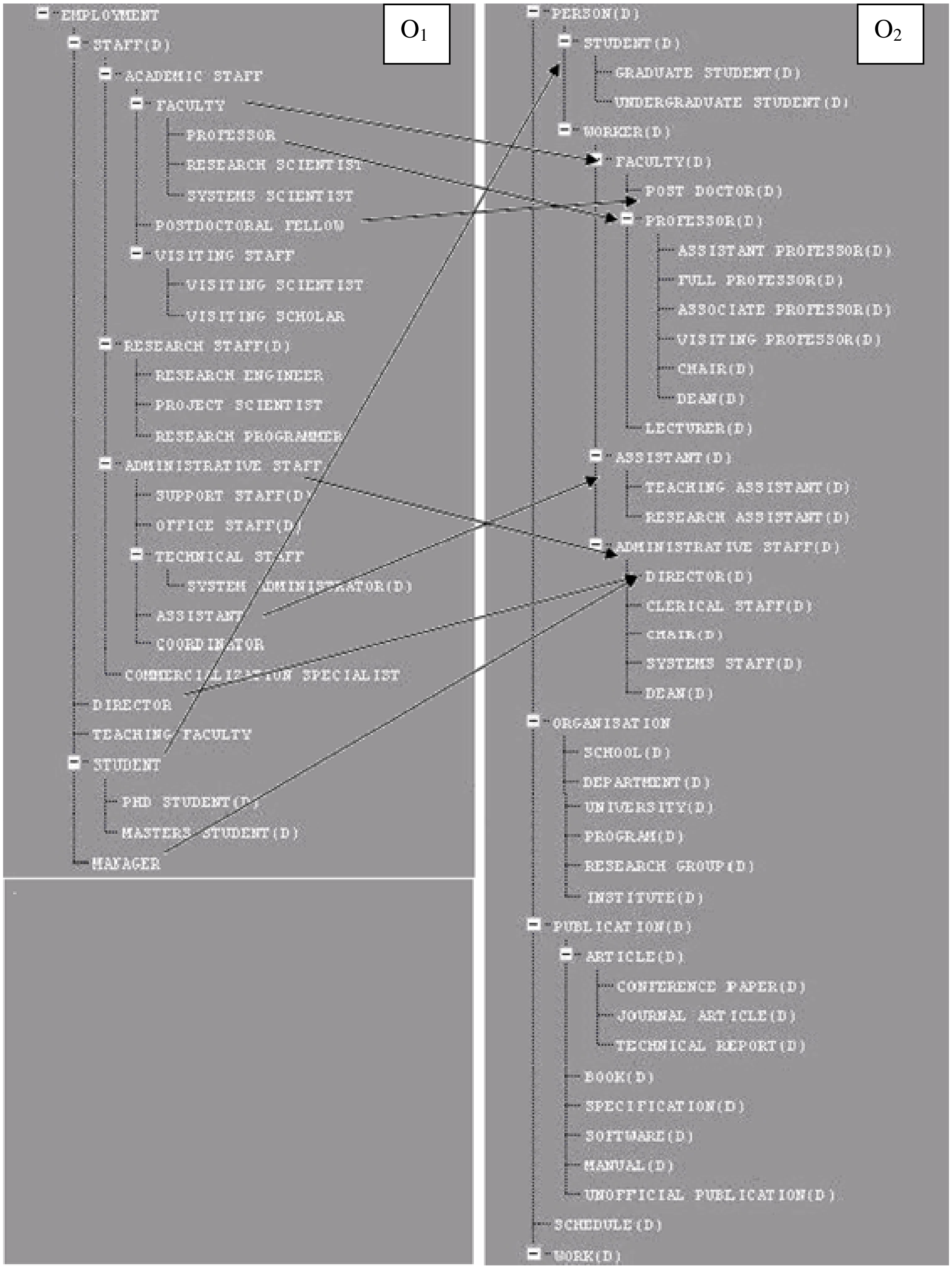


Fig. 11. Source ontologies and their mapping