Automatic Ontology Mapping for Agent Communication in an e-Commerce Environment

Marina Mongiello and Rodolfo Totaro

Dipartimento di Elettrotecnica ed Elettronica,
Politecnico di Bari, Bari - Italy
{mongiello, r.totaro}@poliba.it

Abstract. Internet-based e-commerce provides a high level of flexibility and openness though presenting many drawbacks due to the heterogeneity of the exchanged information. Ontologies are a key technology to solve many of the problems of e-commerce, in fact many companies use ontologies as a method of exchanging meaning between different agents. As ontology usage becomes more prevalent, the need for ontology reconciliation increases. In fact, ontology mapping methods can contribute to solve the problem of knowledge communication and interchange.

In this paper we present an automatic method for ontology mapping. The method is made up of two phases: a lexical-semantic analysis based on the WordNet thesaurus and a structural analysis based on a matching algorithm that finds semantic mappings between two ontologies expressed in Attributive Language with Number description ($\mathcal{ALN}$) Description logic. The mapped ontologies describe the same conceptualization through a set of rules that join related concepts. We deployed the proposed approach in a prototype system that currently is employed for large scale experiments. A simple experiment with a case study domain has shown a good correspondence with human mapping manually conducted and the system provided results.

1 Introduction

Web-enabled e-commerce helps user contact a large number of potential clients, hence it needs to be open to a large numbers of suppliers and buyers. However the open and flexible e-commerce requires to deal with the question of heterogeneity in the product, catalog and document description standards of the trading partners. Hence it is necessary to provide solutions to the problem of openness, heterogeneity and dynamic nature of the exchanged content, through the normalization, mapping and updating of the exchanged data. Intelligent solutions that help to mechanize the process of structuring, aligning and standardizing are key requisites to successfully overcoming the current bottlenecks of e-commerce and enabling its further growth. Ontology technology can solve many of this problems, in fact they are used as a method of exchanging meaning between different agents. As ontology usage becomes more prevalent, the need for ontology reconciliation increases. Ontology mapping methods can contribute to solve the problem of knowledge communication and interchange. Approaches to carry

K. Bauknecht et al. (Eds.): EC-Web 2005, LNCS 3590, pp. 21–30, 2005.
© Springer-Verlag Berlin Heidelberg 2005

out the mapping process either in a completely automated or semi-automatic fashion have been proposed. A comprehensive review of the state of the art in the semantic integration using ontologies is in [17,10]. However, let us recall the more relevant approaches. A semi-automatic mapping is proposed in [15]: SKAT system requires the interaction with a human expert to find the best set of mapping rules. A different approach is the method proposed in [20]. It can be applied when two agents with different ontologies would interact and have some common individuals in their knowledge bases. The method assumes that there are common individuals. A method based on a statistical approach is implemented in Glue [5] system that uses an automatic method for ontology mapping by estimating concepts similarity between ontologies comparing individuals stored in knowledge bases. Glue system works only on taxonomy of concepts in ontologies. Prompt [18] performs merging of ontologies. Ontomorph [4], Chimaera [13], Hical [9] use a variety of heuristics to match ontology elements. Anchor-Prompt [19] treats ontology as a graph with classes as nodes and slots as links. In [11] similarity between the nodes of two taxonomy is computed based on an TDF/IF model. In [8] the author reformulate the matching problem as that of propositional satisfiability.

In this paper we propose an algorithm for matching the concepts of two ontologies with the aim to completely automate the mapping. The algorithm is composed by a lexical-semantic analysis based on WordNet [7] thesaurus and a structural analysis based on a matching algorithm. The lexical-semantic analysis gives a measure of similarity between concept names in ontologies and returns the possible sets of mapping rules. After that, the structural analysis gives an evaluation of each set of mapping rules and chooses the best set.

With respect to existing methods, our approach is completely automated. This is an advantage since the manual specification of correspondences between concepts would be a time-consuming and error-prone process. Besides, it considers not only the hierarchical structure of concepts in the ontologies but also relations between them. This is important for conducting the structural analysis in which it is possible to reason also on the relations between concepts. A third feature of our method is the fact that it does not use individuals hence it is well-suited for reasoning also on not data-oriented ontologies. We deployed the proposed approach in a prototype system. While large scale experiments are in progress, a simple experiment with a case study ontology has shown a good correspondence with human mapping manually conducted and the system provided results.

The remaining of the paper is organized as follows. In Section 2 we briefly recall the formalism we use for representing ontologies and the tools for conducting lexical and structural analysis, after that we define the method through the description of the lexical and the structural analysis. Section 3 describes an application of the method in an applicative scenario concerning the apartment rental environment. Section 4 and 5 describe respectively our implementation and empirical evaluation. Last section concludes the paper and proposes further developments.

2 Two-Phases Ontology Mapping

In this section we describe the proposed method that is made up of two phases, a lexical-semantic analysis and a structural one. We use formalisms and tools to support our method. More precisely, we use Description Logics (DL) [6], [1] to describe ontologies and Classic reasoner [2] for the implementation of the algorithm developed for ontology mapping. The lexical-semantic analysis is based on WordNet [14,7] thesaurus. To determine the measure of semantic similarity between two concepts we use the Leacock and Chodorow measure [3]. It is based on the length of paths between concepts in a taxonomy and to WordNet for the representation of concepts.

Our method can be applied to ontologies described in $\mathcal{ALN}$ logic [6].

Lexical-semantic analysis. The first phase of the analysis is a lexical semantic analysis for comparing the names of concepts in ontologies. The analysis follows the method proposed by Cupid [12] through a normalization step and the evaluation of a similarity measure. Semantically similar concept names contain abbreviations, acronyms, punctuation, etc. that make them syntactically different. To make them comparable, according to Cupid method, we normalize them into sets of name tokens.

First of all we perform *Tokenization*, (parsing names into tokens based on punctuations and acronyms): the names are parsed into tokens by a customizable tokenizer using punctuation, upper case, special symbols, digits, etc. e.g. $POLines \Rightarrow \{PO, Lines\}$. The second step is *Expansion* (identifying abbreviations and acronyms) in which abbreviations and acronyms are expanded, e.g. $\{PO, Lines\} \Rightarrow \{Purchase, Order, Lines\}$. The third step is the *Elimination* (discarding prepositions, articles, etc.): tokens such as articles, prepositions or conjunctions are marked to be ignored during comparison. Concept names not included in WordNet are normalized like a composed-word. After that each concept is represented by a set of tokens. Finally we compute *Similarity measure.* Given two ontologies O_1 and O_2, let T_1 and T_2 be respectively a set of tokens in O_1 and O_2; we compute a similarity measure between the two sets as:

$$ns(T_1, T_2) = \frac{\sum_{t_1 \in T_1} [max_{t_2 \in T_2} sim(t_1, t_2)] + \sum_{t_2 \in T_2} [max_{t_1 \in T_1} sim(t_1, t_2)]}{|T_1| + |T_2|}$$

where $sim(t_1, t_2)$ is the Leacock-Chodorow similarity measure and t_1 and t_2 are tokens.

The resulting measure allows us to distinguish between different categories of similarity depending on the relation of the similarity with two threshold parameters α and β. We computed these parameters as an average of similarity measure between pairs of words that normally are considered synonyms and subsumee-subsumer by knowledge engineers respectively. We use $\alpha = 2.85$ and $\beta = 2.16$; anyway their value can be tuned in the prototype system. The sets that can be determined are *Synonym* (if $ns(T1, T2) > \alpha$); *Related* (if $\beta < ns(T1, T2) < \alpha$); *Not related* (if $ns(T1, T2) < \beta$); *Antinomy.* We consider a pair of sets as antinomy if there is a pair of tokens (t_1, t_2), $t_1 \in T_1$ and $t_2 \in T_2$, that WordNet considers antinomy. If a token is not indexed in WordNet, the semantic measure belongs to the set *Not related*, thus the choice depends on the structural analysis phase. Anyway in our prototype it is possible to adopt also different solutions.

Definition 1. *Given a concept C we denote with Syn(C) the set of synonyms of the concept C and with Rel(C) the set of concept related to C.*

Structural analysis and the distance algorithm. In this subsection we present the structural analysis and an algorithm for computing semantic distance between concepts.

Discovering potentially acceptable mappings. First of all we determine all the potentially acceptable mappings. To this purpose let us give some useful definitions.

Definition 2. *Given two ontologies O_1 and O_2 and a concept C in O_1, A(C) is the set of possible associations of the concept C with concepts of O_2. A(C) is defined as follows: $A(C) = Syn(C)$ if $Syn(C) \neq \emptyset$ and $A(C) = Rel(C) + \{\perp\}$ if $Syn(C) = \emptyset$*

Definition 3. *Given two ontologies O_1 and O_2, a mapping rule is a pair (C,D) where C is a concept of O_1 and $D \in A(C)$. $rule(O_1, O_2)$ is the set of mapping rules between O_1 and O_2.*

Definition 4. *An empty rule is a mapping rule (C, D) with $D \equiv \perp$*

The *empty rule* represents the possibility that the destination ontologies do not provide a corresponding concept in a source ontology.

Definition 5. *Given two ontologies O_1 and O_2 to map, a potentially acceptable mapping is a set of mapping rules defined as follows:*

$$Map = \{(C_i, D) \in rules(O_1, O_2) | i = 1, ...n_{O_1} \wedge C_1 \neq C_2 \neq ...C_{n_{O_1}}\}$$

where n_{O_1} is the number of concepts in the ontology O_1.

After having considered rules between concepts, rules between roles should be considered. They are determined considering that there is a match between pair of roles that joins pairs of matching concepts. Each potentially acceptable mapping is completed with rules between roles according to the previously defined rule, that have a similarity measure greater than a prefixed threshold.

Evaluation of potentially acceptable mappings. The evaluation of a potentially acceptable mapping is obtained through an auxiliary ontology. The concept names in the source ontology are replaced with the names of matched concepts in the destination one according to the rules of the potentially acceptable mappings. Names are replaced only in the right side of the concept definitions. Rules between roles are applied only if they have the same direction in the two ontologies. The number of times in which these rules cannot be applied is computed and used as a parameter for the mapping evaluation.

During the previous phases inconsistences can be determined in the knowledge base, this can happen while loading the destination and auxiliary ontology in the same KB; in this case uncorrect mappings are marked.

The measure for the evaluation of the correctness of potentially acceptable mapping is computed as average of concept distance, i.e. the structural similarity between two concepts in a rule. Such a value is returned by an algorithm we will describe later. In fact, since concepts in the destination and auxiliary ontologies use the same vocabulary, for a given rule in the potentially acceptable mapping we can compute the distance between two DL descriptions. Evaluation of Potentially acceptable Mappings, *EPM* is computed using the following measure:

$$EPM = \frac{\sum_{i=1}^{N} conceptDistance(C_i^1, C_i^2) - 2 \cdot K \cdot N_R}{2 \cdot N}$$

where N is the number of rules between not atomic concepts that are in the potentially acceptable mapping, N_R is the number of not applicable rules between roles and K is a weight varying in the range $[0,1]$. The obtained measure is an average between *conceptDistance* values corrected by $K * N_R$ since a not applicable rule gives an increment of the *conceptDistance* value. The value is returned by algorithm *conceptDistance*. K is a parameter whose value can be set through the system interface. Let us now describe the algorithm; it extends an algorithm for matchmaking that some of us contributed to define [16], and provides a symmetric measure of concept distance. The algorithm applies to normal form of input concepts.

Algorithm *conceptDistance*(C, D);
input CLASSIC concepts C, D, in normal form,
output distance $n \geq 0$, where 0 means that $C \equiv D$
begin algorithm
  if $C \sqcap D$ is not satisfiable
    return ∞
    let $n := 0$ **in**
      /* add to n the number of concept names in D */
      /* which are not among the concept names of C and viceversa */
      1. $n := n + |D_{names+} - C_{names+}| + |C_{names+} - D_{names+}|$;
      /* add 2 to n for each number restrictions of D */
      /* that is not in C */
      2. **for each** concept $(\geq\ x\ R) \in D_\sharp$
      such that there is no concept $(\geq\ y\ R) \in C_\sharp$ with $y \geq x$
          $n := n + 2$;
      3. **for each** concept $(\leq\ x\ R) \in D_\sharp$
      such that there is no concept $(\leq\ y\ R) \in C_\sharp$ with $y \leq x$
          $n := n + 2$;
      /* for each universal role quantification in D */
      /* add the result of a recursive call */
      4. **for each** concept $\forall R.E \in D_{all}$
        if there does not exist $\forall R.F \in C_{all}$
          then $n := n + conceptDistance(\top, E) + conceptDistance(\top, F)$;
          else $n := n + conceptDistance(F, E)$;
      return n;
end algorithm

Selection of the final mapping. Among all the evaluated potentially acceptable mappings we select the final mapping as the one that satisfies some conditions.

First of all it must have the minimum value of minimum EPM; the second condition to be verified is the minimum number of rules between concepts (for a given EPM); finally it must have the maximum average of the lexical-semantic similarity measure for the rules that appear in the potentially acceptable mappings. The second condition is necessary because if adding a new mapping rule to a possible mapping its EPM does not decrease, the new rule may be uncorrect. In this case, for a given EPM, the mapping with a minor number of rules is selected.

3 Scenario

In this Section we describe an application of the proposed method in a e-commerce scenario as shown in Figure 1. Let us consider two or more agents for e-commerce that can communicate; each agent has an engine and manages several ontologies concerning different domains. The knowledge base concerning a particular domain stores several descriptions of requests. The agent receives a request by an end-user, searches for a description in its knowledge base that satisfies user's request. If the search does not provide satisfactory results, the agent may try to find in an external knowledge base a description better fitting the request. Hence it could forward the request to another agent. To this purpose the request should be mapped in the corresponding ontology. Once determined the mapping rules between the two ontologies, the source description is translated in the destination description. The agent that manages the destination ontology searches for a description that satisfies the request. Finally, the set of results is sent to the client and the final selection is submitted to the end-user. As an example, we refer to an apartment rental environment. The agent should satisfy user's requests concerning, for example, a searched apartment, if the request could not be satisfied, it can be forwarded to another agent.

Let us now describe how to use mapping rules to translate descriptions of concepts belonging to the source ontology to descriptions of corresponding concepts

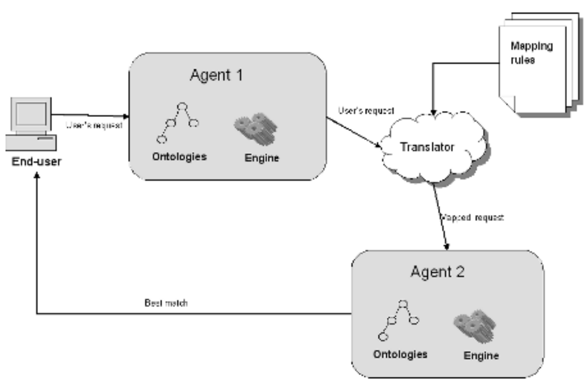

Fig. 1. Scenario

Table 1. How to use mapping rules

Source Concept	**Rules**	**Translated concept**
$Flat$	$Flat->apartment$	$apartment$
$\forall hasRoom.SingleRoom$	$SingleRoom->single\_room$ $hasRoom->contains\_rooms$	$\forall contains\_room.single\_room$
$(\geq\ 2\ hasRoom)$	$hasRoom->contains\_rooms$	$(\geq\ 2\ contains\_room)$
$\forall occupants.Student$	$occupants->occupants$ $Student->student$	$\forall occupants.student$
$\forall hasFacilities.ADSL$	$ADSL->adsl$	$---$

belonging to the destination ontology. A concept description is a conjunction of concepts S_i of the source ontology: $C \equiv S_1 \sqcap S_2 \sqcap ... \sqcap S_n$. The translated description will represent only concepts S_i for which there exist mapping rules for all concepts used in the description of S_i. As an example, let us consider the mapping rules described in Table 1 and the following description:

$$Flat \sqcap \forall hasRoom.SingleRoom \sqcap (\geq\ 2\ hasRoom) \sqcap \forall occupants.Student \sqcap$$
$$\sqcap \forall hasFacilities.ADSL$$

Translation of a description is performed by considering for each concept in the conjunction the proper rules. A concept is omitted if the corresponding rule is not available, thus obtaining a more general description that is anyway useful for retrieval. Using mapping rules, we obtain the following translated description: $apartment \sqcap \forall contains\_room.single\_room \sqcap (\geq\ 2\ contains\_room) \sqcap \forall occupants.student$ In the previous description, "$\forall hasFacilities.ADSL$" is not present in the translated description, since a rule for "$hasFacilities$" is not available.

4 A Prototype System

To validate the proposed approach a prototype system for mapping between ontologies has been implemented. The system, as shown in the component diagram in Figure 2, embeds NeoClassic reasoner and WordNet dictionary.

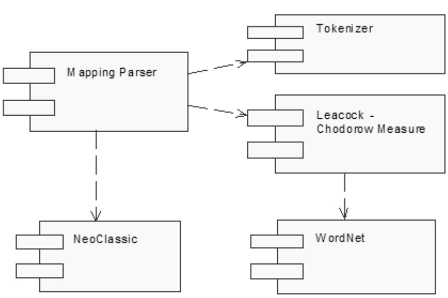

Fig. 2. Component diagram of the prototype system architecture

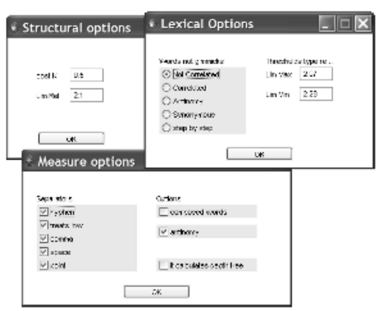

Fig. 3. Options

The module Mapping Parser extracts concepts from the two ontologies, uses the two modules Leacock-Chodorow measure and the Tokenizer. It computes semantic similarity between concept names, extracts and evaluates the potentially acceptable mappings through the concept distance algorithm and NeoClassic reasoner, a C++ implementation of the CLASSIC reasoner. The Tokenizer module implements the tokenization described in the Section 2; we can configure Tokenizer options through the Measure Options Window shown in the Figure 3 that shows also the Lexical Options Window and the Structural Options one. In the Lexical Options Window we can set α and β thresholds (respectively Lim Min and Lim Max in the figure) or choose a default set for the measures that can not be computed, if a term is not indexed in WordNet. In the Structural Options Window we can set the K parameter and the threshold for semantic relation similarity.

5 Empirical Evaluation and Discussion

The algorithm has been evaluated in the apartment rental domain. The test was performed on three ontologies created by different expert users. This choice of different users was due to the need of ensuring different models of the given domain, in fact, generally different person models in different way the same domain. This ensures that the ontologies are different both for the vocabulary for expressing concept names and for the choice of relations between concepts. Let us refer to the following example:

Ontology 1: $SeaVilla \equiv House \sqcap \forall isLocated.Sea$; $Flat \sqsubseteq House$;

Ontology 2: $SeaHouse \sqsubseteq House$; $Apartment \sqsubseteq House$.

To evaluate the performances of the algorithm, we compared the set of mapping rules returned by the algorithm with those of a manually conducted mapping.

We chose two parameters to evaluate the performance of the algorithm: the percentage of exact mapping rules, i.e. perfect match and the percentage of acceptable mapping rules, i.e. acceptable rules, obtained as the sum of exact mapping rules and imprecise ones.

Exact mapping rules are those in which the system returns the same rules selected in the manual mapping. Imprecise rules are those found by the system that are not uncorrected but that anyway are not the best ones.

In the previous example, the relation between *SeaHouse* in the second ontology and *House* in the first one is assumed as an imprecise rule since *SeaHouse* is however a *House* but it could be better associated to *SeaVilla* in the first ontology.

We performed the mapping on the three ontologies, determining a range of 63%-81% for Perfect Match and 71%-88% for Acceptable match. Figure 4 demonstrates the results for the various mappings.

Such a variability in the percentage can be explained by considering the single rules found out by the algorithm. It was noticed that the algorithm behaves in a more efficient manner when the compared ontologies are strongly connected inside, hence there is a high number of relations between concepts. On the other

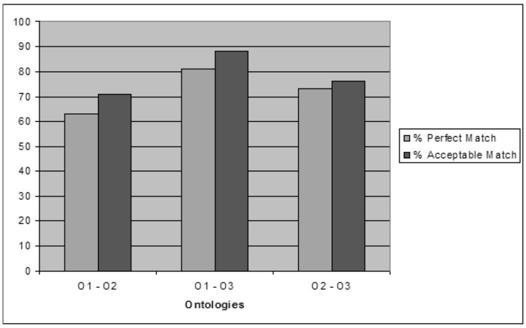

Fig. 4. Empirical evaluation

hand, it is less effective when ontologies have a predominant hierarchical structure, hence the usefulness of the structural analysis is strongly reduced.

Experiments highlighted the ability of the system to find relations between concepts having different semantic but the same name. In fact the system finds the correct associations through structural analysis, though the algorithm found several alternatives for those terms during the lexical analysis and the ontologies are strongly connected.

The parameter K was set to 0.7 in the performed test; it was used for managing inverse relations in the auxiliary ontologies, since such type of relations can not be expressed in the adopted $\mathcal{ALN}$ logic.

6 Conclusion and Future Work

We described an automatic method for semantic mapping of ontologies. The method consists of two phases, a lexical-semantic analysis and a structural one. An application in an e-commerce scenario of the proposed algorithm is described. We deployed the proposed approach in a prototype system. An experiment with a case study domain has shown a good correspondence with human mapping manually conducted and the system provided results. We are currently working on an extension of the distance algorithm to more expressive logics; besides we aim to refine the proposed method in order to reduce the computational time of the structural analysis.

The authors would like to thank Francesco M. Donini, A. Halevy and Natasha Noy for their thoughtful and helpful comments on earlier draft of the paper.

References

1. F. Baader, I. Horrocks, and U. Sattler. Description logics. In S. Staab and R. Studer, editors, *Handbook on Ontologies in Information Systems*, pages 385–405. Springer, 2003.

2. R. J. Brachman et al. Living with CLASSIC: When and How to Use a KL-ONE-Like Language. In J. Sowa, editor, *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, pages 401–456. MK, 1991.
3. Leacock C. and Chodorow M. Combining local context and wordnet similarity for word sense identification. In Fellbaum, editor, *WordNet an electronic lexical database*, pages 265–283. MIT Press, London, England, 1998.
4. H. Chalupsky. Ontomorph: a translation system for symbolic knowledge. In *Principles of knowledge representation and reasoning*, 2000.
5. A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Ontology matching: A machine learning approach. In S. Staab and R. Studer, editors, *Handbook on Ontologies in Information Systems*, pages 385–405. Springer, 2003.
6. F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in Description Logics. In Gerhard Brewka, editor, *Principles of Knowledge Representation*, Studies in Logic, Language and Information, pages 193–238. CSLI Publications, 1996.
7. C. Fellbaum, editor. *Wordnet: an electronic lexical database*. MIT press, 1999.
8. F. Giunchiglia and P. Shvaiko. Semantic matching. *Knowledge Engineering Review Journal*, 18(3):265–280, 2004.
9. R. Ichise, H. Takeda, and S. Honiden. Rule induction for concept hierarchy alignement. In *Proc. of workshop on ontology learning at IJCAI 2001*, 2001.
10. Y. Kalfoglou and M. Schorlemmer. Ontology mapping: the state of the art. *The knowledge engineering review*, 18(1):1–31, 2003.
11. M. Lacher and G. Groh. facilitating the exchange of explicit knowledge through ontology mappings. In *Proc. of FLAIRS 2001*, 2001.
12. J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *Proceedings of the 27th VLDB Conference*, Roma, Italy, 2001.
13. D. McGuinnes, R. Fikes, J. Rice, and S. Wilder. The chimaera ontology environment. In *Proc. of AAAI 2000*, 2000.
14. G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Introduction to wordnet: an on-line lexical database. *Journal of Lexicography*, 3(4):235–244, 1990.
15. P. Mitra, G. Wiederhold, and J. Jannink. Semi-automatic integration of knowledge sources. In *Proc. of Fusion '99*, Sunnyvale, USA, 1999.
16. T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. A system for principled Matchmaking in an electronic marketplace. In *Proc. of WWW-03*, pages 321–330, 2003.
17. N. Noy. Semantic integration: a survey of ontology-based approaches. *Sigmod record*, 33(4), 2004.
18. N. Noy and M. Musen. Prompt: an algorithm and tool for automated ontology merging and alignment. In *Proc. of AAAI 2000*, 2000.
19. N. Noy and M. Musen. Anchor-prompt: using non-local context for semantic matching. In *Proc. of workshop on ontologies and information sharing at IJCAI 2001*, 2001.
20. F. Wiesman, N. Roos, and P. Vogt. Automatic ontology mapping for agent communication. In G. Schreiber & M. van Someren (Eds.) B. Krose, M. de Rijke, editor, *Proc. of BNAIC2001*, pages pp. 291–298, Amsterdam: Universiteit van Amsterdam., 2001.