Framework for Automatic Generation of Ontology Mappings

Gonzalo Cañadas[(1)], Mariano Fernández-López[(1)], Raúl García-García[(1)], Manuel Lama[(2)], Alfredo Sánchez-Alberca[(1)], Carlos Óscar S. Sorzano[(1)]

(1) Escuela Politécnica Superior. Universidad San Pablo CEU
Ctra. de Boadilla del Monte km 5,300
28668 - Boadilla del Monte. Madrid. Spain
{gcanadas, mfernandez, rgg}.eps@ceu.es, asalber@ceu.es

(2) Departamento de Electrónica y Computación. Facultad de Física.
Campus Sur, s/n. Universidad de Santiago de Compostela,
15782 Santiago de Compostela. A Coruña. Spain
lama@dec.usc.es

Abstract. Some of the most outstanding problems in Computer Science (e.g. access to heterogeneous information sources, use of different e-commerce standards, ontology translation, etc.) are often approached through the identification of ontology mappings. A manual mapping generation slows down, or even makes unfeasible, the solution of particular cases of the aforementioned problems via ontology mappings. Some algorithms and formal models for partial tasks of automatic generation of mappings have been proposed. However, an integrated framework to solve this problem is still missing. In this paper, we present a framework for automatic ontology mapping generation, and a partial implementation of it. Our proposal is that this integrated vision can guide, not only our future work, but also the future work of other researchers. In the implementation carried out, we have built a mapping ontology with knowledge on ontology mappings.

1. Introduction

An *ontology* is a reusable and shareable vocabulary coded in such a way it can be processed by a computer. That is, *ontologies* aim to capture consensual knowledge of a given domain in a generic and formal way, so that it can be reused and shared across applications and by groups of people. From this definition we could wrongly infer that there is only one ontology for modeling a domain. However, we can find on the literature several ontologies that model similar knowledge in different ways [14]. For instance, in the e-commerce field, there are several standards and joint initiatives for the classification of products and services (UNSPSC, e-cl@ss, RosettaNet, NAICS, SCTG, etc.). Of course, the case of e-commerce is not unique, since it also happens in medicine, law, art, sciences, etc. Besides, the existence of different standards is not the only reason why the resolution of a problem may require the manipulation of different ontologies modeling similar knowledge. Thus, the language translation of an ontology needs to deal with two ontologies (the input and the output one) [6, 8, 12], or ontology evolution needs to deal with several ontologies (the different versions of the original ontology) [33]. Even when there are no ontologies given *a priori*, the resolution of a problem may require the manipulation of heterogeneous information sources. If the problem is approached through the construction of ontologies, the heterogeneity of information usually leads to heterogeneous ontologies [26, 35].

Whichever is the case of use of different ontologies of the same domain, they are usually linked through mappings. A *mapping* between ontologies is a function that associates terms and expressions defined in a source ontology with terms and expressions of a target ontology [35]. Currently, mappings between ontologies are identified by hand. This leads to the following drawbacks: (1) the generation of mappings between large ontologies or among a large amount of different ontologies consumes a huge quantity of resources; and (2) if some of the ontologies changes, the generation has to be carried out manually again. As a consequence, a satisfactory solution is sometimes unfeasible. The Semantic Web is a good sample scenario where automatic ontology mapping is absolutely required. According to Berners-Lee [4], the *Semantic Web* is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. To attain the Semantic Web, the web pages are annotated with ontologies. Such ontologies are different and ever changing. Moreover, the number of ontologies to annotate a significant part of the Web pages is huge. Consequently, the automatic generation of mappings is essential in the future of the Semantic Web.

In this paper, we present a three layer framework for automatic ontology mapping generation (see figure 1), and the implementation of part of such a framework. Our proposal is that this integrated vision can guide, not only our future work, but also the future work of other researchers.

The middleware layer is the one in charge of the automatic generation of ontology mappings. The structure of the middleware layer is an evolution of the structure of classical knowledge based systems. The role of the classical knowledge base is played by a mapping ontology. We also propose a module to learn new rules of ontology mapping generation, and to modify the former ones. Such module should work with the supervision of a user.

The service layer uses the mappings to perform tasks where links between ontologies are useful (ontology translation, expression translation, etc.). Finally, the application layer uses the services of the former layer in sophisticated applications (integration of heterogeneous databases, semantic web services, etc.).

The current implementation of the proposed framework includes a first version of the mapping ontology (with the purpose of being used by other researchers in the field), the first version of the database, some similarity measure procedures, and the integration of basic machine learning algorithms.

Section 2 presents a brief state of the art on automatic generation of ontology mappings. Section 3 presents our framework. Section 4 shows the current implementation of the framework. Finally, section 5 presents the conclusions and the future lines.

2. A brief state of the art on automatic generation of ontology mappings

Former works to solve the problem of automatic generation of ontology mappings can be divided into two categories: formal contributions and material contributions. Formal contributions deal with the problem conceptualization, while material contributions provide software systems addressing the problem.

Formal contributions. Ontologies are modeled using graphs [13, 25, 30], logic theories in logic notation [18], frame based models [22, 32], etc. Concerning the conceptualization of mappings, they are represented using morphims between graphs [30] morphisms between logic theories [18], and relations between classes [22, 32]. A graph morphism is a function that preserves the structure, and a theory morphism is a function that preserves the axioms. The conceptualization of mappings is sometimes carried out through an ontology. Thus, some authors have elaborated mapping ontologies to provide support to their mapping applications [22, 32].

Concerning the mapping generation methods, they basically operate in two phases. Firstly, they specify how to establish how similar the terms of an ontology are to the terms of another ontology. Then, according to the similarity measures obtained in the first step, the methods specify how to generate the mappings. Each of the features of the concepts to be mapped can be used to calculate the similarity measures between the concepts of two ontologies [29]: the name (lexical similarity) [20, 23], the natural language description (in the ontology, in thesauri, in documents, etc.) [29], the structural relations (e.g. subclass of) [13, 23], the instances in databases [29], etc.

Material contributions. Currently, the most outstanding software systems that automatically generate ontology mappings are ONION [30], MAFRA [22], IFF [18], and Ehrig and Sure's system [11]. ONION generates mappings using graph transformations. MAFRA combines different similarity measures, both lexical and structural, to establish the mappings. IFF is based on morphisms between logical theories. Ehrig and Sure's system determines similarity through rules that have been encoded by ontology experts. There are other systems that automatically generate mappings as an intermediate step to carry out other task, for example, PROMPT [31] and FCA-Merge [34] for ontology merging. The mappings are established by extracting, from the documents, instances that belong to concepts of both ontologies. Finally, we would like to mention MetaMap [1], which maps texts in natural language into medical ontologies.

The aforementioned proposals have the following drawbacks:

1) Some algorithms and formal models for partial tasks of automatic generation of ontology mappings have been proposed. However, no integrated framework to solve the automatic ontology mapping generation problem has been provided.
2) None of the approaches take advantage of the integrated use of knowledge and information in databases, ontololgies, thesauri, Web pages, plain texts, etc.
3) None of the approaches is integrated in more general problems (access to heterogeneous information sources, use of different e-commerce standards, ontology translation, etc.). Consequently, it is difficult to find the ontology mapping generation inside distributed systems.

See [19] for a more thorough review on ontology mappings.

3. The proposed framework

According to the study of the work of other authors, and according to our own experience in the problem, we think that the automatic ontology mapping generation has the following features:

a) The problem is manually solved by people with experience on ontological engineering and/or on the domain of the ontologies.
b) The ontology mapping generation requires the manipulation of symbolic knowledge (e.g. `concept 1 and concept 2 are similar enough to establish a mapping between them`).
c) Heuristics are required to limit the search space. Mappings can involve combinations of terms (e.g. `concept 1 is similar to the intersection of concept 2 and concept 3`) and combinations of similarities measures (e.g. if $sim_1(C_1,C_2)>0.8$ and $sim_2(C_1,C_2)>0.5$ then a mapping should be established between C1 and C2). Taking into account that it is not feasible to consider the whole set of combinations, search guides are needed to prune the worst options.
d) The ontologies to be mapped are often incomplete (significant attributes may not be modeled, or the structure of the ontology may be poor). Therefore, it is a problem with incomplete information.

That is, it is a typical problem to be solved with a knowledge based system [15]. Therefore, the structure of the core of our framework is similar to the one of classical knowledge based systems, excepting that we use an ontology instead of a traditional knowledge base.

We assume that the ***external sources*** of the system are, at least (see figure 1): the ontologies to be mapped (*source ontology* and *target ontology*), *plain texts* and *Web pages* describing the concepts defined by the ontologies, *databases* with instances of the ontologies, *external resources* (other ontologies; thesauri; lexical databases, for example, WordNet [27]; etc.), *mapped ontologies* (which can be used in mapping generation rule learning), the *supervision* of the user and the *point of view* provided by the user. The supervision consists in a series of modifications of the generated ontology mappings. The point of view allows establishing mappings combining different approaches: analysis of the attributes of the concepts, analysis of their instances, analysis of the concepts taxonomy, etc.

Being inspired by the framework proposed in [2] for ontology development platforms, we propose a framework in three layers:

Ontology mapping middleware. It is the core of the system, since it is the one really performing the automatic generation of ontology mappings. This layer is composed by the following elements:

- *Inference engine*. It reasons using the knowledge provided by the rest of the modules of the middleware.
- *Wrappers*. They deal with formats and protocols of external sources so that they can be manipulated inside the system.
- *Similarity measure procedures*. They perform the first phase in the mapping generation. The inference engine can execute them using the information provided by the point of view provided by the user. The software system should

be prepared so that similarity measure procedures could be integrated in run time. We can distinguish three main kinds of analysis to obtain the similarity measure:

- *The ontology schema analysis module*. This module processes the data schemas followed by the source and the target ontology, the databases on the domain of the ontologies, the structure of the Web pages, etc. A schema analysis may determine, for instance, how similar is the table that represents the concept C_1 in a database to the table that represents the concept C_2 in another database. Let's note that the links between concepts and database tables should have been previously identified. This problem is known as mappings between ontologies and databases, and its details fall outside the scope of this paper (this problem is tackled by Barrasa and colleagues in [3]).
- *The ontology content analysis module*. This module processes the instances of the input ontologies that appear in plain texts, databases, Web pages, etc. A content analysis can determine, for example, how similar is the set of instances of the concept C_1 to the set of instances of the concept C_2. Such instances may be stored in a database.
- *The mapping analysis module*. It uses mappings previously generated by both our system or external systems.

- *The mapping ontology*. It models the knowledge on ontology mappings. Consequently, it models concepts manipulated in ontology mapping generation: mapping, similarity measure, point of view of the mapping, concept, etc. The mapping ontology also contains rules, for example: "if the similarity between the concept C_1 and the concept C_2 is greater than 0.8, then a mapping between C_1 and C_2 is established".
- *The mapping database*. It stores the generated mappings so that they can be used by different modules, specially the ontology mapping based services and the learning module
- *The ontology mapping supervision module*. It allows the user to carry out changes when (s)he disagrees with the automatically generated mappings.
- *The ontology mapping learning module*. It modifies the rules of mapping generation from user supervision. That is, the disagreement of the user in the mapping obtained can provoke the change of rules that have generated the mappings. Moreover, this module can learn new rules and instances of the mapping ontology from ontologies that have been mapped by our system and by others.

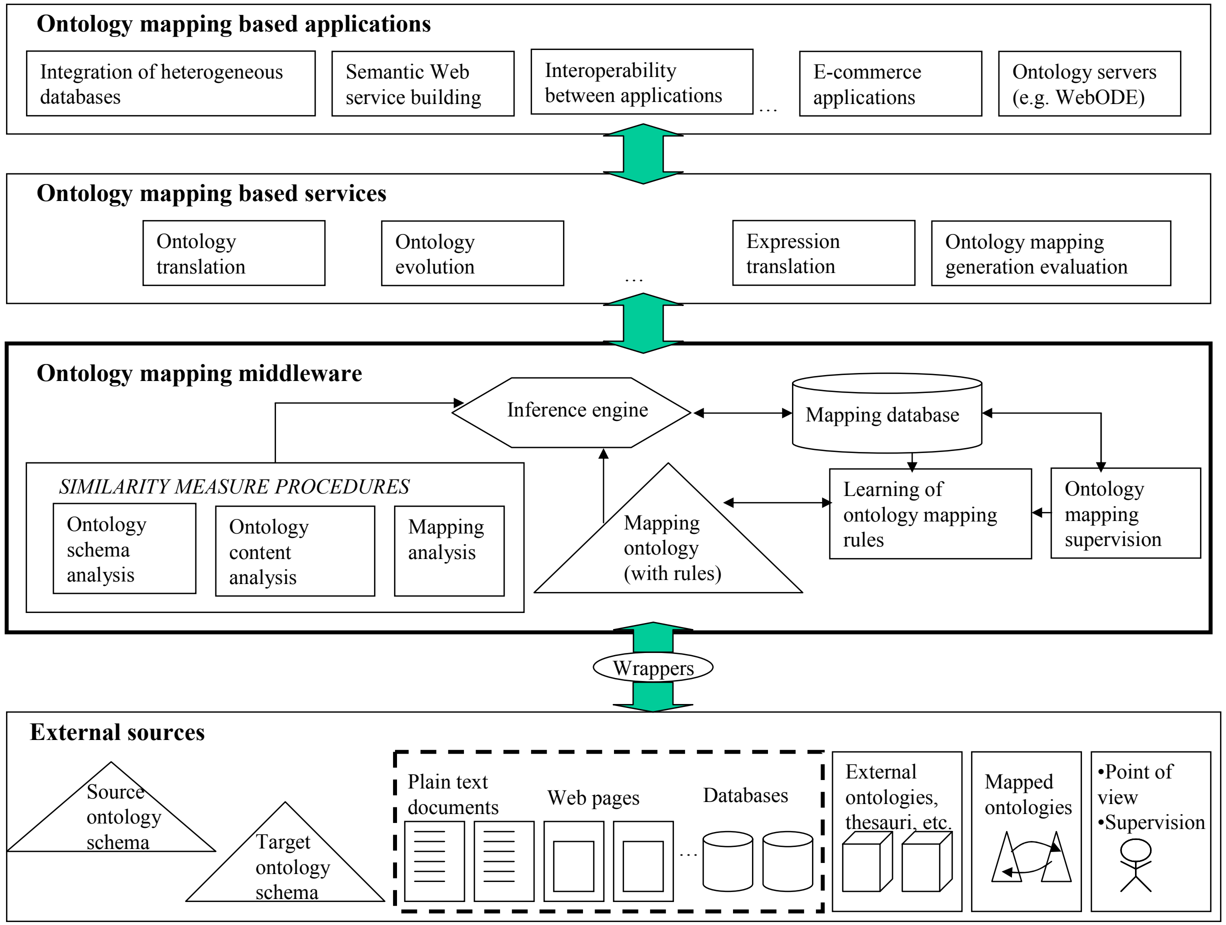

Figure 1. Framework for automatic generation of ontology mappings

Ontology mapping based services. They use the generated mappings to solve problems that require links between ontologies. Examples of these services are:

- *Ontology translation*. The use of mappings for ontology translation is based on the concept of knowledge representation (KR) ontology. A KR captures the representation primitives used to formalize knowledge under a given KR paradigm. The most representative examples are the *Frame Ontology* [16] and the *OKBC Ontology* [7], both available in the Ontolingua Server[1]. They provide formal definitions of the representation primitives used mainly in frame-based languages (i.e., classes, subclasses, attributes, values, relations and axioms). Besides, they allow building other ontologies by means of frame-based conventions. Thus, if we have mappings between KR_1 and KR_2, we can translate every ontology modeled according to KR_1 into another ontology modeled according to KR_2 [8].
- *Expression translation*. Let's suppose, for instance, that the ontology O_1 represents the schema of the database DB_1, and O_1, the schema of the database DB_2. A query database that uses the vocabulary of the ontology O_1 can be expressed using the vocabulary of the ontology O_2. The translation can be carried out using mappings between both ontologies.
- *Ontology evolution*. Mappings between different versions of the same ontology can be established. This can be useful, for instance, to allow the interoperability between applications using different versions of an ontology.
- *Ontology mapping generation evaluation*. The comparison of results using different approaches to generate ontology mappings is absolutely required to assess the accuracy and usefulness of a mapping generation approach.

Ontology mapping based applications. Sophisticated applications can be built using the ontology mapping services, for example:

- *Integration of heterogeneous databases*. Different databases can be integrated through global and local ontologies linked via mappings [26, 35].
- *Semantic Web service building*. Semantic Web services can be built using problem solving methods [9]. According to this approach, there is an ontology that models the method that has to be mapped to the ontology that models the domain where the method is applied. For example, a problem solving method of diagnosis of car motors has to use mappings between a method ontology on diagnosis and a domain ontology on car motors.
- *Interoperability between applications*. In order to interoperate two applications using ontologies, their ontologies should be mapped, i.e., a correspondence between their vocabularies should be established.
- *E-commerce applications*. Currently, different e-commerce standards are used. As Corcho and Gómez-Pérez proposed [10], e-commerce standards can be mapped so that they can be used by the same applications.
- *Ontology servers*. Ontology servers for ontology development perform ontology maintenance, ontology translation, etc. require ontology mappings. In fact, the system described in this work is intended to interoperate with WebODE [2].

---

[1] http://ontolingua.stanford.edu/

4. The current implementation of the proposed framework

AMON (Agent for Mapping Ontologies in the Network), our software system, currently implements the following elements:

- *Inference engine*. The application uses the Python interpreter 2.3.2[2] and the Oracle SQL processor 9.2.0.1.0[3]. However, in future versions, we will use the Ciao Prolog interpreter to execute the mapping generation rules [17].
- *Similarity measure procedures*. At present time, we have implemented and integrated several classical similarity measures in the system, for example, Jaccard coefficient, Dice coefficient, matching coefficient, cosine coefficient [23], edit distance similarity measure (based on the edit distance of Levenshtein) [23] and other string similarity measures based on *n*-grams [20]; semantic similarities based on synonymy relations using WordNet [5] and distances as Minkowski distance, Manhattan distance, Euclidean distance, dominance distance [24]. Other important measures, specially the ones based on schema information, are being implemented and integrated in the ontology.
  Each similarity measure uses partial information of the Ontology (e.g. the edit distance similarity measure is focused on lexical similarity between terms, while a semantic similarity measure is focused on semantic relationships between terms like synonymy, hypernymy and meronymy.) The framework aim is to include as many measures as possible and combine them to use all the information available in order to establish a mapping. The combination of them will depend of the point of view given by the user, and the rules for mapping generation (e.g. a user could only be interested in lexical similarities and a rule could combine several similarities measures through an aggregation function like the arithmetic mean or logical operators to establish a mapping).
- *The mapping ontology*. Given that our intention is that the system interacts with WebODE, its knowledge model contains the concepts: `concept`, `attribute`, `instance`, `ontology mapping`, `similarity measure`, and `point of view`. There are rules to establish mappings from similarity measures such as `if the similarity between two concepts is greater than 0.8, then a mapping between them is established`.
- *The mapping database*. It stores the generated mappings so that they can be used by different modules, for example, `the concept` *`temporal interval`* `in time ontology 1 maps with the concept` *`time interval`* `of time ontology 2`.
- *Machine Learning Module*. Some known machine learning algorithms as Naive-Bayes, K nearest neighbors, Quinlans's C4.5 algorithm for decisions trees or Agrawal's algorithm for association rules, have been also implemented and included in the machine learning module [28]. At the present moment, these algorithms are used to calculate similarities between concepts from their instances in a similar way as in [21]. For example, a Naive-Bayes classifier or a

[2] www.python.org
[3] www.oracle.com

decision tree constructed from the instances of the target ontology could be used to map a concept from the source ontology to the target ontology based on the classification of the concept instances in the first one. As happened with similarity measures, each of these algorithms take advantage of partial information and must be combined. Another interesting issue, which will be faced in the future, is how to combine these learners to learn rules for mapping generation.

The current implementation of the software uses a Model/View/Presenter paradigm[4], which is a variant of the classic Model/View/Controller paradigm[5]. The core of the system is the Model object, which implements the functionality described above by means of Command objects, which can modify its internal and observable state.

The software is written in Python (see note 2), a modern, interactive, dynamic object-oriented language with excellent integration capabilities in heterogeneous environments. The software is able to execute in any Unix, Microsoft Windows, Apple MacOS or another machine that has a Python interpreter properly installed. It also can be executed in any Java environment with the aid of the Jython interpreter[6], a pure-Java implementation of the Python programming language.

5. Conclusions and future lines

So far, partial methods and tools for particular tasks of ontology mapping had been proposed, however an integrated framework is missing. In this paper, we have presented a global framework for automatic ontology mapping generation. Being inspired by WebODE, we have established three layers: a middleware layer to create the ontology mappings, a service layer to use the mappings in different tasks (translation of expressions, ontology evolution, etc.), and an application layer for sophisticated applications (integration of heterogeneous databases, e-commerce applications, etc.).

Ontology mapping generation is a typical problem to be solved through a knowledge based system (it requires experience, manipulation of symbolic knowledge, heuristics, etc.). Therefore, the proposed framework includes an inference engine, an ontology (which plays the role of knowledge base), a machine learning module, etc. The framework is thought so that new similarity measure procedures can be dynamically attached to the system. The knowledge and the information that the system needs to manipulate can be obtained from the analysis of database of instances of ontologies, Web pages, plain texts, etc.

So far, we have implemented some basic parts of this framework, in particular, a first version of the similarity measure procedures, the mapping database and the mapping ontology. The mapping ontology will be available in the WebODE ontology server so that it can be reused by other researchers.

---

[4] http://www.object-arts.com/EducationCentre/Patterns/MVP.htm
[5] http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html
[6] http://www.jython.org

Concerning future lines, we have to develop part of the proposed framework. In fact, the problem of ontology mapping generation has proven to be very complex, and it has still a long way ahead.

Acknowledgments

We thank the Ontology Engineering Group at University of Madrid to allow us the use of WebODE, its software environment for ontology development. We are also grateful of David Manzano-Macho for his comments to the work that have made possible this paper. This work has been supported by the CEU project "Generación automática de correspondencias entre ontologías con aplicación al análisis de bases de datos de genómica y proteómica".

6. References

1. Aronson AR (1996) *MetaMap: Mapping Text to the UMLS Metathesaurus*. Technical Report of NLM's Indexing Initiative (*http://ii.nlm.nih.gov/resources/metamap.pdf*)
2. Arpírez JC, Corcho O, Fernández-López M, Gómez-Pérez A (2003) *WebODE in a nutshell*. AI Magazine, 24(3)-37-47
3. Barrasa J, Corcho O, Gómez-Pérez A (2003) *Fund Finder: A case study of database-to-ontology mapping*. In: AnHai Doan, Halevy A, Noy N (eds) Semantic Integration Workshop, Sanibel Island, Florida (*http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-82/*)
4. Berners-Lee T (1999) *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. HarperCollins Publishers, New York
5. Budanitsky A, Hirst G (2001) *Semantic Distance in WordNet: An Experimental, Application-oriented Evaluation of Five Measures*. In: Meteer M (ed) Workshop on WordNet and Other Lexical Resources, in the North American Chapter of the Association for Computational Linguistics (NAACL-2000). Pittsburgh, Pennsylvania
6. Chalupsky H (2000) *OntoMorph: a translation system for symbolic knowledge*. In: Cohn AG, Giunchiglia F, Selman B (eds) 7th International Conference on Knowledge Representation and Reasoning (KR'00). Breckenridge, Colorado. Morgan Kaufmann Publishers, San Francisco, California, pp 471–482
7. Chaudhri VK, Farquhar A, Fikes R, Karp PD, Rice JP (1998) *Open Knowledge Base Connectivity 2.0.3*. Technical Report. *http://www.ai.sri.com/~okbc/okbc-2-0-3.pdf*
8. Corcho O (2004) *A Declarative Approach to Ontology Translation with Knowledge Preservation*. PhD Thesis, Universidad Politécnica de Madrid, Madrid, Spain
9. Corcho O, Fernández-López M, Gómez-Pérez A, Lama M (2003) *An Environment for Development of Semantic Web Services*. In: Giulchiglia F, Gómez-Pérez A, Pease A, Stuchkenschmidt H, Sure Y, Willmott S (eds) IJCAI'03 Workshop Ontologies and Distributed Systems de la conferencia International Joint Conference for Artificial Intelligence, Acapulco, México, pp 13-20
10. Corcho O, Gómez-Pérez A (2001) *Solving integration problems of e-commerce standards and initiatives through ontological mappings*. In: Gómez-Pérez A, Grüninger M, Stuckenschmidt H, Uschold M (eds) IJCAI'01 Workshop on Ontologies and Information Sharing. Seattle, Washington, pp 131–140. CEUR Workshop Proceedings 47:131–140.

Amsterdam, The Netherlands
(*http:/ceur-ws.org/Vol-47/*)

11. Ehring M, Sure Y (2004) *Ontology Mapping – An Integrated Approach*. In: Bussler C, Davis J, Fensel D, Studer R (eds) 1st European Semantic Web Symposium, Heraklion, Greece, (http://www.aifb.uni-karlsruhe.de/WBS/meh/publications/ehrig04ontology_ESWS04.pdf)
12. Euzenat J, Stuckenschmidt H (2002) *The "family of languages" approach to semantic interoperability*. In: Omelayenko B, Klein M (eds) ECAI'02 Workshop on Knowledge Transformation on the Semantic Web, Lyon, France, pp 93–100
13. Giunchiglia F, Shvaiko P (2003) *Semantic Matching*. In: Giunchiglia F, Gómez-Pérez A, Pease A, Stuckensmidt H, Sure Y, Willmott S (eds) IJCAI'03 Workshop on Ontologies and Distributed Systems. Acapulco, Mexico, pp 139–146. CEUR Workshop Proceedings 139–146. Amsterdam, The Netherlands (*http:/ceur-ws.org/Vol-XX/*)
14. Gómez-Pérez A, Fernández-López M, Corcho O (2003) *Ontological Engineering: with examples from the areas of knowledge management, e-commerce and the Semantic Web*, Springer-Verlag, London
15. Gómez-Pérez A, Juristo N, Montes C, Pazos J (1997) *Ingeniería del Conocimiento: Diseño y Construcción de Sistemas Expertos*. Ceura, Madrid, Spain
16. Gruber TR (1993) *A translation approach to portable ontology specification*. Knowledge Acquisition 5(2):199–220
17. Hermenegildo M, Bueno F, Cabeza D, Carro M, García M, López P, Puebla G (2000) *The Ciao Logic Programming Environment*. In: Lloyd JW, Dahl V, Furbach U, Kerber M, Lau K, Palamidessi C, Pereira LM, Sagiv Y, Stuckey PJ (eds) International Conference on Computational Logic (CL'00). London, United Kingdom. (Lecture Notes in Computer Science LNCS 1861), Springer-Verlag, Berlin, Germany
18. Kalfoglou Y, Schorlemmer M (2003) *IF-Map: an ontology mapping method based on Information Flow theory*. Journal on Data Semantics 1:98-127
19. Kalfoglou Y, Schorlemmer M (2003) *Ontology mapping: the state of the art*. The Knowledge Engineering Review 18(1):1-31
20. Lin D (1998) *An Information-Theoretic Definition of Similarity*. In: Shavlik J (ed) 15th International Conference on Machine Learning, Madison (ICML-98), Wisconsin, pp 296–304 (*ftp://ftp.cs.umanitoba.ca/pub/lindek/papers/sim.ps.gz*)
21. Madhavan J, Philip A. Bernstein PA, Kuang Chen K, Alon Y. Halevy AY, Shenoy P (2003) *Corpus-based Schema Matching*. In Kambhampati S, Knoblock CA (eds) IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03), Acapulco, Mexico, pp 59-63
22. Maedche A, Motik B, Silva N, Volz R (2002) *MAFRA- A MApping FRAmework for Distributed Ontologies*. In: Gómez-Pérez A, Benjamins VR (eds) 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'02). Sigüenza, Spain. (Lecture Notes in Artificial Intelligence LNAI 2473) Springer-Verlag, Berlin, Germany, pp 235-250
23. Maedche A, Staab S (2002). *Measuring Similarity between Ontologies*. In Gómez-Pérez A, Benjamins VR (eds) 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'02). Sigüenza, Spain. (Lecture Notes in Artificial Intelligence LNAI 2473) Springer-Verlag, Berlin, Germany, pp 251-263.
24. Manning C, Schütze H (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts
25. Melnik S, García-Molina H, Rahm E (2002) *Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching*. In: Georgakopoulos D (ed) 18th International Conference on Data Engineering ICDE'2002. San José, California, pp 117-128

26. Mena E, Illarramendi A, Kashyap V, Sheth AP (2000) *OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies*. International Journal on Distributed and Parallel Databases 8(2):223–271
27. Miller GA (1995) *WordNet: a lexical database for English*. Communications of the ACM 38(11):39–41
28. Mitchel T (1997) *Machine learning*. McGraw-Hill, New York
29. Mitra P, Wiederhold G (2002) *Resolving Terminological Heterogeneity in Ontologies* In Euzenat J, Gómez-Pérez A, Guarino N, Stuckensmidt H (eds) ECAI'02 Workshop on Ontologies and Semantic Interoperability CEUR Workshop Proceedings 64:45–50. Amsterdam, The Netherlands (*http://ceur-ws.org/Vol-64/*)
30. Mitra P, Wiederhold G, Kersten (2000) *A graph-oriented model for articulation of ontology interdependencies*. In: Lockemann PC (eds) 7th International Conference on Extending Database Technology, EDBT 2000. Lecture Notes in Computer Science, 1777-1786
31. Noy NF, Musen MA (2000) *PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment*. In: Rosenbloom P, Kautz HA, Porter B, Dechter R, Sutton R, Mittal V (eds) 17th National Conference on Artificial Intelligence (AAAI'00). Austin, Texas, pp 450–455
32. Park JY, Gennari JH, Musen MA (1998) *Mappings for Reuse in Knowledge-Based Systems*. In: Gaines BR, Musen MA (eds) 11th International Workshop on Knowledge Acquisition, Modeling and Management (KAW'98). Banff, Canada
33. Stojanovic L, Maedche S, Motik B, Stojanovic N (2002) *User-Driven Ontology Evolution Management*. Proceedings of the 13th European Conference on Knowledge Engineering and Management, EKAW-2002, Springer, LNAI, Madrid, Spain, 2002
34. Stumme G, Maedche A (2001) *FCA-MERGE: Bottom-Up Merging of Ontologies*. Bernhard Nebel (ed) Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI 2001). Seattle, Washington. Morgan Kaufmann Publishers, San Francisco, California, pp 225–234
35. Visser PRS, Jones DM, Beer MD, Bench-Capon TJM, Díaz BM, Shave MJR (1999) *Resolving Ontological Heterogeneity in the KRAFT Project*. In: Bench-Capon T, Soda G, Tjoa AM (eds) International Conference on Database and Expert System Applications, DEXA-99. Florence, Italy. (Lecture Notes in Computer Science LNCS 1677) Springer-Verlag, Berlin, Germany, pp 668–677 (*http://www.csc.liv.ac.uk/~kraft/publications/dexa-99.ps.gz*)