AMON: A Software System for Automatic Generation of Ontology Mappings

Alfredo Sánchez-Alberca, Rául García-García, Carlos Óscar S. Sorzano, Celia Gutiérrez-Cossío,
Mónica Chagoyen, Mariano Fernández López

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 system to solve this problem is still missing. In this paper, we present AMON, a platform for automatic ontology mapping generation. First of all, we show the general structure. Then, we describe the current version of the system, including the ontology in which it is based, the similarity measures that it uses, the access to external sources, etc.

Index Terms— automatic ontology mapping generation, integrated framework, mapping evaluation, similarity measure

Alfredo Sánchez Alberca is with Universidad San Pablo CEU, Ctra. Boadilla del Monte km 5'300 28668 Boadilla del Monte, Madrid, Spain, e-mail: asalber@ceu.es.

Raúl García García is with Universidad San Pablo CEU, Ctra. Boadilla del Monte km 5'300 28668 Boadilla del Monte, Madrid, Spain, e-mail: rgg.eps@ceu.es.

Carlos Óscar Sánchez Sorzano is with Universidad San Pablo CEU, Ctra. Boadilla del Monte km 5'300 28668 Boadilla del Monte, Madrid, Spain, e-mail: coss.eps@ceu.es.

Celia Gutiérrez Cossío is with Universidad San Pablo CEU, Ctra. Boadilla del Monte km 5'300 28668 Boadilla del Monte, Madrid, Spain, e-mail: cgutiérrez.eps@ceu.es.

Mónica Chagoyen is with Universidad de Biocomputación of the Centro Nacional de Biotecnología (CSIC), Campus Universidad Autónoma sn, Madrid, Spain, e-mail: monica.chagoyen@cnb.uam.es.

Mariano Fernández López is with Universidad San Pablo CEU, Ctra. Boadilla del Monte km 5'300 28668 Boadilla del Monte, Madrid, Spain, e-mail: mfernandez.eps@ceu.es.

Alfredo Sánchez Alberca is with Universidad San Pablo CEU, Ctra. Boadilla del Monte km 5'300 28668 Boadilla del Monte, Madrid, Spain, e-mail: asalber@ceu.es.

Raúl García García is with Universidad San Pablo CEU, Ctra. Boadilla del Monte km 5'300 28668 Boadilla del Monte, Madrid, Spain, e-mail: rgg.eps@ceu.es.

Carlos Óscar Sánchez Sorzano is with Universidad San Pablo CEU, Ctra. Boadilla del Monte km 5'300 28668 Boadilla del Monte, Madrid, Spain, e-mail: coss.eps@ceu.es.

I. INTRODUCTION

AN *ontology* is a reusable and shareable vocabulary coded in such a way that 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 in the literature several ontologies that model similar knowledge in different ways [1]. 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. For example, language translation of an ontology needs to deal with two ontologies (the input and the output one) [2], or ontology evolution needs to deal with several ontologies (the different versions of the original ontology) [3]. 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 [4] [5].

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 [5]. 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

Celia Gutiérrez Cossío is with Universidad San Pablo CEU, Ctra. Boadilla del Monte km 5'300 28668 Boadilla del Monte, Madrid, Spain, e-mail: cgutiérrez.eps@ceu.es.

Mónica Chagoyen is with Universidad de Biocomputación of the Centro Nacional de Biotecnología (CSIC), Campus Universidad Autónoma sn, Madrid, Spain, e-mail: monica.chagoyen@cnb.uam.es.

Mariano Fernández López is with Universidad San Pablo CEU, Ctra. Boadilla del Monte km 5'300 28668 Boadilla del Monte, Madrid, Spain, e-mail: mfernandez.eps@ceu.es.

any 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 [6], 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 it. 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 under the supervision of a user through the module of ontology evaluation.

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, 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.

II. 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 [7], logic theories in logic notation [8], frame based models [9], etc. Concerning the conceptualization of mappings, they are represented using morphims between graphs [7] morphisms between logic theories [8], and relations between classes [9]. 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 [9]. With regard mapping evaluation, an interesting work that integrates different approaches of evaluation can be found at the Knowledge Web network [10].

Concerning the mapping generation methods, they basically operate in two phases. First, they specify how to establish the similarity between terms of different ontologies. Then, according to the similarity measures obtained in the first step, the methods specify how to generate the mappings. Any of the features of the concepts to be mapped can be used to calculate the similarity measures between the concepts of two ontologies [11]: the name (lexical similarity) [12], the natural language description (in the ontology, in thesauri, in documents, etc.) [11], the structural relations (e.g. subclass of) [13], the instances in databases [11], etc.

Material contributions. Currently, the most outstanding software systems that automatically generate ontology mappings are ONION [7], MAFRA [14], IFF [8], Ehrig and Sure's system [15], PromptDiff [16], and the Euzenat ontology alignment API [17]. 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. PromptDiff automatically generates mappings between different versions of the same ontology using heuristics. There are other systems that automatically generate mappings as an intermediate step to carry out other task, for example, Prompt [18] and FCA-Merge [19] for ontology merging. The mappings are established by extracting, from the documents, instances that belong to concepts of both ontologies. We would like to mention MetaMap [20], which maps texts in natural language into medical ontologies. With regard to Euzenat's ontology alignment API, it standardizes typical operations on mappings and supports an easy evaluation [17].

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, ontologies, 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 [21] and http://ceur-ws.org/Vol-128/ for a more thorough review on ontology mappings.
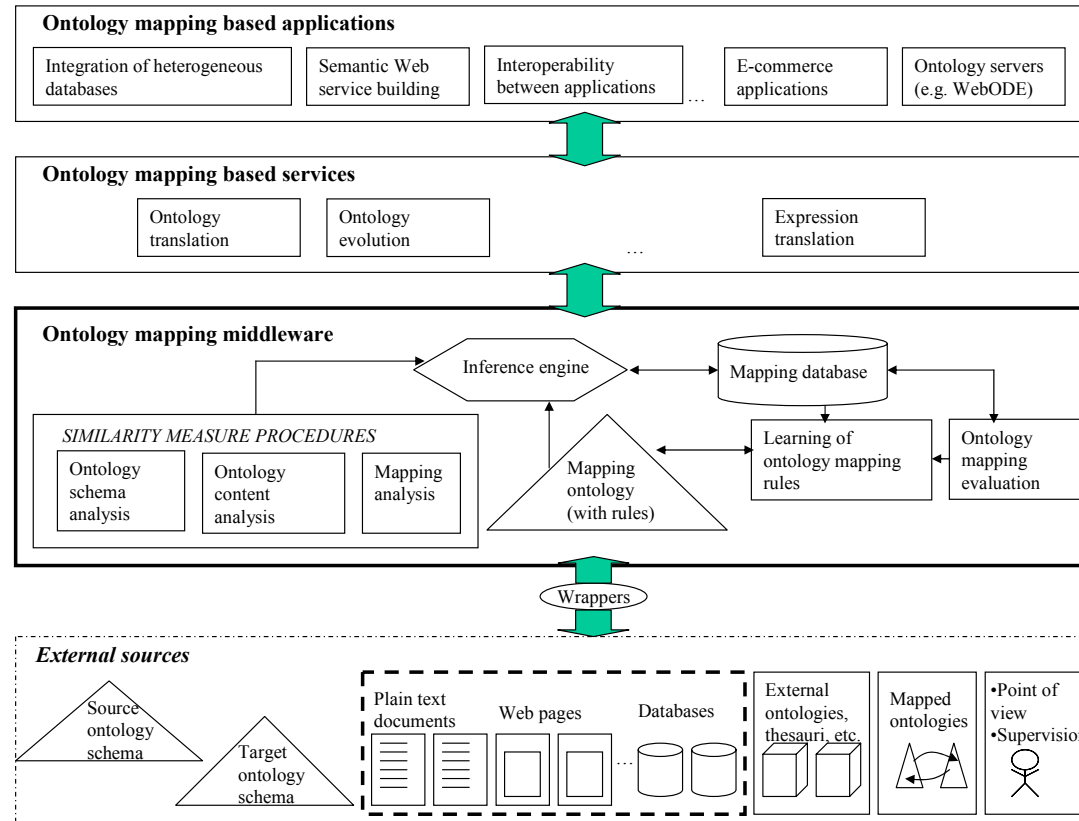
Fig. 1. Framework for automatic generation of ontology mappings

III. 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 sim₁(C₁,C₂)>0.8 and sim₂(C₁,C₂)>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.

Thus, ontology mapping is a typical problem to be solved with a knowledge based system [22]. Therefore, the structure of the core of our framework is similar to the one of classical knowledge based systems, except 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 [23]; 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 by combination of different approaches: analysis of the attributes of the concepts, analysis of their instances, analysis of the concepts taxonomy, etc.

Let's note that the arrow labeled "wrappers" is very abstract. For the case of ontology schemas it abstracts away the need for wrappers for different schema languages (e.g. RDF, OWL), whereas for the case of plain text documents/web pages it abstracts away the need for a variety of methods (e.g. ontology learning, information extraction) to identify ontological information in those sources, and for the case of supervision the need to input

(somehow) the information provided by the user, which is useful for the learning part of the framework.

Inspired by the framework proposed in [24] 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 functions*. 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:

    o   *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 [25]).

    o   *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.

    o   *The mapping analysis module*. It uses mappings previously generated by both our system or external systems.

    Let's note that some types of measures can be applied to both ontology schema and content. Thus, for example, both the schema analysis module and the content analysis module can call the same lexical similarity measure procedure.

-   *The mapping ontology*. It is a metaontology that models the knowledge on ontology mappings and

defines concepts like mapping, similarity measure, point of view of the mapping, concept, etc.

- *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.
- *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.
- *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.

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* [26] and the *OKBC Ontology* [27], both available in the Ontolingua Server (http://ontolingua.stanford.edu/). 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 [2].
- *Expression translation*. Let's assume that the ontology O_1 represents the schema of the database DB_1, and O_2, 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 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 [4] [5].
- *Semantic Web service building*. Semantic Web services can be built using problem solving methods [28]. 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 [29], 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 [24].

IV. THE CURRENT IMPLEMENTATION OF THE PROPOSED FRAMEWORK

AMON, our software system, currently implements the following elements:

- *Inference engine*. The application currently uses a Prolog library and inference engine written in the Python language. However, in future versions, we foresee the use of other logic engines (i.e. the Ciao Prolog interpreter) to execute the mapping generation rules [30].
- *Similarity functions*. Section 4.1 briefly describes de similarity measure procedures that we have implemented.
- *The mapping ontology*. Our ontology mapping addresses our current needs for expressing the mapping implemented by AMON. This ontology will be available in WebODE for its reuse and it is open to suggestions. In its current version, it represents the concepts mapping, concept (in the role of source and target concept), similarity, point of view of the similarity, etc. This last concept allows understanding the similarity between concepts according to different perspectives: their name, their natural language descriptions, their structure, combined perspectives, etc. Therefore, the point of view of a similarity integrates different ways of calculate it.
- *The mapping database*. Both the working ontologies as the mapping results have to be saved into some storage

system in order to achieve persistence. Currently the software system uses an ORM (Object-Relational Mapper) layer written in Python called SQLObject (http://sqlobject.org). The ORM layer allows the storage of Python live objects into a relational database and the system uses it to store both the source ontologies, with all of its concepts, relationships, etc, and the mappings obtained as results of the system engine; these mappings are stored as result-sets, in order to allow validation of its degree of similarity. To achieve persistence the ORM layer relies on a relational RDBMS as the storage back-end; currently we are using both MySQL (http://www.mysql.com) and SQLite (http://www.sqlite.org) as storage back-ends, buy it is likely that the system will use an Oracle Database Server in the future, when the ontologies and resultsets grow up in size.

- *Ontology mapping generation evaluation.* We have elaborated a mathematical model, and its implementation, to evaluate the mappings generated by our system. This model is further described in section 4.2.

The current implementation of the software uses a Model/View/Presenter paradigm, which is a variant of the classic Model/View/Controller paradigm (http://www.object-arts.com/EducationCentre/Patterns/MVP.htm). The core of the system is the Model object the functionality described above by means of Command objects can modify its internal and observable state.

The software is written in Python, 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 (http://www.jython.org), a pure-Java implementation of the Python programming language.

A. Similarity measure in AMON

To take advantage of all the information contained in an ontology (name concepts, natural language descriptions of concepts, concept attributes, concept taxonomy, relations, instances, axioms, etc.) we need different similarity functions and ways of combine and summarize their similarity measurements.

At present time, we have implemented and integrated several classical similarity functions, like Jaccard coefficient, Dice coefficient, cosine coefficient and correlation coefficient [12], for similarity between tuples; edit distance similarity function (based on the edit distance of Levenshtein [12] and n-grams similarity function [31] for lexical similarity between terms; WordNet based

similarity functions [32] for semantic similarity between terms; inverse document frequency cosine similarity [33] and information theoretic document similarity [34], for similarity between documents. Also, we have implemented several dissimilarities functions like Minkowski distance, Manhattan distance, Euclidean distance, Chebychev distance, dominance distance or Hamming distance [33]. Other important similarity and dissimilarity functions, specially the ones based on schema and instances information, are being implemented.

The system is capable of integrating new similarity functions as instances of different similarity function classes in the mapping ontology, just defining the source of information to apply over, and their properties.

Each similarity function uses partial information of the ontology, and the system tries to obtain as many measurements 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 (e.g. a user could only be interested in lexical similarities or taxonomical similarities) and the rules for combining (e.g. through an aggregation function like the arithmetic mean or a non-linear combination).

Currently, we are researching on a new module to create new combinations, which would generate better mappings, by means of a genetic algorithm (GA). The GA is being used to get the best similarity measurement or combination of them, for each problem. GAs solve optimization problems where traditional methods fail (called NP complete problems). Each iteration offers a better performance that optimizes the problem. The inputs of the GA are the discrepancies between the expert similarity measurement and each similarity measurement taken from AMON. The final output of the GA is the measurement itself and the constants that optimize the linear combination of the discrepancies.

B. The mapping generation evaluation

In order to validate the automatic mappings performed by AMON, we compare them to mappings generated by experts (so-called "ideal" mappings). We generate the "ideal" mappings taking mappings already generated by experts in some domain. Thus, for example, we are working with the mapping already generated between GeneOntology [35] and the Enzyme Commission (EC) classification [36]. Moreover, in the future, we think to obtain ideal mappings creating variations of the same ontology (renaming terms, changing relations, removing elements, etc.) [13].

Concerning the comparison between the "ideal" and the AMON mappings, we have elaborated a ratio of discrepancy between ontology mappings. This ratio, based on one of the presented in [10], assumes that an ontology mapping is a graph whose nodes are the concepts of the

mapped ontologies, and the edges the mapping links between concepts. Inspired in different works based in edit graph distances [37], we define:

$$discrepancyRatio = \frac{\sum_{i=1}^{nOp} \cos tOfOperation(i)}{(\sum_{i=1}^{n \operatorname{Im} I} simIM_i + \sum_{i=1}^{nN \operatorname{Im} I} simNIM_i)}$$

where each `costOfoperation`$_i$ is the cost of each operation to transform the non ideal graph in the ideal graph (see table I); `nOp` is the number of necessary operations; `simIM`$_i$ is the i-th similarity of the ideal mapping, `nIML` is the total number of similarities in the ideal mapping, `simNIM`$_i$ is the i-th similarity of the non-ideal mapping, and `nNIML` is the total number of similarities in the non ideal mapping. The denominator of the discrepancy ratio acts as a normalization factor.

TABLE I
COSTS OF OPERATIONS IN GRAPH TRANSFORMATION

Operation	Cost
Change of similarity in an edge	|newSimilarity – oldSimilarity|
Edge removal	Deleted similarity
Edge insertion	Added similarity

For example, the comparison between two test time ontologies, using just a lexical similarity, has been shown in Figure 2.

V. CONCLUSIONS AND FUTURE TRENDS

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. 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.

Concept in time ontology 1	Concept in time ontology 2	AMON result	Expert proposal	Difference
interval	time interval	0,76	0,6	0,16
temporal interval	time interval	0,8	1	0,2
temporal point	time interval	0,74	0,5	0,24
temporal point	time point	0,75	1	0,25
temporal point	time position	0,74	0,8	0,06
hour	hour	1	1	0
hour	hourfn	0,8	0,8	0
month	month	1	1	0
month	monthfn	0,83	0,8	0,03
year	year	1	1	0
year	yearfn	0,8	0,8	0
		9,22	9,3	0,94
			RESULT	0,050756

Fig. 2. Example of AMON mapping generation evaluation

So far, we have implemented some parts of this framework, in particular, a first version of the similarity measure procedures and of the mapping ontology, the mapping generation evaluation, and the mapping database.

As work in progress, we are experimenting with genetic algorithms to combine similarity measures. Moreover, we want to generate different variations of the same ontology to have a new way to evaluate our system.

REFERENCES

[1] A. Gómez-Pérez, M. Fernández-López, Corcho O. *Ontological Engineering: with examples from the areas of knowledge management, e-commerce and the Semantic Web*, Springer-Verlag, London, 2003

[2] O. Corcho *A Declarative Approach to Ontology Translation with Knowledge Preservation*. PhD Thesis, Universidad Politécnica de Madrid, Madrid, Spain, 2004

[3] L. Stojanovic, S. Maedche, B. Motik, N. Stojanovic "User-Driven Ontology Evolution Management". *Proceedings of the 13th European Conference on Knowledge Engineering and Management, EKAW-2002*, Springer, LNAI, Madrid, Spain, 2002

[4] E. Mena, A. Illarramendi, V. Kashyap, A.P. Sheth "OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies".

International Journal on Distributed and Parallel Databases vol. 8, no. 2, 2000, pp. 223–271

[5] P.R.S. Visser, D.M. Jones, M.D. Beer, T.J.M. Bench-Capon, B.M. Díaz, M.J.R. Shave "Resolving Ontological Heterogeneity in the KRAFT Project". In: T. Bench-Capon, G. Soda, A.M. Tjoa (eds) *International Conference on Database and Expert System Applications, DEXA-99*. Florence, Italy. (Lecture Notes in Computer Science LNCS 1677) Springer-Verlag, Berlin, Germany, 1999, pp 668–677 (http://www.csc.liv.ac.uk/~kraft/publications/dexa-99.ps.gz)

[6] T. Berners-Lee *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. HarperCollins Publishers, New York, 1999

[7] P. Mitra, G. Wiederhold, Kersten "A graph-oriented model for articulation of ontology interdependencies". In: P.C. Lockemann (eds) *7th International Conference on Extending Database Technology, EDBT 2000*. Lecture Notes in Computer Science, 2000, pp. 1777-1786

[8] Y. Kalfoglou, M. Schorlemmer (2003) "IF-Map: an ontology mapping method based on Information Flow theory". *Journal on Data Semantics*, vol. 1, 2003, pp. 98-127

[9] J.Y. Park, J.H. Gennari, M.A. Musen "Mappings for Reuse in Knowledge-Based Systems". In: B.R. Gaines, M.A. Musen (eds) *11th International Workshop on Knowledge Acquisition, Modeling and Management (KAW'98)*. Banff, Canada, 1998

[10] J. Euzenat, R. García-Castro, M. Ehring "Specification of a benchmarking methodology for alignment techniques". *KnowledgeWeb deliverable D2.2.2*, 2005. http://www.aifb.uni-karlsruhe.de/WBS/meh/publications/kweb-222.pdf

[11] P. Mitra, G. Wiederhold (2002) "Resolving Terminological Heterogeneity in Ontologies". In: J. Euzenat, A. Gómez-Pérez, N. Guarino, H. Stuckensmidt (eds) *ECAI'02 Workshop on Ontologies and Semantic Interoperability*. CEUR Workshop Proceedings vol. 64, pp. 45–50. Amsterdam, The Netherlands, 2002 (http://ceur-ws.org/Vol-64/)

[12] A. Maedche, S. Staab. "Measuring Similarity between Ontologies". In A. Gómez-Pérez, V.R. Benjamins (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, 2002, pp 251-263.

[13] J. Euzenat "Introduction to the EON Ontology alignment context". In: Y. Sure, O. Corcho, J. Euzenat, T. Hughes, L. Martin (eds) *Proceedings of the 3rd International Workshop on Evaluation of Ontology based Tools (EON 2004)*. Hiroshima, Japan, 2004, pp 47-50

[14] A. Maedche, B. Motik, N. Silva, R. Volz. "MAFRA- A MApping FRAmework for Distributed Ontologies". In: A. Gómez-Pérez, V.R. Benjamins (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, 2002, pp 235-250

[15] M. Ehring, Y. Sure "Ontology Mapping – An Integrated Approach". In: C. Bussler, J. Davis, D. Fensel, R. Studer (eds) *1st European Semantic Web Symposium*, Heraklion, Greece, 2004 (http://www.aifb.uni-karlsruhe.de/WBS/meh/publications/ehrig04ontology_ESWS04.pdf)

[16] N.F. Noy, M.A. Musen "Ontology versioning in an ontology-management framework". *IEEE Intelligent Systems* vol. 19, no. 4, 2004, pp. 6-13

[17] J. Euzenat "An API for ontology alignment". In: F. van Harmelen, S. McIlraith, D. Plexousakis (eds) *3rd International Semantic Web Conference (ISWC 2004)*. Hiroshima, Japan. (Lecture Notes in Computer Science LNCS 3298), Springer-Verlag, Berlin, Germany, 2004, pp 698-712.

[18] N.F. Noy, M.A. Musen "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment". In: P. Rosenbloom, H.A. Kautz, B. Porter, R. Dechter, R. Sutton, V. Mittal (eds) *17th National Conference on Artificial Intelligence (AAAI'00)*. Austin, Texas, 2000, pp 450–455

[19] G. Stumme, A. Maedche "FCA-MERGE: Bottom-Up Merging of Ontologies". B. Nebel (ed) *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI 2001)*. Seattle, Washington. Morgan Kaufmann Publishers, San Francisco, California, 2001, pp 225–234

[20] A.R. Aronson *MetaMap: Mapping Text to the UMLS Metathesaurus*. Technical Report of NLM's Indexing Initiative, 1996 (http://ii.nlm.nih.gov/resources/metamap.pdf)

[21] Y. Kalfoglou, M. Schorlemmer M "Ontology mapping: the state of the art". *The Knowledge Engineering Review*, vol. 18, no. 1, 2003, pp 1-31

[22] A. Gómez-Pérez, N. Juristo, C. Montes, J. Pazos *Ingeniería del Conocimiento: Diseño y Construcción de Sistemas Expertos*. Ceura, Madrid, Spain, 1997

[23] G.A. Miller "WordNet: a lexical database for English". *Communications of the ACM*, vol. 38, no. 11, 1995, pp. 39–41

[24] J.C. Arpírez, O. Corcho, M. Fernández-López, A. Gómez-Pérez "WebODE in a nutshell". *AI Magazine*, vol. 24, no. 3, 2003, pp 37-47

[25] J. Barrasa, O. Corcho, A. Gómez-Pérez "Fund Finder: A case study of database-to-ontology mapping". In: D. AnHai, A. Halevy, N. Noy (eds) *Semantic Integration Workshop*, Sanibel Island, Florida, 2003 (http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-82/)

[26] T.R. Gruber "A translation approach to portable ontology specification". *Knowledge Acquisition*, vol. 5, no. 2, 1993, pp. 199–220

[27] V.K. Chaudhri, A. Farquhar, R. Fikes, P.D. Karp, J.P. Rice *Open Knowledge Base Connectivity 2.0.3*. Technical Report, 1998. http://www.ai.sri.com/~okbc/okbc-2-0-3.pdf

[28] O. Corcho, M. Fernández-López, A. Gómez-Pérez, M. Lama M "An Environment for Development of Semantic Web Services". In: F. Giulchiglia, A. Gómez-Pérez, A. Pease, H. Stuchkenschmidt, Y. Sure, S. Willmott (eds) *IJCAI'03 Workshop Ontologies and Distributed Systems de la conferencia International Joint Conference for Artificial Intelligence*, Acapulco, México, 2003, pp 13-20

[29] O. Corcho, A. Gómez-Pérez "Solving integration problems of e-commerce standards and initiatives through ontological mappings". In: A. Gómez-Pérez, M. Grüninger, H. Stuckenschmidt, M. Uschold (eds) *IJCAI'01 Workshop on Ontologies and Information Sharing*. Seattle, Washington, pp 131–140. CEUR Workshop Proceedings vol. 47, 2001, pp. 131–140. Amsterdam, The Netherlands (http://ceur-ws.org/Vol-47/)

[30] M. Hermenegildo, F. Bueno, D. Cabeza, M. Carro, M. García, P. López, G. Puebla "The Ciao Logic Programming Environment". In: J.W. Lloyd, V. Dahl, U. Furbach, M. Kerber, K. Lau, C. Palamidessi, L.M. Pereira, Y. Sagiv, P.J. Stuckey (eds) *International Conference on Computational Logic (CL'00)*. London, United Kingdom. (Lecture Notes in Computer Science LNCS 1861), Springer-Verlag, Berlin, Germany, 2000

[31] D. Lin "An Information-Theoretic Definition of Similarity". In: J. Shavlik (ed) *15th International Conference on Machine Learning, Madison (ICML-98)*, Wisconsin, 1998, pp 296–304 (ftp://ftp.cs.umanitoba.ca/pub/lindek/papers/sim.ps.gz)

[32] A. Budanitsky, G. Hirst "Semantic Distance in WordNet: An Experimental, Application-oriented Evaluation of Five Measures". In: M. Meteer (ed) *Workshop on WordNet and Other Lexical Resources, in the North American Chapter of the Association for Computational Linguistics (NAACL-2000)*. Pittsburgh, Pennsylvania, 2001

[33] C. Manning, H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999

[34] A. Javed A "Aslam and Meredith Frost. An information-theoretic measure for document similarity". In C. Clark, C. Cormak (eds) *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval SIGIR'03.* Toronto, Canada, 2003, pp 449-450

[35] The Gene Ontology Consortium (2000) *Gene Ontology: tool for the unification of biology.* Nature Genetics 25:25-29.

[36] E.C. Webb *Enzyme Nomenclature.* Academic Press, San Diego, California, 1992

[37] H. Bunke "Error-correcting graph matching: On the influence of the underlying cost function". *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, 1999, pp. 917–922