Context-Sensitive Referencing for Ontology Mapping Disambiguation

Heiko Paulheim, Michael Rebstock, and Janina Fengel*

Hochschule Darmstadt, University of Applied Sciences
{lastname}@fbw.h-da.de

Abstract. Ontologies can be used for e-business integration, for example by describing existing e-business standards as ontologies. If cooperating parties use different ontologies, ontology mappings are needed, which can be ambiguous, thus making ontology mapping disambiguation necessary. Different disambiguation strategies exist, such as community-driven or context-sensitive referencing of ontologies, where the latter is what we developed in our project. In this paper, we show that community-driven referencing can be realized using a context-sensitive referencing service in a way that the user administration is transparent to the referencing system.

Keywords: Semantic Synchronization, Ontology Mapping, Ontological Engineering, Context-Sensitivity, Communities

1 E-Business Integration with Ontologies

Standards play an important role in electronic business. Unfortunately, there are different and competing standards for describing products, processes, documents, and the like. To allow interoperability, mechanisms that allow parallel usage of elements from different e-business standards in the same process are needed.

Nowadays, such mechanisms mainly either exist in the users' minds, or in fixed translation tables that require a major project effort and do not allow dynamic change. Furthermore, semantic synchronizations carried out manually are not persistent. With the framework presented in this paper, we provide a general architecture for the implementation of an evolutionary semantic synchronization service that can be integrated into different e-business systems to support users with semantic knowledge.

Following [1], we look at e-business standards as ontologies, thus, the elements to be synchronized are the ontologies' concepts and properties. This enables us to use methods and tools from the field of ontological engineering. Some existing e-business standards, like UN/SPSC [2] and eCl@ass [3], have already been transferred into ontology languages. Furthermore, a lot of research has been conducted in the past years on technologies for processing ontologies, so there are a couple of components ready to use, including ontology representation, visualization, mapping, and reasoning. We have implemented a framework on

* The authors are supported by the German Federal Ministry of Education and Research under grant 1716X04, project ORBI.

top of JENA2 [4] and Java that allows connecting such components to form a coherent semantic referencing service [5], as well as reusing techniques from information retrieval (IR).

The service allows users to find references between ontologies. References may either be created manually or established automatically by a mapping tool. However, as stated in [6], more than one reference can exist for the same element, caused by different modelling approaches and granularities of the individual standards, even more so if proprietary or in-house standards are used. Therefore, reference disambiguation strategies are needed, which filter appropriate results and/or sort results by relevance. The framework developed in our project evaluates context information to provide reference disambiguation.

The rest of this paper is structured as follows. Section 2 describes the basics on ontologies, references, and context. Section 3 explains two approaches for reference disambiguation: community-based and context-sensitive referencing. Section 4 shows how community-based referencing can be realized using context-sensitive referencing. Section 5 provides an overview on related work, and section 6 closes with a discussion of our results.

2 Ontologies, Semantic References, and Context

Ontologies are structured, machine-readable representations of knowledge. There are many different definitions of what an ontology actually is (for a comprehensive overview see [1]), however, we will look at ontologies as a collection of definitions of elements and their relations. Ontologies can be represented in different languages, the most dominant are RDF Schema [7], and the various dialects of OWL [8]. Ontologies are considered as a means for e-business integration [9], however, if two or more cooperating parties use different ontologies, further steps have to be taken to allow seamless interoperability.

Therefore, ontology matching solutions are needed, which produce mappings from elements in one ontology to elements in another. There are two main categories of ontology matching algorithms [10]. One are element-based approaches, which try to match single elements of an ontology, either using only the information given in the ontology itself (e.g., by measuring string distance using the edit distance), or by using external information, e.g. upper-level ontologies, such as WordNet [11]. The second are structure-level approaches, which do not only analyze elements isolated from each other, but also their relations and patterns they form in graphs. An overview and more detailed analysis of matching approaches can be found in [10] and [12]. Some approaches, like [13], combine the weighted results of several matching solutions in order to obtain mappings of higher quality.

Ontology matching tools provide references. In extension of [14], references can be described as a five-dimensional vector of the form

$$reference := \langle entity1, entity2, type, confidence, acceptance \rangle \,. \tag{1}$$

The first two entries *entity*1 and *entity*2 are URIs of the elements from both ontologies to be referenced, *type* describes the kind of relation (like “equal”, "subclass of", etc.), $confidence$ describes the degree of probability of the relation, and *acceptance* expresses the users' rating of that reference. For example, the reference

$$r_1 = \langle StandardA\#X, StandardB\#Y, equal, 0.87, 0.95\rangle \qquad (2)$$

is read as "Element X in StandardA and element Y in StandardB are equal with a probability of 87%, and 95% of all users agreed on that statement". The acceptance value is calculated from the users' ratings.

In order to disambiguate such semantic references, we have developed an approach which uses context information. There has been a lot of research on context in the fields of machine translation and IR, yielding several ways of describing context. In machine translation, shallow and deep approaches [15], bag of words and relational approaches [16] are distinguished to solve the problem of word sense disambiguation. In IR, context data can be represented in different forms, from simple binary vectors to highly complex graphs, as proposed by [17]. An introduction to context queries in IR can be found in [18].

3 Approaches for Mapping Disambiguation

3.1 Community-Based Referencing

The idea of context mapping disambiguation by using communities has first been developed by Anna V. Zhdanova and Pavel Shvaiko in [19]. The general problem of community-based referencing can be formally defined as follows:

Definition 1. *Given a user being member in a non-empty set of communities S_U, find those references for an element x from a set of ontologies O_1 to a set of ontologies O_2 that have been created by a user being member in a non-empty set of communities S_C under the condition that $S_U \cap S_C$ is not empty.*

That means that a user issuing a query for semantic references on an element is presented all references for that element created by users with whom he has at least one community in common (note that we are considering the creators of *ontology references*, not of the ontologies themselves). The user's login and community data are directly processed by the referencing system.

Although the authors of [19] primarily focused on mapping reuse, this community-driven approach can also be seen as an ontology mapping disambiguation strategy: different semantic references caused by ambiguous use of elements in different communities are filtered and thereby disambiguated. We will call a semantic referencing service that allows disambiguation by using context information a *community-based semantic referencing service.*

Figure 1 demonstrates the idea of community-driven mapping disambiguation. There are two references for the element "switch" from a rather coarse-grained proprietary standard P to the more fine-grained standard eCl@ss [20], each having its right to exist in a given context. User 1 is a network administrator using standard P for ordering an ethernet LAN switch. Since the supplier

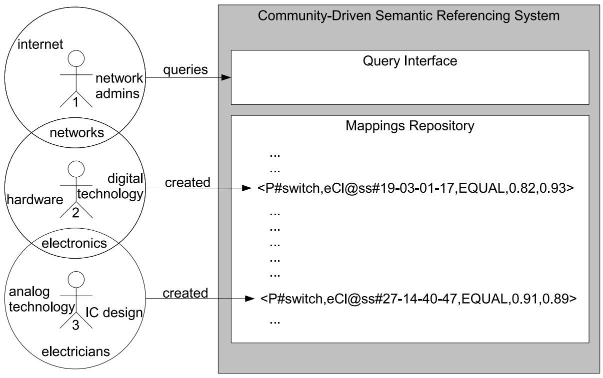

Fig. 1. Community-driven mapping disambiguation

uses eCl@ss, user 1 queries the semantic referencing system for references for the element "switch". The system returns the reference to "19-03-01-17" (which is the eCl@ss code for "network switch") created by user 2, since both users are in the "networks"-community, but does not return the reference to "27-14-40-47" (which is the eCl@ss code for "toggle switch") created by user 3, since users 1 and 3 do not share any communities. The list of references that exist for the element "switch" is thus filtered and thereby disambiguated.

3.2 Context-Sensitive Referencing

A different approach for disambiguating semantic references is the evaluation of the context of the term to be referenced. The general problem of context-sensitive referencing can be defined as follows:

Definition 2. *Given some context information $C(x)$, find the references for an element x from a set of ontologies O_1 to a set of ontologies O_2, with an acceptance value $acc_{C(x)}$ (which is the higher the more appropriate the reference is in this context), calculated dynamically for that context information and exceeding a minimum acceptance threshold acc_{min}.*

Such an acceptance value $acc_{C(x)}$ can be obtained in different ways. Since one of the design aims of our system was to minimize the need for manual preparatory work, we decided to calculate $acc_{C(x)}$ based on user ratings. Each user can rate (in the easiest case: accept or deny) a reference in his or her context, and the ratings are stored in the system. Each time a user requests a reference for an element in a context, the acceptance value is calculated using the distance-weighted k-nearest-neighbor rule [21], with the difference between the similarity of the request's context $C_Q(x)$ and the rating's context $C_R(x)$ as distances, given any similarity function sim. In other words, $acc_{C(X)}(Ref)$ is calculated as

$$acc_{C_Q(X)}(Ref) = \begin{cases} \sum_{R \in Ratings(Ref)} \frac{sim(C_Q(x), C_R(x))}{sum_{sim}} \cdot acc(R) & sum_{sim} > 0 \\ acc_{def} & sum_{sim} = 0 \end{cases} \quad (3)$$

where sum_{sim} is calculated as

$$sum_{sim} := \sum_{R \in Ratings} sim(C_R(x), C_Q(x)). \quad (4)$$

and acc_{def} is a configurable parameter which serves as a default acceptance if no ratings exist or if none of the ratings is at least minimally similar to the query's context. In the latter case, it is also possible to use the unweighted median of all ratings.

We will call a semantic referencing service which uses context-sensitive reference disambiguation a *context-sensitive semantic referencing service.*

As already stated in section 2, there are different ways to describe context. Since different client applications can have different strategies of gathering context information, using more specific context information (as in deep and relational approaches) narrows the variety of possible client applications. Therefore, we decided for a relational approach which uses a weighting factor for each context term, where the context terms are simple strings. Therefore, the context of an element x is defined as a set of context terms $C(x)$, and a normalized weighting function ω, defined as

$$\omega_{C(X)} : C(X) \rightarrow [0,1] \text{ with } \max_{y \in C(X)} \omega_{C(X)} = 1. \quad (5)$$

That function can also be interpreted as a reverse of a distance function: the higher a context term's weight, the closer it is to the term in question.

Since many context similarity measures are defined for vectors, with the context terms used as dimensions and the weights as values, the weighting function can also be regarded as a weighting vector $w_{C(X)}$ with

$$w_{i,C(X)} := \omega_{C(X)}(t_i), t_i \in C(X), 1 \leq i \leq |C(X)|. \quad (6)$$

With those definitions, an acceptance value can be calculated for each reference, determining that reference's appropriateness in the query's context. Thereby, semantic references can be disambiguated. Details on context-sensitive reference disambiguation can be found in [22].

4 Community information as a special kind of context

4.1 Using communities as context information

A query for references in a community-driven scenario, as stated in definition 1, can be identified by a query term X and by a set S_U of community identifiers, where $S_U \subseteq S$, and S represents the set of all communities. A query in a context-sensitive scenario, as stated in definition 1, is identified by a query term X, a context set $C(X)$ (containing context terms), and a weighting function $\omega_{C(X)}$ as defined in (5).

Since, according to definition 1, the result set would be empty if the user was not a member of any community, we assume that each user issuing a query is a member of at least one community.

In order to transform a community-driven query to a context-sensitive one, we treat the community identifiers as simple strings and define:

$$C(X) := S \text{ and } \omega_{C(X)}(t) := \begin{cases} 1 & \forall\, t \in S_U \\ 0 & \forall\, t \in S - S_U \end{cases} \tag{7}$$

We are now going to show that our context-sensitive reference disambiguation approach answers context-based queries as defined above such that the following requirements are fulfilled:

Requirement 1: All references created by users that share at least one community with the user issuing the query are returned.

Requirement 2: No references created by users that do not share any community with the user issuing the query are returned.

To this end, we use the cosine similarity [18] as a similarity measure, and a default acceptance $acc_{def} = 0$. Furthermore, we assume that for each reference that one and only one rating exists, whose context is the community information of the reference's creator as defined above and whose acceptance value is 1. We will elaborate on how to assure this assumption in the next section.

Let $w_{C_Q(X)}$ be the query's weighting vector and $w_{C_Q(X)}$ be the rating's vector (containing the community information of the reference's creator), according to (6).

The cosine similarity is defined as

$$sim_{cos}\left(w_{C_Q(X)}, w_{C_R(X)}\right) := \frac{w_{C_Q(X)} \bullet w_{C_R(X)}}{\left\|w_{C_Q(X)}\right\| \left\|w_{C_R(X)}\right\|}. \tag{8}$$

Since each user is a member of at least one community, at least one element in both w_Q and w_R has a value of 1, thus, the denominator never equals 0. Furthermore, $w_{C_Q(X)} \bullet w_{C_R(X)}$ is greater than zero if and only if both vectors contain a non-zero element in the same position, e.g. if both users have at least one community in common, and zero otherwise. Thus, (3) reduces to

$$acc_{C_Q(X)}(Ref) = \begin{cases} > 0 & \text{if } sim\left(w_{C_Q(X)}, w_{C_R(X)}\right) > 0 \\ 0 & \text{if } sim\left(w_{C_Q(X)}, w_{C_R(X)}\right) = 0 \end{cases} \tag{9}$$

Thus, if all semantic references are filtered with a threshold of $acc_{min} = 0$, and only references with an acceptance value $acc_{C_Q(X)}(Ref) > 0$ are returned, the two requirements stated above are fulfilled. That shows that our system can provide community-driven reference disambiguation, put down to context-sensitive referencing.

4.2 Providing community-based reference disambiguation by a context-sensitive referencing service

Our original context-sensitive referencing service provides three main functions:

- Create a new reference,
- get a list of references in a given context,
- and rate a reference in a given context.

In order to assure that only one rating exists for each reference, as proposed in the section above, those functions are encapsulated to form a community-based referencing service as follows:

- Each time a user creates a reference using the community-driven referencing service, the reference is automatically rated with an acceptance value of 1 in the context derived from the user's community information.
- The request for a list of references remains the same.

With this approach, we have created a community-driven semantic referencing service by encapsulating our context-sensitive semantic referencing service, where the latter remains unchanged. The referencing system only processes context data, thus abstracting away from user and community administration. In principal, the algorithm is generic enough to solve other context-based disambiguation tasks as well.

5 Related Work

In the area of ontological engineering, much research work has already been conducted on ontology matching and ontology reasoning. Ontology matching deals with finding similarities between ontologies, often in order to merge them [10]. Ontology reasoning tries to derive new knowledge from knowledge already present in an ontology. There are also approaches trying to improve ontology mappings by means of ontology reasoning [23], while others propose an ontology mapping language capable of mapping heterogeneous information, like concepts to relations [24].

Some research projects deal with providing semantic references between e-business standards to allow semantic integration. Besides the already mentioned community-based approach developed by Zhdanova and Shvaiko [19], some other projects exist. [25] combine agents and ontology mapping to allow automatic e-business transactions. Some approaches try to collect references under the umbrella of one global ontology, like WordNet [26]. [27] propose a hierarchy of ontologies connected by mappings. Zimmermann and Euzenat haven shown in [28] that a context-sensitive approach is not possible for ontology alignment. However, it is a feasible approach for disambiguating semantic references. Other works, like [29], use ontologies, for example, to disambiguate items like person names in unstructured text by searching context terms in ontologies, unlike our approach, where context terms can be arbitrary strings that need not exist in any ontology.

The problem of context-sensitive referencing can be regarded as a special information retrieval problem. Extensive research has been conducted in this area. The present approaches stretch from using simple context term vectors [18] to describe context in rich semantic structures like RDF graphs [17]. There are also community-based information retrieval approaches, like [30], which uses the visualization of different perspectives in distinct communities for sharing information across community borders.

While our system is based on creating a collection of references, other approaches try on-the-fly mapping of ontologies [31], which is a reasonable approach when, like in the case of very large ontologies, the collection of mappings tends to become rather extensive. There are also works on matching blocks of partioned ontologies [32], which could be a possible approach to deal with the problem of large ontologies.

6 Discussion

In this paper, we have shown that a context-sensitive semantic referencing service, combined with user's ratings, can also be used for providing community-based semantic referencing. Both are feasible approaches for ontology mapping disambiguation, each having their advantages and drawbacks:

- Both approaches provide mechanisms to create a growing knowledge base of semantic references.
- Community-based referencing needs the additional implementation of user and community administration, while context-sensitive referencing also works from scratch (our implementation of the service also works with empty context information).
- On the other hand, community-based referencing is an appropriate approach to ensure that references remain private in a community and users from other communities will never come to see those references.
- The rating mechanism underlying our context-sensitive approach can also be made transparent to the user by observing the user's behavior: if a user works with a reference, it receives a positive rating, if s/he decides not to work with a proposed reference, it receives a negative rating.
- Both approaches have to cope with erroneous user's entries. Community-based referencing only has to deal with wrong references. Context-sensitive referencing also has to handle wrong ratings, which can mislead the system to calculate a wrong acceptance value and thus present a reference not appropriate in a context as being highly appropriate, and vice versa. However, the ratio of correct ratings to incorrect ones is high enough, the weight of wrong ratings decreases, and it is likely that many negative ratings will make a wrong reference fall below the lower acceptance threshold and thus make it "disappear" from the list of results displayed for the user.
- Since the usage context of a term in general can be expected to be similar within a community and different between distinct communities, context information can be looked at as implicit community information, and vice versa.

The approach presented in this paper does not yet allow using context-sensitive and community-driven semantic referencing in parallel (e.g. to further disambiguate different references used in a community). However, if this can be achieved by allowing two sets of context (the community information and the actual context information), calculating an acceptance value for each context and applying filters to each of the calculated acceptance values. Such an approach would also make the use of further types of context information possible, like documents, bookmarks, the user's role in a company, or previous projects the user has worked on, as proposed by [17].

References

1. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering. Springer, London et al. (2004)
2. Klein, M.: DAML+OIL and RDF Schema Representation of UNSPSC. http://www.cs.vu.nl/mcaklein/unspsc, October 31, 2006
3. Hepp, M.: Ontologizing eCl@ss. http://www.heppnetz.de/eclassowl/, October 31, 2006
4. Jena: Jena - A Semantic Web Framework for Java. http://jena.sourcefourge.net/, October 25, 2006
5. Paulheim, H., Rebstock, M., Fengel, J.: Dynamic Semantic Synchronization in E-Business. In: Proceedings of the 1st European Semantic Technology Conference, May 31st - June 1st, Vienna, Austria (CD-ROM). (2007)
6. Schulten, E., Akkermans, H., Guarino, N., Botquin, G., Lopes, N., Dörr, M., Sadeh, N.: The E-Commerce Product Classification Challenge. IEEE Intelligent Systems **16**(4) (2001) 86–c3
7. RDF Core Working Group: RDF Vocabulary Description Language 1.0: RDF Schema (2004) http://www.w3.org/TR/rdf-schema/.
8. Web Ontology Working Group: Web Ontology Language (OWL) (2004) http://www.w3.org/2004/OWL/.
9. Uschold, M., Grüninger, M.: Ontologies: principles, methods, and applications. Knowledge Engineering Review **11**(2) (1996) 93–155
10. Shvaiko, P., Euzenat, J.: A Survey of Schema-based Matching Approaches. In: Journal on Data Semantics. (2005)
11. Fellbaum, C., ed.: WordNet: An Electronic Lexical Database. The MIT Press, Cambridge, MA (1998)
12. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. The Knowledge Engineering Review **18**(1) (2003) 1–31
13. Kalfoglou, Y., Hu, B., Reynolds, D., Shadbolt, N.: CROSI project, final report. Technical Report 11717, School of Electronics and Computer Science, University of Southampton, Southampton, UK (2005)
14. Euzenat, J.: Evaluating ontology alginment methods, extended abstract. In: Dagstuhl Seminar Proceedings 04391, Semantic Interoperability and Integration, http://drops.dagstuhl.de/opus/volltexte/2005/36, June 06, 2005. (2005)
15. Uszkoreit, H.: Shallow Language Processing, Deep Language Processing and Domain Ontologies. In: Natural Language Processing and Knowledge Engineering. (2005)
16. Ide, N., Véronis, J.: Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. In: Computational Linguistics 24 (1). (1998) 1–40

17. Henrich, A., Morgenroth, K.: Kontextbasiertes Information Retrieval zur Unterstützung von Entwicklungsprozessen. In: eOrganiztion: Service-, Prozess-, Market-Engineering. Volume 2., Universitätsverlag Karlsruhe (2007) 575–592
18. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press, New York (1999)
19. Zhdanova, A.V., Shvaiko, P.: Community-Driven Ontology Matching. In Sure, Y., Domingue, J., eds.: The Semantic Web: Research and Applications ESWC 2006, LNCS 4011, Berlin Heidelberg, Springer (2006) 34–49
20. ECl@ss: Internationaler Standard zur Klassifizierung u. Beschreibung von Produkten u. Dienstleistungen. http://www.eclass.de/, December 12, 2006
21. Dudani, S.A.: The distance-weighted k-nearest-neighbor rule. In: IEEE Transactions on Systems, Man and Cybernetics Vol. 6. (1976) 325–327
22. Rebstock, M., Fengel, J., Paulheim, H.: Context-Sensitive Semantic Synchronization in Electronic Negotiations. In: Proceedings of the Group Decision and Negotiation Meeting, Mt. Tremblant (Montreal), May 14-17, 2007. Volume 2. (2007) 253–264
23. Huang, J., Gutiérrez, R.L.Z., García, B.M., Huhns, M.N.: A Schema-Based Approach Combined with Inter-Ontology Reasoning to Construct Consensus Ontologies. In: Proc. AAAI Workshop on Contexts and Ontologies: Theory, Practice and Applications. (2005) 80–87
24. Ghidini, C., Serafini, L.: Reconciling Concepts and Relations in Heterogeneous Ontologies. In Sure, Y., Domingue, J., eds.: The Semantic Web: Research and Applications ESWC 2006, LNCS 4011, Berlin Heidelberg, Springer (2006) 50–64
25. Malucelli, A., Palzer, D., Oliveira, E.: Combining Ontologies and Agents to help in Solving the Heterogeneity Problem in E-Commerce-Negotiations. In: Proceedings of the 2005 International Workshop on Data Engineernig Issues in E-Commerce. (2005)
26. Kim, W., Choi, D.W., Park, S.: Product Information Meta-search Framework for Electronic Commerce. In Gómez-Pérez, A., Euzenat, J., eds.: ESWC 2005, LNCS 3532, Berlin Heidelberg, Springer (2005) 408–422
27. Corcho, O., Gómez-Pérez, A.: Solving Integration Problems of E-commerce Standards and Initiatives through Ontological Mappings. In: Proceedings of the Workshop on E-Business and intelligent Web at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI2001), Seattle, USA (2001)
28. Zimmermann, A., Euzenat, J.: Three Semantics for Distributed Systems and Their Relations with Alignment Compositions. In: Proc. of 5th International Semantic Web Conference, Athens, GA, USA, November 5-9, 2006, LNCS 4273. 16–29
29. Hassell, J., Aleman-Meza, B., Arpinar, I.B.: Ontology-Driven Automatic Entity Disambiguation in Unstructured Text. In: Proc. of 5th International Semantic Web Conference, Athens, GA, USA, November 5-9, 2006, LNCS 4273. 44–57
30. Novak, J.: Multiperspektivische Wissensvisualisierung für Wissensaustausch in heterogenen Netzen. In: eOrganiztion: Service-, Prozess-, Market-Engineering. Volume 2., Universitätsverlag Karlsruhe (2007) 611–628
31. Lopez, V., Sabou, M., Motta, E.: PowerMap: Mapping the Real Semantic Web on the Fly. In: Proc. of 5th International Semantic Web Conference, Athens, GA, USA, November 5-9, 2006, LNCS 4273. 414–427
32. Hu, W., Qu, Y.: Block Matching for Ontologies. In: Proc. of 5th International Semantic Web Conference, Athens, GA, USA, November 5-9, 2006, LNCS 4273. 300–313