International Journal Multimedia and Image Processing (IJMIP), Volume 3, Issues 3/4, September/December 2013

Comparative Analysisof Ontology Alignment Methodology

Said Rabah Azzam
School of Engineering
University of Portsmouth, United Kingdom

Shikun Zhou
School of Engineering
University of Portsmouth, United Kingdom

Abstract

Context based searches is always the area which admires by different researchers. Ontology hence, is the way to develop such a semantic web. However, alignment is the way to properly align Ontologies to actually develop the semantic environment. Thus interoperability between different classes can be maintained in order to optimize the context based searches. There exist different methods to map the given Ontologies to make their classes or attributes unique. However, every of them have some drawbacks. We will critically analyze different published methods according to their common attributes.

1. Introduction

There exist different algorithms that can align the Ontology. The term "mapping" is referred when doing an alignment. Each has its own advantages and drawbacks. In this paper, we will define those methods, present work of other authors and critically analyze each method as per their common attribute in order to make an effective way to align the Ontology. This will enable other authors to analyze from the proposed dimension and subsequently refine to make it more effective.

2. Background

Ontology as a language to develop such a platform that provides properties for semantic web. In the past, there has been conducted extensive research of what path will enable the semantic web to share the information as well as reutilizing the same and to make context-based searches. With different proposed solutions, the latest ones is the constructive language OWL Ontology that enables developing a semantic environment. With OWL-DL Ontology, one can develop comprehensive scenarios to capture entire semantics of the system.

Ontology will therefore is a technique to develop a functional semantic web. The goal is to come up with the solution that machines can understand in order to automate the possible systems, developing an environment for data integrity and harmony by having intra-systems integrated solutions, and auto-decision makers to enable addressing issues. Thus, Ontology is a language to achieve the semantic web.

There are various forms of ontological heterogeneity [1]. These include:

Syntactic heterogeneity

This occurs when two ontologies are not expressed in the same
ontology language. A typical example is when ontologies are modeled using different knowledge representation formalisms suchas OWL and RDF.

Terminological heterogeneity

This occurs when different terms represent the same entities in different ontologies. This can be caused by the use of different natural languages. An example of this type of heterogeneity is surname and family name or paper and article which are synonymous terms.

Conceptual heterogeneity or semantic heterogeneity

This is the collective term for different possibilities to model a domain. It can be divided into three sub problems. These include the difference in coverage, granularity and perspective. Coverage differences occur if ontologies are written from the same point of view, that is, they are written in the same context and with comparable vocabulary, but the part of the domain that is described differs and there are only overlapping parts. Difference in granularity occurs when the same section of the domain is described but the depth of details is not equal. If the point of view from which ontology is designed differs, there is a difference in perspective.

By Ontology Alignment, we mean to come up with the Ontology, having sensible meaning with reutilization of objects in other Ontologies. The other Ontologies can be OR cannot be interconnected. The main idea behind the Ontology Alignment is to utilize the already available classes within the Ontology from other Ontologies. This reduces the reoccurrence of same classes multiple times to single instance based. In other words, to make them aligned with the entire ontology.

While comparing two or more Ontologies, the alignment refers to checking the occurrence of classes and instances along with their relationship in both or more Ontologies, thus to make them reutilized from one Ontology to all other, having same semantics. For instance, Name is the class, which can be associated with the Person Class as well as with the Brand class. They both have the same semantics. Syntactically, they are string type with Full Name and Short Name. Both aforementioned classes will utilize the same, therefore can be aligned.

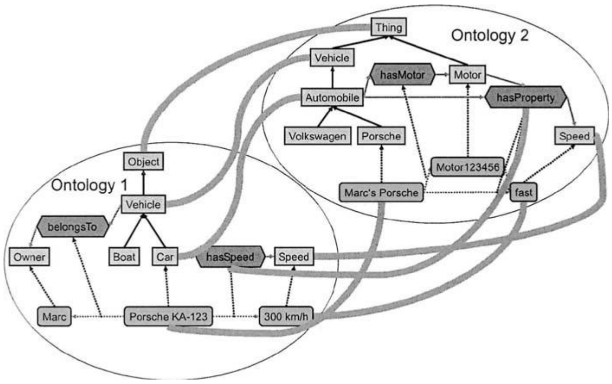

Figure 1. An Example of Ontology Alignment

According to the dictionary it gives a general comprehensible sense for alignment [1]. To align something means, "to bring into line". This very brief definition already emphasizes that aligning is an activity after which the involved objects are in some mutual relation. We here define our use of the term ontology alignment similarly to [2]. Given two ontologies, aligning one ontology with another one means that for each entity (concept, relation, or instance) in the first ontology, we try to find a corresponding entity, which has the same intended meaning, in the second ontology. An alignment therefore is a one-to-one equality relation. Obviously, for some entities no corresponding entity might exist

3. Related Work

This section focuses is to analyze how the different frameworks function and what similarities and differences exist among them, which is based on Shvaiko & Euzenat's classification [5].

Ontology matching can be processed by exploiting a number of different techniques. To provide a common conceptual basis, researchers have started to identify different types of ontology matching techniques and propose classifications to distinguish them. For example, Abels et al. propose a classification that consists of nine matching techniques based on existing literature studies. Another example is the classification developed by Shvaiko and Euzenat [3] [5]. Building on the foundation of Rahm and Bernstein's schema matching techniques classification, Shvaiko and Euzenat develop a meticulous classification to categorize elementary ontology and schema matching techniques [4]. Their classification focuses on techniques that exploit ontology-level information excluding instance data.

There are two synthetic classifications that can be viewed in top-down and bottom-up manner. The top-down view is called "granularity/input interpretation layer" which is based on granularity of match and then on how input information is interpreted. The bottom-up view is called "kind of input layer" and it is based on the kind of input requires in the matching process. "Granularity/input interpretation layer" and "kind of input layer" are further divided into one common layer called "basic techniques layer". Eight different types of elementary matching techniques are identified in this layer: string-based, language-based, constraint-based, linguistic resource, alignment reuse, upper level formal ontologies, graph-based, taxonomy-based, repository of structures and model-based.

Our finding consists of fifteen mediation tools, frameworks and methods with their inherent matching techniques. The most popular ontology matching techniques are string-based, taxonomy-based, constraint-based as well as linguistic resources techniques. Each of them is used by at least seven out of the fifteen mediation systems as shown in Table 1. In contrast, the least popular matching techniques are repository of structures technique and upper level formal ontologies. While the former technique is adopted by only one mediation system, the latter is not adopted by any system at all. Almost all systems in the survey incorporate a graph algorithm as their matching technique (either graph-based or taxonomy-based technique) with the exception of iPROMPT and Chimaera. For those who use graph algorithm as a matching technique except Glue, they include at least one additional matching technique in the system. Most of the mediation systems exploit multiple matching strategy which contains more than one matching technique. For instance, both COMA and COMA++ include six matching techniques in their inherent matching strategy. Thus leaving iPROMPT, Chimaera and Glue to engage with a single strategy in which only one matching technique is included in each system. In terms of execution approach, heuristic is widely implemented for carrying out string-based, language-based, constraint-based, linguistic resources, alignment reuses, graph-based, taxonomy-based and repository of structures matching techniques. Probabilistic reasoning approach, such as Bayesian network and machine learning, also play a part in the execution of taxonomy-based technique, whereas semantic reasoning is the dedicated approach used to execute model-based technique. Out of the fifteen mediation systems, eight of them are capable of performing ontology matching automatically, five of them still rely on human intervention and the remaining two allow users to execute ontology matching either automatically or semi-automatically.

4. Application of Ontology Matching

There are six central applications to Ontology mapping which are discussed in order below [11].

4.1 Ontology evolution

Ontology development which is known to be similar to software development, has developed in a distributed and collaborative manner. This enables multiple versions of the same ontology to often exist. A matching system is vital in this process if there is a system that uses a version of the ontology and intends to upgrade to another version. In this process, the aim of ontology mapping to discover the correspondences between the ontology versions.

4.2 Information Integration

Information Integration is certainly the most important application scenario for ontology matching systems because when multiple heterogeneous information sources are utilised for a task such as query answering, the sources must be matched and integrated [11]. A matching system is able to produce the connections between the sources by finding similar attributes, concepts or terms. This provide the basis for the integration process. This enables semantic interoperability among the systems to be integrated. Ontology mapping must be used by using the three methods discussed next, in order to facilitate information integration.

1. Single Ontology Approach: All of the data source schemas are directly mapped to a shared global ontology that provides a uniform interface to the user. But for this to be possible, all of the data sources must have nearly the same level of granularity, with the same perspective on a domain.

2. Multiple Ontology Approach: Each data source is represented on its own (local) ontology. Instead of Also additional representation formalism is required to define the inter-ontology mappings.

3. Hybrid Ontology Approach: This is the most effective approach. It is a combination of the two previous approaches .In this approach, local ontology is built for each data source schema, without mapping to other local ontologies, but to a global ontology. New data sourcescan be easily added by not modifying existing mappings.

4.3 Peer-to-Peer Networks

In peer-to-peer networks, systems may exchange information. The peers can be totally autonomous; they might describe their data using different kinds of concepts or labels. In order to achieve meaningful information exchange among the systems, the different concepts or labels have to be matched.

4.4 Web Service Composition

Web services may present their interfaces using different languages. A matching process is needed to match the service descriptions as well as the inputs and outputs.

4.5 Agent Communication

Agents are computer entities characterized by autonomy and capacity of interaction. They communicate through speech-act inspired languages, such as the FIPA Agent Communication Language. The actual content of the message is represented in knowledge representation languages which often refer to some ontology. As a consequence, when two autonomous and independently designed agents meet, they have the possibility of exchanging messages; however they would find it difficult to understand each other if they do not share the same content language and ontology. Thus, it is necessary to provide the possibility for these agents to match their ontologies.

4.6 Query Answering on the Web

The answering of queries is an important issue for the web. Search engines are used very often and they work with ontologies to refine queries. Each question that is posed by a user is translated into terms of the local ontology which is done by a matching process. Meta search engines translate a query into the terms of the different ontologies that are provided by the underlying search engines, collect the results and translate them back according to the original ontology.

5. Ontology Alignment Approaches

We will discuss each of the Ontology alignment approach with an example to make it understand in order to critically analyze in the next section.

5.1. IF-Map

Another system inspired by formal concept analysis is IF-Map [6]. It is an automatic method for ontology mapping based on the Barwise-Seligman theory of information flow [7]. The basic principle of IF-map is to align two local ontologies by looking at how these are mapped from common reference ontology. It is assumed that such reference ontology is not populated with instances, while local ontologies usually are. IF-Map generates possible mappings between an un-populated reference ontology and a populated local ontology by taking into account how local communities classify instances with respect to their local ontologies

The problem with using this approach to align our ontology is it assumes reference ontology is not incorporated with any of the instances. However, in majority of the cases instances do exist. Thus, voiding this approach for majority of the Ontologies.

5.2. Glue

Glue is a semi-automatic system that employs machine-learning techniques to find mappings [8]. It uses multiple learning strategies to cope with different types of information, either in data instances or in the taxonomic structure of the ontologies, in order to make predictions.

It consists of three main modules: Distribution Estimator, Similarity Estimator, and Relaxation Labeler.

The Distribution Estimator takes as input two taxonomies O1 and O2, together with their data instances. Then it applies machine learning techniques to compute for every pair of concepts their joint probability distributions. The Distribution Estimator uses a set of base learners and a meta-learner. Next, GLUE feeds the above numbers into the Similarity Estimator, which applies a user-supplied similarity function to compute a similarity value for each pair of concepts. The output from this module is a similarity matrix between the concepts in the two taxonomies. The Relaxation Labeler module then takes the similarity matrix, together with domain-specific constraints and heuristic knowledge, and searches for the mapping configuration that best satisfies the domain constraints and the common knowledge, taking into account the observed similarities. This mapping configuration is the output of GLUE.

The problem with using this approach to align our ontology is to have utilization of heuristic knowledge. For the specific reason, as it operates on machine learning approaches, it should first be trained well to actually apply prior knowledge in order to perform heuristic. Therefore extensive training to let the agent be knowledgeable is to be applied before actually utilizing the approach.

5.3. ONION

In their tool ONION (ONtology compositlON system), [9] provide an approach for resolving heterogeneity between different ontologies. Their basic assumption is that merging of whole ontologies is too costly and inefficient. Therefore, they focus on creating so called articulation rules, which link corresponding concepts. As manual creation of these rules is not very efficient either, they use a semi-automatic approach, which takes into account heuristics on several simple relations, such as labels, subsumption hierarchies and attribute values. Dictionary information is also used for the alignment process. From these relations a match is presented to the user who then has to decide whether the alignment is valid or not. The articulation rules linking can be applied when an application inquires information from two ontologies. The work is based on their theory on composition algebras [10].

As it uses the dictionary information as well as applies heuristic techniques to find the relationship between different classes or its instances to actually make the ontology aligned, it first proposes the matching results to user. The user is then to decide either to adopt it or not. Thus a semi-automatic approach, comparatively effective as it proposes to user to either accept or reject the alignment, while its not much effective in the way that they match the chunks of systems rather evaluating the entire system and let it be filtered for specific domain.

5.4. Prompt

PROMPT [11] is a tool that provides a semi-automatic approach to ontology merging. It is based on the SMART algorithm. After having identified alignments by matching of labels, the user is prompted to mark the entity pairs that should actually be merged. During merging, PROMPT presents possible inconsistencies such as name conflicts or relations not pointing anywhere any longer. The user then decides on how to react and resolve the issues manually.

In this approach, user has to apply the markers to each entity, thus having a time consuming activity before actually merging the ontologies. SMART algorithm works as follows:

- Setup: load files, set preferences,
- Execute operation: perform automatic updates, detect conflicts, create suggestions
- Select operation: choose from suggestion list, create a new operation,
- Initial suggestions: identical names, synonyms,
- Super-classes for top-level classes in alignment

It can therefore be seen that the algorithm cannot automatically be able to align the entire Ontology; rather user has to be involved extensively in putting up the markers.

5.5. Chimaera

Chimaera [12] is an interactive tool for ontology merging. Its basic ontology format is OKBC, but it can also handle other languages. After executing a linguistic matcher, Chimaera uses the results for triggering the merging operation. During this process, the human has to decide whether to merge or not. Chimaera also provides proposals on reorganizing.

The taxonomy once a merge has been processed. Overall, Chimaera allows diagnosing and manual editing for ontology merging. The actual alignment of entities however is based on simple measures.

Although this tool is impressive but for the basic alignment only. The issue with it lies that it is not even a semi-automatic alignment tool. It only provides platforms to syntactically check the entities and if find any, proposes the solution to user. The user is actually the person that will align after approving the proposed solution to merge the same structural entities, while the person has to do himself/herself the complete alignment by going through the entire ontology.

5.6. FCA-Merge

The FCA-Merge approach has been presented by [13]. As the name already suggests, its goal is to merge ontologies. It is based on formal concept analysis as described in [14]. Given two ontologies in a first step FCA-Merge populates them with instances that are extracted from a set of documents. This step is necessary, as most ontology does not have sufficient instances, but these are required for formal concept analysis. Based on these instances the ontologies are represented as concept lattices, i.e., concepts are seen as sets of instances. At this point lexical information is used to retrieve domain specific information. Using formal concept analysis the two contexts are integrated and a new lattice is created thereof. Pruning steps are applied to keep the size of the lattices small. In a last step, the lattice has to be transformed back into an ontology. This step has to be done manually. To solve conflicts, such as duplicates, FCA-Merge has an automatic support to guide the user through the process. One should mention that FCA-Merge deals only with concept hierarchies and underlying instances - alignment of relations is not supported.

The drawback to utilize is it relies on lexical information of specific domain. It only merges the ontologies rather actually aligning them. Set of Documents is therefore should be well documented with all the semantics of the entities so that the approach can find the relevant alignment entity from the set of document of the source Ontology. As being heavily relied on pre-documented material, it therefore makes it ineffective.

5.7. H-CONE

It is an approach to domain Ontology matching and merging exploiting different levels of interaction with a user [15] [16] [17]. First, an alignment between the two input ontologies is computed with the help of WordNet. Then, the alignment is processed straightforwardly by using some merging rules, e.g., renaming, into the new merged ontology. The HCONE basic matching algorithm works in six steps

- Chose a concept from first ontology
- Retrieve the WordNet senses of this concept;
- Retrieve hypernyms and hyponyms of these senses.
- Evaluate for the most frequent terms within the vicinity (senses, hponyms and herperonyms) their frequency of occurrence.
- Build a query from the related concepts related to the initial concept in the input ontology.
- Use Latent Semantic Indexing for determining the best sense to be used in the query

This method is way better than all the previous. Though it's a time consuming as well as requires extensive machine resources, but it actually check the concepts of the entity from its WordNet. Senses are then mapped and checked with senses in our Ontology and thus evaluation is performed, resulting in way better aligned Ontology. The issue remains there that WordNet should have an extensive repository to actually reconcile the concept of the entity in order to merge the ontologies.

5.8. S-Match

It is a generic semantic matching tool. It takes two tree-like structures and produces a set of mappings between their nodes. S-Match implements semantic matching algorithm in 4 steps. On the first step the labels of nodes are linguistically preprocessed and their meanings are obtained from the Oracle (in the current version WordNet 2.0 is used as an Oracle). On the second step the meaning of the nodes is refined with respect to the tree structure. On the third step the semantic relations between the labels at nodes and their meanings are computed by the library of element level semantic matchers. On the fourth step the matching results are produced by reduction of the node matching problem into propositional validity problem, which is efficiently solved by SAT solver or ad hoc algorithm [18] [19].

As it works on the method of testing entities on the basis of its requirements by comparing multiple entities having similar structure and therefore adopting the semantics from already aligned ontology to assign the same concept to this method, it makes relatively better aligned ontology. This has the same problem as earlier discussed, which is to have extensive WordNet to actually check the semantics through traversing its structural properties with the one, maintained in repository.

6. Critical Analysis

Following is the summarized critical analysis which we will discuss later in the section.

TABLE I. Summary of Mediation Systems

Approach	***Automation***	***String Based***	***Language Based***	***Linguistic Based***	***Taxonomy Based***	***Model Based***
IF-Map	Full	Heuristic			Heuristic	Semantic Reasoning
GLUE	Semi				Probabilistic Reasoning	
ONION	Full	Heuristic			Probabilistic Reasoning	
PROMPT	Semi	Heuristic				
Chimaera	Semi	Heuristic				
FCA-Merge	Full		Heuristic	Heuristic		
H-CONE	Semi					
S-Match	Full	Heuristic	Heuristic	Heuristic	Heuristic	Semantic Reasoning

We can see that the fully automated includes IF-MAP, ONION, FCA Merge and S-Match while other remains Semi-Automatic. Although we have seen that automatic cannot completely aligned the Ontology, as we discussed earlier due to the limited specified in each approach. However, in Semi-automatic, there is a reliance over end-user that gets suggestions from the software, while proposing an idea to end-user either the person wants to pursue with the proposed solution or not. Other than FCA-Merge and H-Cone, methods works over the Heuristic String based searches. This means, all other required well-trained intelligent Agents (IAs) that are well updated of the string based matching for alignment of entire ontology. However, we can see that FCA-Merge is heuristic in terms of language based searches. This is more appropriate to use the dictionary based approach to align the already maintained repository. While, language based, it also applicable for linguistic based which means to have multiple semantics of single class thus aligning heuristically as per the context of the Ontology. Taxonomy based approaches include IF-MAP and S-Match which actually needs IAs to get trained to taxonomically align the Ontology, while Probabilistic approaches, working on the chances, includes GLUE and ONION. It also depends on the past records to come up with the percentage chance of occurrence to near-to-accurate auto-alignment of Ontology.

Similarly, both aforementioned also are Model-based semantically reasoning approaches to align the Ontology. Semantic alignment through reasoning is way better than any heuristic approach as the Ontology can better be aligned as per their context. Combining both the techniques nearly aligned Ontology to perfect.

With the above summary, we can say that the S-Match approach has almost all the attributes that actually dependent to accurately align the Ontology. Thus we can recommend to the readers to utilize the same in their Ontology and come up with the better results than other mentioned approaches.

TABLE II. Characteristics of studied ontology mapping and merging systems

Approach	***Interoperability Language***	***Ontology Structure***	***Strategy***	***Additional Resources***
IF-Map	RDF,KIF, Ontologua, Protégé -KB, Prolog	Concept	Linguistic, Reasoning	Reference Ontology
GLUE	-	Concept, Properties, Instance	Probability	-
ONION	IDL, XML-Based	Concept, Properties	Linguistic	WordNet
PROMPT	OWL, RDFS	Concept, Properties, Instance	Linguistic, Heuristic	-
Chimaera	Ontologua		Linguistic, Heuristic	-
FCA-Merge	-	Concept, Instance	Linguistic, Heuristic	-
H-CONE	OWL-DL	Concept, Properties,	Linguistic, Reasoning	WordNet
S-Match	-	Concept	Linguistic, Reasoning	WordNet

From table II , we can clearly see that there exists common attributes while automation is semi in effective approaches while fully automated works on probabilistic reasoning.

7. Rely/Guarantee Approach for Ontology Alignment

In the past, it was easy to make an output of a system as an input of another system. This is because all the programs being generated were actually based on sequential methods. That is, the systems always rely on the output of its dependent system. However, when it comes to parallel processing, more research was performed which results in providing solution as an integral way, which was previously being done independently.

Rely-Guarantee is a method which assumes that not only a component is verified by just satisfying all of its commitments, but also verifies all the assumptions, being exposed from the internal or external environment.

This can better be explained with the real-life example that if we want to have dinner, we can either go out to some restaurant, can cook at home, or can place order to get it delivered at the door-step. Now, the process of having dinner is dependent on either or the commitments from the mentioned processes. For first option, we need to drive way to the restaurant to reach the destination, so “Traveling” is the process which needs to be committed. Similarly, if we want to cook at home, we first need to go for “Shopping” to buy ingredients. The process commitment will enable us to cook or Initiate cooking process, so that dinner system can be practiced. The last option is to let the dish be delivered at home. For that, two commitments, a phone call and book and order while another process by the delivery-boy to bring it to your doorstep, should be committed so have dinner.

So, if all the assumptions from the environment are verified, while the commitments are fulfilled, the system can have its input to ignite the system. In other words, all pre-requisite systems should have commitments in order to provide feed to dependent system to let the process flow

Any analyst can better come up with the systems and apply the assumption/guarantee method to align the ontology of that particular system. For that, the analyst first have to come up with the no. of systems exists within entire system. This means to, bifurcate the entire process into small systems which are dependent on each other, if related. The focus while deriving the sub-systems should be on how the system verifies the initial process, and all sub-systems do rely on prior system to come-up with the consequence.

The technique can be adopted to align the ontologies. That creates the possibility of aligning such ontologies that cannot be aligned with the traditional methods. We will further discuss different methods to align the ontologies in different ways. Further, with the methods, we can also automate the alignment process by developing a program that can align ontology accordingly.

8. Critical Analysis of Rely/Guarantee

As per our proposed method, we clearly see that the approach relies on how actually the Ontology is architected. It is the approach in a way to avoid medications rather relying on Prevention. It is therefore more focused on how the Ontology is being developed, i.e. proper Input and Output are analyzed while designing the Ontology. Context based searched are utilized at the time when the Ontology is being designed. This is because while creating the Ontology, one needs to find out if particular class or attribute exists. If yes, then it is assigned other-wise new Inputs and Outputs along with the specific Attribute or Class are published before its actual utilization.

In comparison to the approaches, discussed above, it is a semi-automatic approach, depends heavily on Taxonomy, heuristic in language, linguistics, and semantics and works on additional resource as repository. A person, designing the ontology and aligning the same through this approach requires concentration as being a time-consuming, but the most effective way of making the Ontology effective for context based searches.

Therefore, we can conclude that as it has all the attributes which are not completely available in any of the aforementioned approach, while it also aligns more effectively as dependent on comparing Inputs and Outputs and referring to other Ontologies in similar fashion, while is required extensive efforts is the most effective way of designing the Ontology and making it Aligned as per the requirement.

9. Conclusion

With critically analyzing all the approaches, we can say that there isn't any approach exists that can be fully automated and effective enough to align the entire Ontology. Though, there are a semi-automatic approach that works on probabilistic and heuristics approaches that are effective however no model can completely aligned the entire ontology at once. All have the drawbacks to actually having a strategy of reasoning and linguistic approaches while many are dependent on the dictionary approaches, which means to have a repository of already published classes or attributes that can be referred. Therefore, this area still needs to be researched to come up with the approach of designing the ontology which is actually aligned or an effective alignment methodology to make a semantic environment.

10. References

[1] The Merriam Webster Dictionary(2006). Merriam-Webster. http://www.m-w.com/.

[2] Klein, Michel (2001). Combining and relating ontologies: an analysis of problems and solutions. In Gomez-Perez, Asuncion, Gruninger, Michael, Stuckenschmidt, Heiner, and Uschold, Michael, editors. Proceedings of Workshop on Ontologies and Information Sharing at IJCAI-01, Seattle, WA, USA

[3] Abels, S., Haak, L., Hahn, A. (2005). "Identification of Common Methods Used for Ontology Integration Tasks". In: 1st International ACM Workshop on Interoperability of Heterogeneous Information Systems (IHIS), pp. 75-78. Bremen, Germany

[4] Rahm, E., Bernstein, P. (2001), "2001, A Survey of Approaches to Automatic Schema Matching". The International Journal on Very Large Data Bases (VLDB), 10(1), pp. 334-350.

[5] Shvaiko, P., Euzenat, J. (2005), "A Survey of Schema-based Matching Approaches". Journal on Data Semantics, IV, pp. 146-171

[6] [Kalfoglou and Schorlemmer, 2003a] Yannis Kalfoglou and Marco Schorlemmer. If-map: an ontology mapping method based on information flow theory. Journal of data semantics, 1:98–127, 2003.

[7] [Barwise and Seligman, 1997] Jon Barwise and Jerry Seligman. Information flow: the logic of distributed systems, volume 44 of Cambridge Tracts in Theoretical Computer Science. Cambridge, University Press, Cambridge (UK), 1997.

[8] [Doan, J. Madhavan, P. Domingos, and A.Y. Halevy, "Learning to Map Between Ontologies on the Semantic Web", In Proceedings of WWW, 2002, pp.662-673.

[9] Mitra, Prasenjit, Wiederhold, Gio, and Kersten, Martin (2000). A graphoriented model for articulation of ontology interdependencies. In Proceedings of the Conference on Extending Database Technology 2000 (EDBT-00), volume 1777 of Lecture Notes in Computer Science, pages 86-1-, Konstanz, Germany. Springer.

[10] Mitra, Prasenjit and Wiederhold, Gio (2001). An ontology-composition algebra. Technical report, Stanford University, Stanford, CA, USA.

[11] Noy, Natasha F. and Musen, Mark A. (2000). PROMPT: Algorithm and tool for automated ontology merging and alignment. In Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000), pages 450-455, Austin, TX, USA. AAAI Press / The MIT Press.

[12] Noy, Natasha F. and Musen, Mark A. (2001). Anchor-PROMPT: Using nonlocal context for semantic matching. In Workshop on Ontologies and Information Sharing at the 17th International Joint Conference on Artificial Intelligence (IJCAI-2001), pages 63-70, Seattle, WA, USA.

[13] Noy, Natasha F. and Musen, Mark A. (2002). PromptDifi': a fixed-point algorithm for comparing ontology versions. In Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-02), pages 744-750, Edmonton, AB, Canada. AAAI Press.

[14] McGuinness, Deborah L., Fikes, Richard, Rice, James, and Wilder, Steve (2000). The Chimaera ontology environment. In Proceedings of the 17th National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, pages 1123-1124, Austin, TX, USA. AAAI Press / The MIT Press.

[15] Stumme, Gerd and Maedche, Alexander (2001). FCA-Merge: Bottom-up merging of ontologies. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2001), pages 225-230, Seattle, WA, USA.

[16] Ganter, Bernhard and Wille, Rudolph (1999). Formal Concept Analysis: Mathematical Foundations. Springer.

[17] [Kotis et al., 2006] Konstantinos Kotis, George Vouros, and Konstantinos Stergiou. Towards automatic merging of domain

ontologies: The HCONE-merge approach. Journal of Web Semantics (JWS), 4(1):60–79, 2006.

[18] [Vouros and Kotis, 2005] George Vouros and Konstantinos Kotis. Extending HCONE-merge by approximating the intended interpretations of concepts iteratively. In Proc. 2nd European Semantic Web Conference (ESWC), volume 3532 of Lecture notes in computer science, pages 198–210, Hersounisous (GR), May 2005.

[19] [Kotis and Vouros, 2004] Konstantinos Kotis and George Vouros. HCONE approach to ontology merging. In Proc. 1st European Semantic Web Symposium (ESWS), volume 3053 of Lecture notes in computer science, pages 137–151, Hersounisous (GR), 2004.

[20] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-match: an algorithm and an implementation of semantic matching. In Bussler C., Davies J., Fensel D., and Studer R., editors, The semantic web: research and applications, volume 3053 of LNCS, Heraklion, 10-12 May 2004.

[21] F. Giunchiglia, M. Yatskevich, and E. Giunchiglia. Efficient semantic matching. In Proceedings of the 2nd european semantic web conference (ESWC’05), Heraklion,29 May-1 June 2005.