Ontology Mapping Algorithms and Tools: A State of Art

Prof. Rajesh Gaikwad[1] Prof. Mahendra Patil[2] Prof. Deepali Maste[3]
[1,2,3]Assistant Professor
[1,2,3]Atharva College of Engineering, Mumbai, India

Abstract— In the world of Semantic Web, a single ontology is no longer enough to support the tasks of heterogeneous computing environment. There are so many applications accessing the web and many of them require different ontology at a given time. This makes our task difficult for accessing the ontology in the standard semantic web environment. It is equally important to understand how this ontology is mapped and their mapping algorithm. This paper presents you with few of the Ontology mapping Algorithms and tools used.

Key words: Ontology, Ontology Mapping Algorithms

I. INTRODUCTION

Ontology is defined as explicit specification of conceptualization and the relationships between those concepts. It is used to reason about the properties of that Domain D and also may be used to define that domain. Ontology describes individuals (instances), classes (concepts), attributes, relations and events. We use Ontology typically in Semantic Web, Software Engineering, Biomedical Informatics, Library Science and Artificial Intelligence.

(Shvaiko and Euzenat, 2005). Ontologies are useful for web applications and for those who want to access information from different sources on internet. In particular, on the World Wide Web there is a huge amount of databases serving various applications and web servers, however, these databases and their applications have been designed with their own ontology and schema which leads to a lack of standardization and therefore pose numerous difficulties for other applications to access these various data sources together. An ontology is a formal description of a domain of discourse, intended for sharing among different applications, and expressed in a language that can be used for reasoning. A database schema differs from an ontology. An ontology provides more formal expression than a database schema. A database schema is thus more restricted, generally used for a specific database, does not provide explicit semantics for data and is not reusable. Often, they can be seen as formal descriptions at different levels of abstraction (where ontologies are the higher level). Ontology mapping is the process of aligning two given ontologies, and the similar process done on schemas is called matching (Noy, 2004).

We shall now consider a example of web-based Air conditioner (AC) selling. There is a user who can only afford to spend 25000 rupees, and would prefer a Blue star AC. A web service application provides services for searching for based on input parameters. The user submits a query to search for AC less than 25000 rupees and only ACs from Blue Star manufacturers. The application then needs to search many sources of data over the web (databases of different AC selling companies and databases of personal listings). The problem is, all of these databases have been created independently, and therefore a single query will not work that can be applied to all databases. One possible solution is to determine the schemas for all possible data sources and create unique queries for each one, however, with data sources ranging in the hundreds (if not thousands), this is simply not feasible.

Thus we require another approach for dealing with the above problem is ontology mapping. Ontology mapping allows different schemas or ontologies to be (semi)- automatically mapped (or matched) to each other to determine information that is similar even if the schematic syntax or structure is different. This problem, although widely researched (Kalfoglou and Schorlemmer, 2003; Noy, 2004; Shvaiko and Euzenat, 2005), is still not fully "solved". At best, the results of machine-oriented ontology mapping can produce accuracy to approximately 75% (Noy and Musen, 2001). There are techniques that can improve on this number using human intervention or having developer's *a priori* design ontologies, however, these are more time intensive and therefore cost more and so more automatic solutions are desirable. Because of this ongoing research, the state-of-the art in this area is continually improved.

This paper begins with terminology for ontology and mapping , the third section is on related work in ontology mapping , the fourth section talks about where ontology mapping is placed within ontology operation management, the fifth and sixth sections are different algorithms and tools they support. The paper concludes with seventh section.

II. TERMINOLOGY

A. Ontology

On light-weight ontologies, are build on RDF/S2 to represent ontologies. To facilitate the further description, we briefly summarize its major primitives and introduce some shorthand notations. An RDF model is described by a set of statements, each consisting of a subject, a predicate and an object. An ontology *O* is defined by its set of Concepts *C* (instances of "rdfs:Class") with a corresponding subsumption hierarchy *HC* (a binary relation corresponding to "rdfs:subClassOf"). Relations

R (instances of "rdf:Property") exist between single concepts. Relations are arranged alike in a hierarchy *HR* ("rdfs:subPropertyOf"). An entity $i \in I$ may be an instance of a class $c \in C$ ("rdf:type"). An instance $i \in I$ may have one *j* or many role fillers from *I* for a relation *r* from *R*.We also call this type of triple (*i*, *r*, *j*) a property instance

B. Mapping

We here define our use of the term "mapping". Given two ontologies*O*1 and*O*2, mapping one ontology onto another means that for each entity (concept *C*, relation *R*, or instance *I*) in ontology *O*1, we try to find a corresponding entity, which has the same intended meaning, in ontology *O*2.

1) Definition

We define an ontology mapping function, map, based on the vocabulary, E, of all terms e ∈ E and based on the set of possible ontologies, O, as a partial function:

$$\text{map}: E \times O \times O \_ E,$$
$$with\ \forall e \in O1(\exists f \in O2 : \text{map}(e,O1,O2) = f \vee \text{map}(e,O1,O2) = \bot).$$

A term e interpreted in an ontology O is either a concept, a relation or an instance, i.e. e|O ∈ C ∪ R ∪ I. We usually write e instead of e|O when the ontology O is clear from the context of the writing. We write mapO1,O2 (e) for map(e,O1,O2).We derive a relation mapO1,O2 by defining mapO1,O2 (e, f) ⇔ mapO1,O2 (e) = f. We leave out O1,O2 when they are evident from the context and write map(e) = f and map(e, f), respectively. Once a (partial) mapping, map, between two ontologies O1 and O2 is established, we also say "entity e is mapped onto entity f" iff map(e, f). An entity can either be mapped to at most one other entity.A pair of entities (e, f) that is not yet in map and for which appropriate mapping criteria still need to be tested is called a candidate mapping.

III. Related work

In the previous literature, many Ontology Management Tools/systems and related works are available. Among them, Namyonu et al.,[6] have reported about the tools, systems, and related work of ontology mapping. They explain about three ontology mapping categories as

1) Mapping between an integrated global ontology and local ontologies,
2) Mapping between local ontologies and
3) Mapping on ontology merging and alignment.

In their work comparison has been done on the evaluation criteria, which are input requirements, level of user interaction, type of output, content of output, and the five dimensions: structural, lexical, domain, instance based knowledge, and type of result. Addition, to this comparative results have been given on ontology mapping tools. Natalya [7] has given a brief survey of the approaches to semantic integration developed by researchers in the ontology community. They have focused on the approaches that differentiate the ontology research from other related areas. They have discussed different techniques for finding correspondences between ontologies, declarative ways of representing these correspondences, and use of these correspondences in various semantic-integration tasks. There exist numerous ontology visualization methods and also a number of techniques used in other contexts that could be personalized for ontology representation. Akrivi Katifori et al., [8] have presented these techniques and categorized their characteristics and features in order to support method selection and encourage future research in the area of ontology visualization. Matteo Cristani et al., [9] have provided a framework for analyzing the methodologies that compares them to a set of general criteria. A classification has been obtained based upon the direction of ontology construction; bottom-up or top-down. It is also claimed that the resulting classification is useful not only for theoretical purposes but also in the practice of deployment of ontologies in Information Systems.Elena Simperi et al., [10] have given an article based on empirical evidence and real world findings of the methodologies, methods and tools currently used to perform ontology reuse processes. They have done the analysis on the most prominent case studies for ontology reuse in the area of eHealth and eRecruitment. Bernhard et al., [11] have done the survey and given a general overview of the field of metadata interoperability by providing a categorization of existing interoperability techniques, describe their characteristics, and compare their quality of analyzing their potential for resolving various types of heterogeneities. Their analysis explicitly showed that metadata mapping is the appropriate technique in integration scenarios. Good surveys through the recent years are provided. Yannis et al., [12] focuses the survey on current state of the art in ontology mapping. Ravi et al.,[13] does the analysis on ontology mediation tools. These authors review recent approaches, frameworks, techniques and tools. However, none of the surveys provide a comparative review of the existing ontology management techniques and systems. In this paper an attempt has been carried out by us for the classification of ontology management methodologies.

IV. Ontology operation management

Most of the applications use multiple ontologies, particularly when using modular design of ontologies or when we need to combine with systems that use other ontologies. Ontology management will integrate all the operations together and make finest application for the easy ontology process, this all operations are essential for the preservation and integration of ontologies.

We shall discuss few of these Ontology operation management techniques. Alignment is a method of mapping between ontologies in both orders whereas it is feasible to change original ontologies so that the proper translation exists. Thus it is possible to add new concepts and relations to ontologies that would structure proper equivalents for mapping. The condition of alignment is called articulation. Refinement is mapping from ontology O1 to another ontology O2 so that every concept of ontology O1 has comparable in ontology O2, however primitive concepts of ontology O1 may communicate with non-primitive concepts of ontology O2. The following figure illustrates Ontology operation management techniques. The diagram gives the clear idea that ontology mapping is one of the operation for ontology operation management techniques.

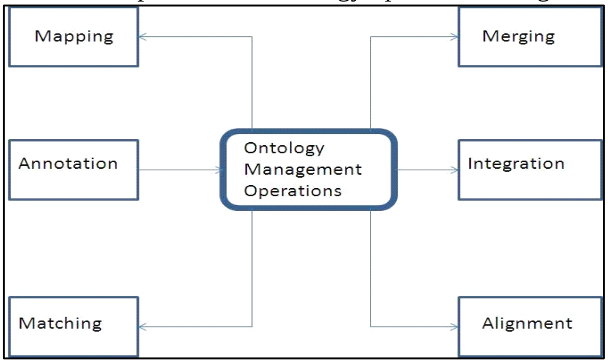

Fig. 1: Ontology Operation Management

A. Ontology Mapping

Mapping from ontology is expressing of the method how to translate statements from ontology to the other one. In the simplest case it is mapped from one concept of the first ontology to one concept of the second ontology. It is not always possible to do such one to one mapping. Some information can be lost in the mapping. This is acceptable; however mapping may not establish

V. ALGORITHM

A few of Ontology mapping Algorithm which are common are discussed in this section. Figure 2 illustrates the use of the Algorithm and tools.

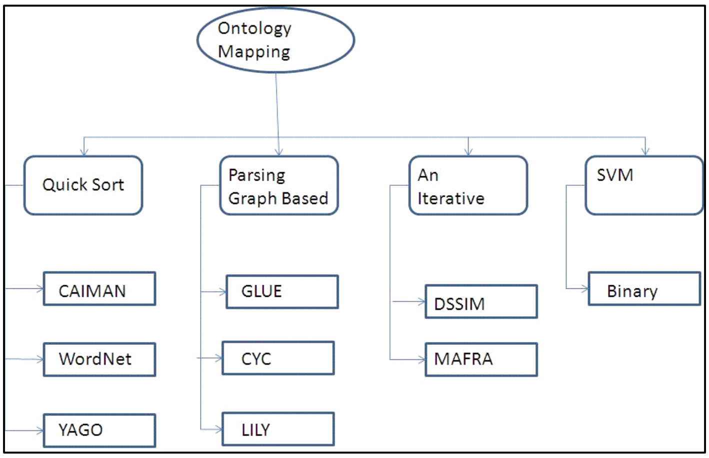

Fig. 2: Ontology Mapping Algorithm and tools

A. Quick-Sort Algorithm

Quicksort Algorithm helps to explain the similarity that is being done with simple additive weighting on ontology concept for ontology Mapping. This Algorithm during its implementation defines the ontology property relation and get the accurate mapping result for solving the problems. It helps the ontology structure similarity in computing the lexical similarity.

B. Parsing graph-based algorithm

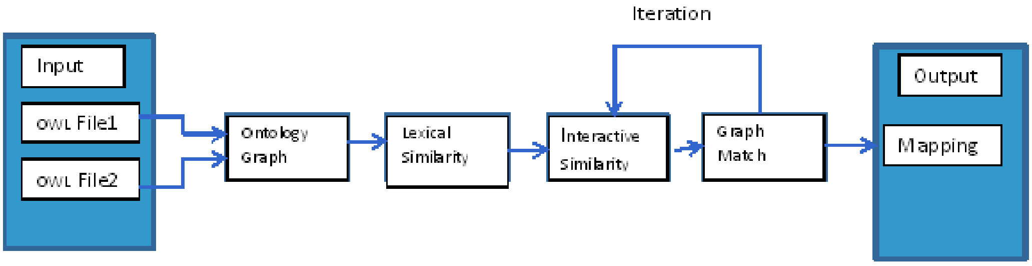

Fig. 3: Ontology parsing graphs Algorithm

This Algorithm has 5 steps viz. ontology parsing, ontology parsing graph generation, lexical similarity calculation, similarity iteration and graph match (Figure 3).

The ontology parsing graph (OP-Graph) extends the general concept of graph and encodes logic relationship and semantic information which the ontology contains into vertices and edges of the graph. Thus, the problem of ontology mapping is translated into a problem of finding the optimal match between two OP-graphs.

C. Iterative Algorithm

Algorithm formally specifies the iterative method of calculating the overall similarity. Iterative algorithm in general allow user-guided input and provide the capability to run multiple iterations of the alignment algorithms as needed until an acceptable result set is determined. In the majority of ontology alignment scenarios, the user will have little knowledge of the contents of at least one of the ontology they desire to align. Therefore, instead of querying the user to identify alignments that they know beforehand to be correct, we instead provide an initial result set for the user to analyze.

He user can then reject alignments they determine to be incorrect. All rejected alignments are submitted back into Iterative algorithm to help guide the alignment process and mold the subsequent result set. This is accomplished by setting the alignment rating for each rejected pair in the sparse matrix data structure used to represent alignments to zero. Every completed iteration will present an original result set to the user. In each set, original alignments are suggested for elements that had been previously misaligned. All nonrejected alignments are implicitly assumed to be correct and remain the same in the new alignment set. A database of rejected alignments is also stored to ensure that previously identified misaligned elements are not presented again in future iterations to the user. Although this approach requires a moderate user degree of involvement to confirm or reject each alignment suggestion, the matching effort to align the elements is still automated and relieves the user of the task of determining correct alignments manual (D. Chen , et al)

D. SVM

The Support Vector Machine (SVM) algorithm was developed by Cortes and Vapnik, 1995. It is probably the most widely used kernel learning algorithm. It achieves relatively robust pattern recognition performance using well established concepts in optimization theory. Despite this mathematical classicism, the implementation of efficient SVM solvers has diverged from the classical methods of numerical optimization.

VI. TOOLS

A. Binary Classification

Ming Mao dealt ontology mapping problem with machine learning techniques. His approach has five steps:

- generated various domain independent features,
- Randomly generates training and testing set for OAEI benchmark tests.
- Train an SVM model on the training set.
- Classify testing data on the trained SVM model.
- Extract mapping results of testing data. Testing data are evaluated against ground truth. Steps 2 to 6 is repeated 10 times and the average evaluation result is used to eliminate bias.

B. GLUE

Doan developed a system, GLUE, which employs machine learning method to discover mappings. The system consists of three phases: the sharing estimator, the similarity estimator and the relaxation labeler. The distribution estimator takes as input the two taxonomies O1 and O2 together with their instances and applies machine learning to calculate the four probabilities. The similarity estimator applies a user-supplied function, such as the Jaccard Coefficient or MSP and computes a similarity value for each pair of concept. The relaxation labeler takes as input the similarity values for the concepts from the taxonomies and searches for the best mapping configuration, exploiting user supplied domain specific constraints and heuristics.

C. CAIMAN

In order to provide the above mentioned services, the CAIMAN system has to map one ontology onto another. This means that for each concept node in ontology graph A, a corresponding concept node in ontology graph B has to be found. We calculate a probability measure that for a node ai in the personal ontology, node bj in the community ontology is the corresponding node. The node bj with the maximum probability measure wins. Currently we perform a mapping that does not consider the graph structure of the personal ontology. Since we expect the mapping to improve when the interconnection of nodes from the personal ontology is considered, we plan to include this in future work. However we take some of the graph structure of the community ontology into account. We calculate two probability measures for the two most probable nodes *bj* and bk that could match ai and if their difference is above a certain threshold, we say that bj and ai are corresponding nodes. However, if their difference is below the threshold, we calculate the probability difference of the parent nodes of bj and bk and so on until the difference is above the threshold. If we get to a common parent node or there is no parent node, the user has to decide, which of the pre-selected nodes are to be considered corresponding nodes. The probability measure *p(ai, bj)* we base our mapping on will be explained in the next section. To find corresponding nodes, we apply the following simple algorithm:

1. for all concept nodes ai E A (breadth first)
(a) for all nodes bj E B (breadth first) calculate p(ai, bj)
(b) Find bj and bk such that p(ai, bj) is maximized and then p(al, bj) - p(ai, bk) is minimized.
(c) /fd := p(ai, bj) - p(ai,bk) < t mark ai and bj corresponding
(d) else repeat for the parent nodes of the current b nodes until a decision has been made
i. calculate d for the current nodes
ii. if d > t, pick the (grand*)child node of bj
iii. if there is no parent node for one of the b nodes or they have a common parent, let the user decide which node to pick

D. DSSIM

Miklos Nagy developed a system for ontology alignment (DSSim). It takes a concept (or property) from ontology 1 and considers it as the query fragment. From that the graph is built. Then takes syntactically similar concepts and properties and its synonyms to the query graph from ontology 2 and graph is built. Different similarity algorithms are used to assess quantitative similarity values between the nodes of the query and ontology fragment. Then the information is combined using the Dempster's rule. Based on the combined evidences they assess semantic similarity between the query and ontology graph fragment structures and select those in which they calculate the highest belief function. The selected concepts are added into the alignment.

E. Mafra

Alexander has proposed a framework for mapping distributed ontologies. MAFRA architecture consists of a set of modules organized along horizontal and vertical dimensions. Horizontal modules correspond to five fundamental phases namely, lift and normalization, similarity, semantic bridging, execution and post-processing. The vertical modules correspond to four phases; namely, evolution, domain knowledge and constraints, cooperative consensual building and GUI. In the lift and normalization phase, ontologies are imported. In similarity phase, similarities between ontology entities are calculated. In semantic bridging phase the similar entities are semantically bridged. In the execution phase, the mappings are exploited. The post processing step is based on the execution results. In the evolution step the changes in the source and target ontologies are synchronized with the semantic bridges defined by the semantic bridge module. In the cooperative consensus-building phase the tool helps to setup a consensus between the various proposals of people involved in the mapping task.

F. WordNet

WordNet was developed in 1998 as a lightweight ontological method, nearer to thesauri, and it is a lexical database in English for the semantic similarly of words in Information recovery research. WordNet contains a vast quantity of information. WordNet represents nouns, adverbs, verbs and adjectives as a group of cognitive synonyms with their own individual concepts. A browser is used to control and navigate the individual component in WordNet. It categorizes English words into some groups, such as hypernyms, synonyms, and antonyms.

G. CYC

CYC is developed by Lenat as part of his research work for MCC Corporation. The ontology in CYC knowledge has 47,000 concepts and 306,000 facts brows able by the CYC web interface. CYC uses a mapping to identify the concepts of each word. For example, CYC provides part of the relationship between tree and leaves every concept mapped to the terms will return either a true or false statement. Based on this return value, users can then decide the suitable actions for potential processing. CYC has been successfully applied to Terrorism Knowledge Based application and has been used as part of Cyclopedia database.

H. Babenet

BabelNet is developed to overcome the drawback of WordNet. BabelNet integrates the domain and knowledge base of these two systems, and could adequately supply the users with higher level ontology domain. In addition, BabelNet is also able to differentiate word sense disambiguation exactly using the information provided by Wikipedia domain knowledge.

I. Yago

Yet Another Great Ontology (YAGO) is developed by Fabian and it is a lightweight ontology with extensible functionalities for high data coverage and accurateness. YAGO achieved an accuracy of 95% on its test cases. YAGO extracted data from Wikipedia and combined it with WordNet, and provides the users with 1 million entities and 5 million facts.

J. Lily

Peng Wang had given an ontology mapping system Lily. Lily realized four main functions: i Generic Ontology Matching method (GOM) is used for common matching tasks with small size ontologies. ii. Large scale Ontology Matching method (LOM) is used for the matching tasks with large size ontologies. iii. Semantic Ontology matching method (SOM) is used for discovering the semantic relations between ontologies. iv. Ontology mapping debugging is used to improve the alignment results. The alignment process mainly contains three steps:

- Preprocessing step parses the ontologies.
- Match computing step uses suitable methods to compute the similarity between elements from different ontologies.
- Post processing step is responsible for extracting, debugging and evaluating mappings.

VII. CONCLUSION

The several Algorithm and tools that we have discussed in our paper, are developed over the period time with the purpose and changes in the technology. Each one has its importance in their respective domains. It has been observed that not all algorithm are semantic based. Semantic based algorithm native for ontology mapping and in order to achieve Semantic Web Service. We need to modify structured algorithm such OWL with Semantic annotation or design algorithm useful for constructions and understanding of Domain Models. A summarized table 1 for Algorithm is as shown below.

Acknowledgement

This paper is prepared for National Conference on Technology Advancement and Atomization in Engineering College which is to be held in AET's Atharva College of Engineering Mumbai, India. I am grateful to kind support and lab facility provided by Principal, Management and Head of Department of Computer Engineering.

Tools Name	Algorithm	Language Used	Application	Methodologies	Operation
Binary Classification	SVM	R language	Structural and Web Features	Machine Learning	Ontology Mapping
GLUE	Parsing graphbased algorithm	*	Structural and Web Features	Machine Learning	Ontology Mapping
Caiman	Quick-sort	*	Text Classification	Machine Learning	Ontology Mapping
DSSIM	QOM	OWL	Assess quantitative similarity values	Structured Based	Ontology Mapping
Mafra	QOM	JAVA and KALOM	Setup Consensus between various proposals	Semantic Based	Ontology Mapping
WordNet	Quick-sort	*	Hypernyms and Synonyms	Semantic Based	Ontology Mapping
CYC	Parsing graphbased algorithm	JAVA	Terrorism Knowledge	Semantic Based	Ontology Mapping
Babenet	Parsing graphbased algorithm	*	High Level Information domains	Semantic Based	Ontology Mapping
Yago	Quick-sort	JEAN API	Artificial Intelligence	Semantic Based	Ontology Mapping
Lily	Parsing graphbased algorithm	JAVA	Preliminary analzer	Hybrid	Ontology Mapping

Table 1: Ontology Mapping Algorithm

References

[1] (Shvaiko and Euzenat, 2005) Pavel Shvaiko and Jérôme Euzenat, A Survey of Schema- Based Matching Approaches, Journal on Data Semantics IV, Volume 3730, 2005, pp. 146-171.
[2] (Noy, 2004) Natalia F. Noy, Semantic integration: a survey of ontology-based approaches, ACM SIGMOD Record: Special section on semantic integration, 2004, pp 65-70.
[3] (Noy and Musen, 2001) N. Noy and M. Musen. *Anchor-PROMPT: Using NonLocal Context for Semantic Matching*. In Proceedings of the Workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence
[4] (Kalfoglou and Schorlemmer, 2003) Yannis Kalfoglou and Marco Schorlemmer, Ontology Mapping: The State of the Art, The Knowledge Engineering Review, Volume 18, Issue 1 (January 2003), pp. 1-31.
[5] "Survey of Ontology Mapping Techniques" By Savithri Godugula Software Quality and Assurance and Professor Gregor Engels Jun 15 2008
[6] Namyoun Choi, Il-Yeol Song, and Hyoil Han, "A Survey on Ontology Mapping", SIGMOD, Vol. 35, No. 3, Sep. 2006
[7] Natalya F. Noy, "Semantic Integration: A Survey Of Ontology-Based Approaches", SIGMOD, Vol.33, No. 4, December 2004.
[8] Akrivi Katifori and Constantin Halatsis, "Ontology Visualization Methods—A Survey", ACM Computing Surveys, Vol. 39, No. 4, Oct 2007.
[9] Matteo Cristani and Roberta Cuel, "A Survey on Ontology Creation Methodologies", International Journal on Semantic Web & Information Systems 1(2), 49-69, June 2005.
[10] Elena Simperi, "Reusing Ontologies on the Semantic Web: A Feasibility Study", Science Direct, Data and Knowledge Engineering, Vol: 68, issue 10, p: 905-925, Oct 2009.
[11] Bernhard haslhofer and Wolfgang Klas, "A Survey of Techniques for Achieving Metadata Interoperability", ACM Computing Surveys, Vol: 42, No:2, Feb 2010.

[12] Yannis, K. and S. Marco, "Ontology mapping: The state of the art", The Knowledge Engineering Review, 18: p: 1-31, 2003.
[13] Ravi Loudrusamy and Gopinath Ganapathy, "Feature Analysis of Ontology Mediation Tools", Journal of Computer Science, 4(6), p: 437-446, 2008.
[14] Chen1 , J. Lastusky , J. Starz , and S. Hookway "A User Guided Iterative Alignment Approach for Ontology Mapping"
[15] "Efficiency of Ontology Mapping Approaches" Marc Ehrig and Steffen Staab ,Institute AIFB, University of Karlsruhe p 2-3.