

# METHODOLOGY BASED SURVEY ON ONTOLOGY MANAGEMENT

Chitra Ramesh<sup>1</sup> and Aghila Gnanasekaran<sup>2</sup>

<sup>1</sup>Department of Computer Science, Pondicherry University, Puducherry  
chitrashanmougam@gmail.com

<sup>2</sup>Professor, Department of Computer Science, Pondicherry University, Puducherry  
aghilaa@gmail.com

## **ABSTRACT**

*Ontologies are currently emerging as representation techniques for overlapping complimentary context domains. A single ontology is no longer enough to support the tasks predicted by a distributed environment like the Semantic Web. Multiple ontologies need to be accessed from several applications. Ontology management is possible through interoperability of semantic data sources. Ontology management includes operations such as mapping, alignment, matching, integration and merging. This work selectively discusses methods of these operations. It provides the researchers a widespread understanding of methods proposed for ontology management operations.*

## **KEYWORDS**

Ontology Mapping, Ontology Alignment, Ontology Matching, Ontology Integration, Ontology Merging, Machine Learning.

## **1. INTRODUCTION**

In the context of the Semantic Web, the available information is organized in ontologies. Ontologies are controlled vocabularies describing objects and relations between them in a formal way, and have a grammar for using the vocabulary terms in order to express something meaningful within a specified domain of interest. Ontologies can be heterogeneous: given two ontologies describing a reference domain, the same real entity can be denoted with different names or it can be defined in different way, may be expressed in different languages, though expressing the same knowledge [2]. Ontology matching discovers a set of relations between entities from two ontologies. This set of relations is called an alignment [12].

Even though the ontologies to be compared/merged are in different representation languages, the mapping process has to be carried out in terms of concepts and properties describing the domain. Mapping between different descriptions of the same object should be discovered, in order to provide ontology integration over Semantic Web sources [3]. Due to the heterogeneity and differences among web sources, data integration becomes a difficult process [14].

The ultimate goal to merge ontologies is to organize and reuse the existing concepts of source ontologies to develop a single terminology. Integration/merging is based on similarity measures that are done through mapping. This work aims at producing a report on methodologies used for ontology mapping, matching, alignment, integration and merging.

Organization of this report is as follow: Section 2 discusses the related work, Section 3 shows the classification model based on different techniques. Section 4 discusses detailed description of each method. Section 5 gives comparison in tabulation format. Section 6 concludes the report.

## 2. RELATED WORK

In the literature, many Ontology Management Tools/systems and related works are available. Among them, Namyoun et al., [21] 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. In addition, comparative results have been given on ontology mapping tools.

Natalya [22] 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., [1] 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., [20] 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., [8] 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., [6] 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., [29] focuses the survey on current state of the art in ontology mapping. Ravi et al., [25] 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.

## 3. CLASSIFICATION MODEL

Generally, the methodology used for ontology management operation may be classified as machine learning, structure based, semantic based and Hybrid. Based on the above classification, the available systems are classified and the Figure 1 clearly portrays that.

Learning is any process by which a system improves performance from experience. A Machine learning is that the computer program is said to *learn* from experience E with respect to some class of tasks T and measure P, if its performance at tasks in T, as measured by P, improves

with experience E. Machine learning is important because some tasks cannot be defined well, except by examples (e.g., recognizing people). Relationships and correlations can be hidden within large amounts of data. Machine Learning may be able to find these relationships. Structure based methodology considers the internal organization or alignment of nodes in taxonomy. Semantic based methodology uses semantics or the science of meaning in language, to produce highly relevant search results. Hybrid includes the combination of more than one technique. The following section briefly explains each methodology with some examples.

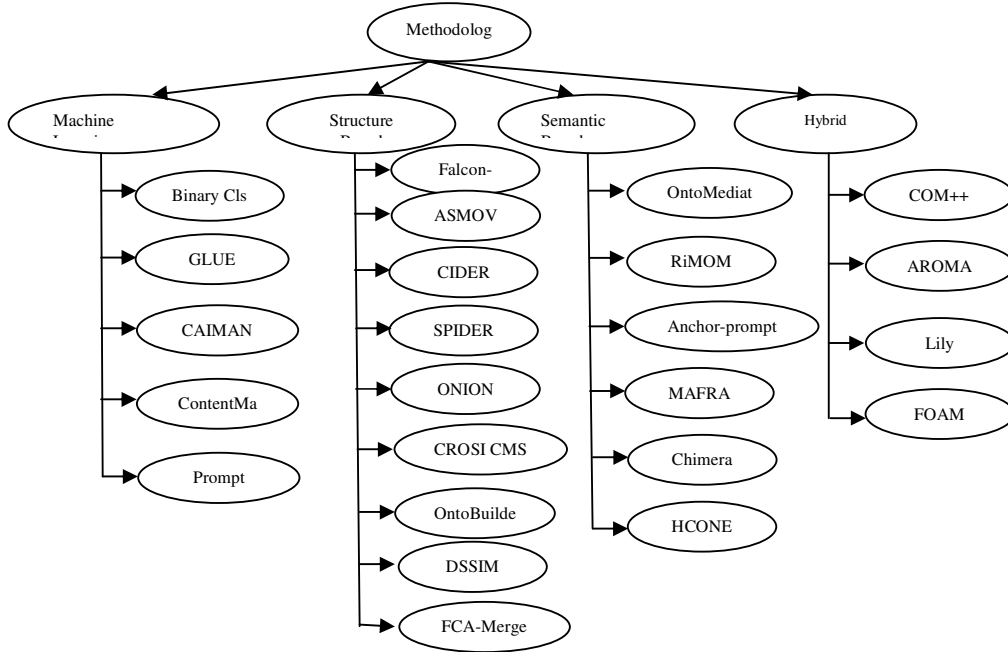


Figure 1. Classification Model based on Methodologies of Ontology Management

## 4. SURVEY

This section elaborately discusses about the existing tools/frameworks based on the classification model. Binary classification [19], GLUE[25], CAIMAN[18], ContentMap[9] and Prompt[23] comes under Machine Learning method. Structured Based method covers Falcon-AO[27], ASMOV[32], CIDER[11], SPIDER[17], ONION[25], CROSI CMS[28], OntoBuilder[5], DSSIM[30] and FCA-Merge[26]. Some of the tools/frameworks in Semantics Based methods are OntoMediate[10], RiMoM[31], Anchor-prompt[23], MAFRA[25], Chimera[7] and HCONE[15]. Hybrid method tools are COMA++[4], AROMA[13], Lily[24] and FOAM[16].

### 4.1. Machine Learning

There exist so many systems using machine-learning technique. Some of them are Binary classification, GLUE, CAIMAN, ContentMap, Prompt and etc.

#### 4.1.1. Binary Classification

Ming Mao et al., [19] dealt ontology mapping problem with machine learning techniques. His approach has five steps: 1. Generated various domain independent features, 2. Randomly generates training and testing set for OAEI benchmark tests. 3. Train a SVM model on training set. 4. Classify testing data on the trained SVM model. 5. Extract mapping results of testing data

using Naïve descendent extraction algorithm. Testing data is evaluated against ground truth. Steps 2 to 6 is repeated 10 times and average evaluation result is used to eliminate bias.

#### **4.1.2. Glue**

Doan et al., [25] developed a system, GLUE, which employs machine learning technique to find mappings. The system consist of three phases: the distribution 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 compute 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.

#### **4.1.3. CAIMAN**

Martin Lacher et al., [18] have proposed a system, CAIMAN that uses machine learning for ontology mapping based text classification. They assumed that community members organize their collection of explicit knowledge (documents) according to their personal categorization scheme. For concept node in the personal ontology, a corresponding node in the community ontology is identified. CAIMAN offers two services to its users: document publication and retrieval of related documents.

#### **4.1.4. ContentMap**

Ernesto Jimenez-Ruiz et al., [9] developed a system called ContentMap A logiC-based ONtology inTEgratioN Tool using MAPpings. The ContentMap evaluates and repairs the logic consequences of merging two independent ontologies using mapping. The method is as: 1. Compute mapping M between O1 and O2 using a mapping algorithm, and filter them using criteria. 2. Compute logic difference and evaluate impact by comparing the entailments holding before and after the integration. 3. Detect unintended entailments and select them. 4. Compute repair plans and execute best one according to the user necessities.

#### **4.1.5. PROMPT**

The PROMPT suite [23] contains of a set of tools that had an important impact in the area of merging, aligning and versioning of ontologies. The suite includes an ontology merging tool (iPROMPT, formerly known as PROMPT), an ontology tool for finding additional points of similarity between ontologies like iPROMPT (Anchor PROMPT), an ontology versioning tool (PROMPT Diff) and a tool for factoring out semantically complete subontologies (PROMPTFactor). PROMPT takes two ontologies as input and guides the user in the creation of a merged ontology as output. First PROMPT creates an initial list of matches based on class names. Then the iterative cycle happens: The user triggers an operation by either selecting one of PROMPT's suggestions from the list or by using an ontology-editing environment to specify the desired operation directly, and PROMPT automatically executes additional changes based on the type of the operation, generates a list of suggestions for the user based on the structure of the ontology around the arguments to the last operation, and determines conflicts that the last operation introduced in the ontology and finds possible solutions for those conflicts.

### **4.2. Structured Based**

Falcon-AO, ASMOV, CIDER, SPIDER, ONION, CROSI CMS, OntoBuilder, DSSIM and FCA-Merge follows structure-based approach. Elaborate description about each system is given below.

#### **4.2.1. Falcon-AO**

Wei Hu et al., [27] have designed a system, Falcon-AO for finding, aligning and learning ontologies, and ultimately for capturing knowledge by an ontology-driven approach. It is an automatic ontology matching system to help enable interoperability between (Semantic) Web applications using different but related ontologies. It consists of five components: the *Repository* to temporarily store the data during the matching process; the *Model Pool* to manipulate ontologies and to construct different models for different matchers; the *Alignment Set* to generate and to evaluate exported alignments; the *Matcher Library* to manage a set of elementary matchers; the *Central Controller* to configure matching strategies and to execute matching operations.

#### **4.2.2. ASMOV**

Yves R. Jean-Mary et al., [32] have developed ASMOV, an automatic ontology matching tool which has been designed in order to facilitate the integration of heterogeneous systems, using their data source ontologies. The current ASMOV implementation produces mappings between concepts, properties, and individuals, including mappings from object to datatype properties and vice versa. The ASMOV algorithm iteratively calculates the similarity between entities for a pair of ontologies by analyzing four features: lexical description (id, label, and comment), external structure (parents and children), internal structure (property restrictions for concepts; types, domains, and ranges for properties; data values for individuals), and individual similarity. The measures obtained by comparing these four features are combined into a single value using a weighted sum.

#### **4.2.3. CIDER**

Jorge Gracia et al., [11] proposed an alignment service called CIDER (Context and Inference based alignER) for semantic similarity measure. It is schema based matching system. It consists of 3-step process. The first step is to extract the ontological context of each involved term. The second step is the computation of similarity for each pair of terms. Comparisons are performed like this: 1. Linguistic similarity between terms, for labels and descriptions is computed. 2. A subsequent computation explores the structural similarity of the terms, exploiting their ontological contexts and vector space modelling is used. 3. The different contributions are weighted, and a final similarity degree is provided. After that, a matrix  $M$  with all similarities is obtained. The final alignment  $A$  is then extracted, finding the highest rated one-to-one relationships among terms, and filtering out the ones that are below the given threshold.

#### **4.2.4. SPIDER**

Marta Sabou et al., [17] given a system SPIDER, that provides alignments with variety of mapping types ( subsumption, disjointness and named relations). This system combines two concrete subsystems. First, the CIDER algorithm to derive equivalence mappings. Second, alignment is extended with non-equivalence mappings derived by Scarlet. CIDER is briefly explained in above section. Scarlet automatically selects and explores online ontologies to discover relations between two given concepts. All relations are obtained by using derivation rules which explore not only direct relations but also relations deduced by applying subsumption reasoning within a given ontology.

#### **4.2.5. ONION**

Mitra et al., [25] have developed a scalable framework (ONtology compositiON) for ontology integration that uses a graph-oriented model for the representation of ontologies. There are two types of ontologies, individual ontologies (source ontologies) and articulation ontologies, which contain the concepts and relationships expressed as articulation rules. The mapping between ontologies is executed by ontology algebra. The architecture of ONION consists of four components namely data layers, viewer, query system and articulation engine. The data layer

contains the wrappers for the external sources and the articulation ontologies that form the semantic bridges between the sources. The viewer is the user interface, which visualizes both the source and the articulation ontologies. The query system translates queries formulated in term of articulation ontology into a query execution plan and executes the query. The articulation engine takes articulation rules proposed by the SKAT and generates sets of articulation rules, which are forwarded to the expert for confirmation.

#### **4.2.6. CROSI CMS**

Yannis Kalfoglou et al., [28] have proposed architecture CROSI CMS, which is structure matching system. The modular architecture employs a multi-strategy system comprising of four modules, namely, *Feature Generation*, *Feature Selection and Processing*, *Aggregator* and *Evaluator*. In this system, different features of the input data are generated and selected to fire off different sorts of feature matchers. The resultant similarity values are compiled by multiple similarity aggregators running in parallel or consecutive order.

#### **4.2.7. OntoBuilder**

Avigdor Gal et al., [5] have proposed tool OntoBuilder for ontology matching. The OntoBuilder project supports the extraction of ontologies from Web search interfaces. It finds the best mapping between two ontologies. It is fully automatic ontology matching system. It contains several unique matching algorithms that can match concepts by their data types, constraints on value assignment, and above all, the ordering of concepts within forms (termed *precedence*).

#### **4.2.8. DSSim**

Miklos Nagy et al., [30] developed a system for ontology alginment (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 informations are 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.

#### **4.2.9. FCA-Merge**

Gerd Stumme et al., [26] developed a framework for ontology merging (FCA-Merge). FCA Merge employs bottom up approach. The process of FCA Merge consists of three steps, namely (i) instances extraction and computing of two formal contexts K1 and K2, (ii) the FCA Merge core algorithm that derives a common context and computes a concept lattice and (iii) the generation of the final merged ontology based on the concept lattice. FCA Merge tool takes as input data the two ontologies and a set D of natural language documents. Instances are extracted from the document in D. The second step comprises the FCA Merge core algorithm that merges two contexts and computes a concept lattice form the merged context using FCA techniques. The final step of deriving the merged ontology from the concept lattice requires human interaction. Based on the pruned concept lattice and the sets of relation names R1 and R2, the ontology engineer creates the concepts and relations of the target ontology.

### **4.3. Semantic Based**

OntoMediate, RiMoM, Anchor-prompt, MAFRA, Chimera, HCONE, etc are semantic based system. Brief discussion about these systems is as follows.

#### **4.3.1. OntoMediate**

Gianluca Correndo et al., [10] proposed a project OntoMediate for the alignment of ontologies and to share mapping results. The system composed of three main subsystems: ontologies and datasets manager; ontology alignment environment; social interaction environment. Ontologies and Datasets Manager part of the system allows users to register/unregister the datasets they intend to share with the community and the ontologies that describe their data vocabulary. Ontology Alignment Environment provides an API for automated ontology alignment tools to be plugged in. Social Interaction Environment functionality allows community members that deal with similar - to socially interact with each other.

#### **4.3.2. RiMOM**

Xiao Zhang et al., [31] proposed a framework RiMOM for ontology matching. The RiMOM consists of six major steps. The input ontologies are loaded into the memory and the ontology graph is constructed in Ontology Preprocessing and Feature Factors Estimation. In Single strategy execution the selected strategies are get to find the alignment independently. Each strategy outputs an alignment result. In Alignment combination phase RiMOM combines the alignment results obtained by the selected strategies. If the two ontologies have high structure similarity factor, RiMOM employs a similarity propagation process to refine the found alignment and to find new alignment according to the structural information. Alignment refinement refines the alignment results from the previous steps.

#### **4.3.3. Anchor-Prompt**

Natalya F. Noy et al [23] have developed a tool in prompt suite called Anchor-prompt for ontology merging. Anchor-PROMPT takes as input a set of pairs of related terms—anchors—from the source ontologies. Either the user identifies the anchors manually or the system generates them automatically. From this set, Anchor-PROMPT produces a set of new pairs of semantically close terms. To do that, Anchor-PROMPT traverses the paths between the anchors in the corresponding ontologies. A path follows the links between classes defined by the hierarchical relations or by slots and their domains and ranges. Anchor-PROMPT then compares the terms along these paths to find similar terms.

#### **4.3.4. MAFRA**

Alexander et al., [25] have proposed a framework for mapping distributed ontologies. MAFRA architecture consists of 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.

#### **4.3.5. Chimera**

Deborah L. McGuinness et al., [7] developed tool called Chimera for ontology merging. Chimaera is aimed to support: (1) merging multiple ontologies and (2) diagnosing (and evolving) ontologies. It facilitates merging by allowing users to upload existing ontologies into a new workspace. Chimaera suggests potential merging candidates based on a number of properties. It generates a name resolution list that may be used as a guide through the merging task.

#### **4.3.6. HCONE**

The goal of HCONE [15] approach is to validate the mapping and to find a minimum set of axioms for the new merged ontology. This approach is based on (a) Capturing the intended informal interpretations of concepts by mapping them to WordNet senses using lexical semantic indexing, and (b) Exploiting the formal semantics of concepts definitions by means of description.

#### **4.4. Hybrid**

This section discusses about the systems that uses combination of more than one technique. Some of the systems are COMA++, AROMA, Lily and FOAM (Framework for Ontology Alignment and Mapping).

##### **4.4.1. COM++**

Erhard Rahm et al., [4] developed COMA++ as a schema and ontology matching tool. The GUI provides access to the five main parts of COMA++, the Repository to persistently store all match-related data, the Model and Mapping Pools to manage schemas, ontologies and mappings in memory, the Match Customizer to configure matchers and match strategies, and the Execution Engine to perform match operations. Automatic match processing is performed in the Execution Engine as three step process, *component identification*, *matcher execution* and *similarity combination*. The obtained mapping can be used as input in the next iteration for further refinement.

##### **4.4.2. AROMA**

Jerome David et al., [13] proposed a method AROMA that is a hybrid, extensional and asymmetric matching approach designed to find out relations between entities from two textual taxonomies. AROMA is divided into three successive main stages: (1) The preprocessing stage allows to represent each entity (classes and properties) by a set of terms, (2) the second stage consists of the discovery of association rules between entities, and finally (3) the post processing stage aims to clean and enhance the alignment.

##### **4.4.3. Lily**

Peng Wang et al., [24] given an ontology mapping system Lily. Lily realized four main functions: (1) Generic Ontology Matching method (GOM) is used for common matching tasks with small size ontologies. (2) Large scale Ontology Matching method (LOM) is used for the matching tasks with large size ontologies. (3) Semantic Ontology matching method (SOM) is used for discovering the semantic relations between ontologies. (4) Ontology mapping debugging is used to improve the alignment results. The alignment process mainly contains three steps: (1) Preprocessing step parses the ontologies. (2) Match computing step uses suitable methods to compute the similarity between elements from different ontologies. (3) Post processing step is responsible for extracting, debugging and evaluating mappings.

##### **4.4.4. FOAM**

Marc Ehrig et al., [16] have proposed a framework for Ontology Alignment and Mapping. This tool has six steps. As the first step feature engineering, the tool selects the ontology for specific domain. Next search step selection: it chooses two entities from the two ontologies to compare (e1, e2). Similarity assessment is the third step, indicates a similarity for a given description (feature) of two entities. Similarity aggregation step aggregates the multiple similarity assessments for one pair of entities into a single measure. To propose the alignment result a threshold and interpretation strategy is used. Finally iteration, i.e. as the similarity of one alignment influences the similarity of neighbouring entity pairs.



## 5. COMPARISON

This section shows the comparative statement in the tabulation format about the tools/frameworks that are discussed in the previous section.

Table 1. Comparative result of tools/frameworks for ontology management

Name	Input & Output	Ontology Management Operation	Representation Language	Algorithm / API Used	Implementation Status	Automation Level	Developed By
<b>Machine Learning</b>							
Binary Classification	Two Ontologies & Evaluation Result	Ontology Mapping	*	SVM Algorithm	Under process	*	*
GLUE	Concepts in Taxonomy & Similarity Measure	Ontology Matching	*	Jaccard Coefficient & MSP	Implemented	Semi-automated	University of Wisconsin, 2005
CAIMAN	Two Ontologies & Mapping result	Ontology Mapping	*	*	Implemented	Semi-automated	Technische University, Munchen, 2001
Content Map	Pre-computed mapping ontologies & Visualizing the mapping results	Ontology Integration Using mapping	OWL 2	*	Implemented	Semi-automated	Universitat Jaume, Spain and University of Oxford, UK, 2008
Prompt	Two DAML Ontologies & Merged Ontology	Ontology Merging and Alignment	OWL	OKBC	Implemented	Automated	Stanford Medical Informatics, Stanford University, 2003
<b>Structure Based</b>							
Falcon-AO	RDFS or Owl Ontology & RDF/XML Format	Ontology Matching	JAVA	Jena	Version 0.7	Semi-automated	Southeast University, China, 2007
ASMOV	OWL-DL	Ontology Alignment	*	Jena ARP	Version 2008	Automated	INFOTECH Soft, Florida, 2007
CIDER	RDF/OWL input & RDF document	Ontology Alignment	JAVA	Alignment API	Version 2008	Semi-automated	University of Zaragoza, Spain, 2007
SPIDER	RDF/OWL ontologies & RDF	Ontology Alignment	JAVA	CIDER & Scarlet	Version 2008	Automated	Knowledge Media Institute,

	document						UK, 2008
ONION	RDF file & Plain text	Ontology merging	JAVA	SKAT	Version 3.5	Semi-automated	Stanford University, 2004
CROSI CMS	OWL Ontologies & Matched result	Ontology matching	JAVA	Jena & JWNL	Version 1.1	Automated	Hewlett Packard Lab, 2005
OntoBuilder	Two Ontologies & produce matching	Ontology Matching	JAVA	*	Version 2.1.5	Automated	Avigdor Gal Technion - Israel Institute of Technology, 2004
DSSim	Two Ontologies & Mapping result	Ontology mapping	OWL	SKOS parser	Version 0.3	Semi-automated	United Kingdom, 2006
FCA-Merge	Two ontologies + DL & produce merged ontology	Ontology merging	*	Concept lattice	*	Automated	University of Karlsruhe Germany, 2001
Semantic Based							
OntoMediate	OWL ontologies & produce result	Ontology alignment and mapping	J2EE & AJAX	Jena API	*	Semi-automated	University of Southampton, UK, 2008
RiMO M	RDF & OWL ontologies and produce result	Ontology Matching	JAVA	OWL API	Version 2008	Automated	IBM China Research Lab, China, 2004
Anchor-Prompt	Pair of anchors & generates new pairs with close similarity	Ontology matching	OntoViz	Protege-2000 plugin		Automated	Stanford University, USA, 2003
MAFRA	RDF files & produce output	Ontology mapping	JAVA & KAON	MAFRA Service Interface API	Version 0.2	Semi-automated	DEI - ISEP - IPP Porto, Portugal, 2003
Chimera	RDF and DAML & merged output	Ontology merging & diagnosing	*	*	*	Semi-automated	Stanford University, USA, 2005
HCON E	Two ontologies & result	Ontology merging & alignment	JAVA	Neo-Classic Description Logic	Version 1.0	Semi-automated	AI-Lab, Greece, 2002
Hybrid							
COM+	XML, OWL files as input & generates	Ontology matching system	*	OWL API	Version 1.0	Semi-automated	Database group Leipzig,

	XML output						UK, 2003
AROMA	OWL ontologies & produce output	Ontology alignment	*	OWL Libraries	*	Semi-automated	INRIA Rhone-Alpes, France, 2006
Lily	RDF ontologies & generates output in text	Ontology Mapping	JAVA	LOM, GOM and SOM	Version 2.0	Automated	Southeast University, China, 2006
FOAM	OWL ontologies & generates aligned text file	Ontology alignment and mapping	JAVA	Google API and KAON2	*	Semi-automated	Institute AIFB, Germany, 2005

\* Could not get the exact data.

## 6. CONCLUSIONS

Ontology management is one of the essential processes in any Semantic Web application. Almost every Semantic Web related application needs part or the whole Ontology management support. This work clearly depicts the importance of ontology management operation keeping in view of 24 tools/frameworks collected. The comparison table given shows the important features of the tools that are being discussed in this work. The result of this survey and analysis, provides comprehensive understanding of ontology management operations and emphasizes the importance for the development of standard system specifically giving due consideration to ontology management operations. This work can be extended with other tools/frameworks for the complete ontology management operations.

## REFERENCES

- [1] Akrivi Katifori and Constantin Halatsis, "Ontology Visualization Methods—A Survey", ACM Computing Surveys, Vol. 39, No. 4, Oct 2007.
- [2] Alfio Ferrara, Davide Lorusso, Giorgos Stamou, Giorgos Stoilos, Vassilis Tzouvaras, Tassos Venetis, "Resolution of conflicts among ontology mappings: a fuzzy approach", Proceedings of 3<sup>rd</sup> International Workshop on Ontology Matching, Oct 2008.
- [3] Alfio Ferrara, Davide Lorusso, Stefano Montanelli, Gaia Varese, "Towards a Benchmark for Instance Matching "7th International Semantic Web Conference (ISWC-2008), vol 431, Oct 2008.
- [4] Aumueller,D.; Do,H.H.; Massmann,S.; Rahm,E.," Schema and ontology matching with COMA++", SIGMOD Conference, 2005.
- [5] Avigdor Gal, Giovanni Modica and Hasan Jamil, " OntoBuilder: Fully Automatic Extraction and Consolidation of Ontologies from Web Sources", Proceedings of 20<sup>th</sup> International Conference on Data Engineering, IEEE Computer Society, 2004.
- [6] Bernhard Haslhofer and Wolfgang Klas, "A Survey of Techniques for Achieving Metadata Interoperability", ACM Computing Surveys, Vol: 42, No:2, Feb 2010.
- [7] Deborah L. McGuinness, Richard Fikes, James Rice and Steve Wilder, "The Chimaera Ontology Environment", American Association for Artificial Intelligence, 2000.
- [8] 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.
- [9] Ernesto Jimenez-Ruiz, Bernado Cuenca Grau, Ian Horrocks and Rafael Berlanga, "Logic-Based Ontology Integration Using ContentMap", In proceedings of BPI 2006, 372.
- [10] Gianluca Correndo and Harith Alani, "Collaborative Ontology Mapping and Data Sharing", 2008.
- [11] J. Gracia and E. Mena. Ontology Matching with CIDER: Evaluation report for the OAEI 2008. In Proc. of 3rd Ontology Matching Workshop (OM'08), at ISWC'08, Karlsruhe, Germany, Oct 2008.

- [12] Jerome David and Jerome Euzenat, "On fixing semantic alignment evaluation measures" Proceedings of the 7th International Semantic Web Conference (ISWC-2008), vol 431, Oct 26, 2008.
- [13] Jerome David, Fabrice Guillet and Henri Briand, "Matching directories and OWL ontologies with AROMA", Proceedings of the 15th ACM international conference on Information and knowledge management, p: 830 – 831, 2006.
- [14] Jitendra kumar, Sanjay kumar malik, nupur prakash, SAM rizvi, "Ontology merging: Role In semantic web", International Conference on Operations Research for a growing nation, Dec 2008.
- [15] Konstantinos Kotis and George A. Vouros, "The HCONE Approach to Ontology Merging", Springer Berlin / Heidelberg, LNCS 3053, p: 137-151, 2004.
- [16] Marc Ehrig and York Sure, "FOAM-Framework for Ontology Alignment and Mapping Results of the Ontology Alignment Evaluation Initiative", Integrating Ontologies Workshop Proceedings K-Cap Conference, p: 72-76, Oct 2005.
- [17] Marta Sabou and Jorge Gracia, "SPIDER: Bringing Non-Equivalence Mappings to OAEI", The 7<sup>th</sup> International Semantic Web Conference, p: 199-205, Oct 2008.
- [18] Martin S. Lacher and Georg Groh, "Facilitating the exchange of explicit knowledge through ontology mappings", American Association for Artificial Intelligence, 2001.
- [19] Ming Mao, Yefei Peng, "Ontology Mapping: As a Binary Classification Problem", 4<sup>th</sup> International Conference on Semantics, Knowledge and Grid, IEEE Computer Society, P: 20-25, 2008.
- [20] Matteo Cristani and Roberta Cuel, "A Survey on Ontology Creation Methodologies", International Journal on Semantic Web & Information Systems 1(2), 49-69, June 2005.
- [21] Namyoun Choi, Il-Yeol Song, and Hyoil Han, "A Survey on Ontology Mapping", SIGMOD, Vol. 35, No. 3, Sep. 2006.
- [22] Natalya F. Noy, "Semantic Integration: A Survey Of Ontology-Based Approaches", SIGMOD, Vol. 33, No. 4, December 2004.
- [23] Noy N.F. and M.A. Muser, "The PROMPT suite: Interactive tools for Ontology merging and mapping", Int. J. Human Comput. Stud, 59: 983–1024, 2003.
- [24] Peng Wang and Baowen Xu, "Lily: Ontology Alignment Results for OAEI 2008", Proceedings of the 7th International Semantic Web Conference on Ontology Matching, p: 167-175, 2008.
- [25] Ravi Loudrusamy and Gopinath Ganapathy, "Feature Analysis of Ontology Mediation Tools", Journal of Computer Science, 4(6), p: 437-446, 2008.
- [26] Stumme, G. and A. Maedche, FCA-Merge: Bottom-up Merging of Ontologies, *In seventh international Conference on Artificial Intelligence (IJCAI'01)*, seattle, WA, USA, 2001, pp: 225-230.
- [27] Wei Hu, Yuzhong Qu, "Falcon-AO: A practical Ontology Matching System", Journal of Web Semantics, Elsevier, p: 237-239, 2008.
- [28] Yannis Kalfoglou and Bo Hu, "CROSI Mapping System (CMS) Results of the 2005 Ontology Alignment Contest", 2005.
- [29] Yannis, K. and S. Marco, "Ontology mapping: The state of the art", The Knowledge Engineering Review, 18: p: 1-31, 2003.
- [30] Ying Wang, "Integrating Uncertainty into Ontology Mapping", Springer Berlin / Heidelberg, LNCS 4825, p: 961-965, 2007.
- [31] Yi Li, Juanzi Li and Jie Tang, "RiMOM – Ontology Alignment with Strategy Selection", supported by NSFC (No. 60409025) and Tsinghua-IBM innovative project, 2006.
- [32] Yves R. Jean-Mary, E. Patrick Shironoshita and Mansur R. Kabuka, "Ontology matching with semantic verification", Journal of Web Semantics, Article in Press, Elsevier, 2009.