INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATION & MANAGEMENT

IJRCM

A Monthly Double-Blind Peer Reviewed (Refereed/Juried) Open Access International e-Journal - Included in the International Serial Directories

Indexed & Listed at:

Ulrich's Periodicals Directory ©, ProQuest, U.S.A., EBSCO Publishing, U.S.A., Cabell's Directories of Publishing Opportunities, U.S.A., Open J-Gage, India [link of the same is duly available at Inflibnet of University Grants Commission (U.G.C.)], Index Copernicus Publishers Panel, Poland with IC Value of 5.09 & number of libraries all around the world.

Circulated all over the world & Google has verified that scholars of more than 4064 Cities in 176 countries/territories are visiting our journal on regular basis.

Ground Floor, Building No. 1041-C-1, Devi Bhawan Bazar, JAGADHRI – 135 003, Yamunanagar, Haryana, INDIA

http://ijrcm.org.in/

CONTENTS

CALL FOR MANUSCRIPTS

We invite unpublished novel, original, empirical and high quality research work pertaining to recent developments & practices in the areas of Computer Science & Applications; Commerce; Business; Finance; Marketing; Human Resource Management; General Management; Banking; Economics; Tourism Administration & Management; Education; Law; Library & Information Science; Defence & Strategic Studies; Electronic Science; Corporate Governance; Industrial Relations; and emerging paradigms in allied subjects like Accounting; Accounting Information Systems; Accounting Theory & Practice; Auditing; Behavioral Accounting; Behavioral Economics; Corporate Finance; Cost Accounting; Econometrics; Economic Development; Economic History; Financial Institutions & Markets; Financial Services; Fiscal Policy; Government & Non Profit Accounting; Industrial Organization; International Economics & Trade; International Finance; Macro Economics; Micro Economics; Rural Economics; Co-operation; Demography: Development Planning; Development Studies; Applied Economics; Development Economics; Business Economics; Monetary Policy; Public Policy Economics; Real Estate; Regional Economics; Political Science; Continuing Education; Labour Welfare; Philosophy; Psychology; Sociology; Tax Accounting; Advertising & Promotion Management; Management Information Systems (MIS); Business Law; Public Responsibility & Ethics; Communication; Direct Marketing; E-Commerce; Global Business; Health Care Administration; Labour Relations & Human Resource Management; Marketing Research; Marketing Theory & Applications; Non-Profit Organizations; Office Administration/Management; Operations Research/Statistics; Organizational Behavior & Theory; Organizational Development; Production/Operations; International Relations; Human Rights & Duties; Public Administration; Population Studies; Purchasing/Materials Management; Retailing; Sales/Selling; Services; Small Business Entrepreneurship; Strategic Management Policy; Technology/Innovation; Tourism & Hospitality; Transportation Distribution; Algorithms; Artificial Intelligence; Compilers & Translation; Computer Aided Design (CAD); Computer Aided Manufacturing; Computer Graphics; Computer Organization & Architecture; Database Structures & Systems; Discrete Structures; Internet; Management Information Systems; Modeling & Simulation; Neural Systems/Neural Networks; Numerical Analysis/Scientific Computing; Object Oriented Programming; Operating Systems; Programming Languages; Robotics; Symbolic & Formal Logic; Web Design and emerging paradigms in allied subjects.

Anybody can submit the **soft copy** of unpublished novel; original; empirical and high quality **research work/manuscript** ***anytime*** in ***M.S. Word format*** after preparing the same as per our **GUIDELINES FOR SUBMISSION**; at our email address i.e. infoijrcm@gmail.com or online by clicking the link **online submission** as given on our website (***FOR ONLINE SUBMISSION, CLICK HERE***).

GUIDELINES FOR SUBMISSION OF MANUSCRIPT

1. **COVERING LETTER FOR SUBMISSION:**

DATED: _____________

THE EDITOR
IJRCM

Subject: **SUBMISSION OF MANUSCRIPT IN THE AREA OF ____.**

(e.g. Finance/Marketing/HRM/General Management/Economics/Psychology/Law/Computer/IT/Education/Engineering/Mathematics/other, please specify)

DEAR SIR/MADAM

Please find my submission of manuscript entitled '____' for possible publication in your journals.

I hereby affirm that the contents of this manuscript are original. Furthermore, it has neither been published elsewhere in any language fully or partly, nor is it under review for publication elsewhere.

I affirm that all the authors have seen and agreed to the submitted version of the manuscript and their inclusion of names as co-authors.

Also, if my/our manuscript is accepted, I/We agree to comply with the formalities as given on the website of the journal & you are free to publish our contribution in any of your journals.

NAME OF CORRESPONDING AUTHOR :
Designation :
Institution/College/University with full address & Pin Code :
Residential address with Pin Code :
Mobile Number (s) with country ISD code :
WhatsApp or Viber is active on your above noted Mobile Number (Yes/No) :
Landline Number (s) with country ISD code :
E-mail Address :
Alternate E-mail Address :
Nationality :

NOTES:
a) The whole manuscript is required to be in ***ONE MS WORD FILE*** only (pdf. version is liable to be rejected without any consideration), which will start from the covering letter, inside the manuscript.
b) The sender is required to mention the following in the **SUBJECT COLUMN** of the mail:
New Manuscript for Review in the area of (Finance/Marketing/HRM/General Management/Economics/Psychology/Law/Computer/IT/ Engineering/Mathematics/other, please specify)
c) There is no need to give any text in the body of mail, except the cases where the author wishes to give any specific message w.r.t. to the manuscript.
d) The total size of the file containing the manuscript is required to be below **500 KB**.
e) Abstract alone will not be considered for review, and the author is required to submit the complete manuscript in the first instance.
f) The journal gives acknowledgement w.r.t. the receipt of every email and in case of non-receipt of acknowledgment from the journal, w.r.t. the submission of manuscript, within two days of submission, the corresponding author is required to demand for the same by sending separate mail to the journal.
g) The author (s) name or details should not appear anywhere on the body of the manuscript, except the covering letter and cover page of the manuscript, in the manner as mentioned in the guidelines.

2. **MANUSCRIPT TITLE**: The title of the paper should be in a 12 point Calibri Font. It should be bold typed, centered and fully capitalised.

3. **AUTHOR NAME (S) & AFFILIATIONS**: The author (s) **full name**, **designation**, **affiliation** (s), **address**, **mobile/landline numbers**, and **email/alternate email address** should be in italic & 11-point Calibri Font. It must be centered underneath the title.

4. **ACKNOWLEDGMENTS**: Acknowledgements can be given to reviewers, funding institutions, etc., if any.

5. **ABSTRACT**: Abstract should be in fully italicized text, not exceeding 250 words. The abstract must be informative and explain the background, aims, methods, results & conclusion in a single para. Abbreviations must be mentioned in full.

6. **JEL CODE**: Provide the appropriate Journal of Economic Literature Classification System code (s). JEL codes are available at **www.aeaweb.org/econlit/jelCodes.php**

7. **KEYWORDS**: JEL Code must be followed by a list of keywords, subject to the maximum of five. These should be arranged in alphabetic order separated by commas and full stops at the end.

8. **MANUSCRIPT**: Manuscript must be in ***BRITISH ENGLISH*** prepared on a standard A4 size ***PORTRAIT SETTING PAPER***. It must be prepared on a single space and single column with 1" margin set for top, bottom, left and right. It should be typed in 8 point Calibri Font with page numbers at the bottom and centre of every page. *It should be free from grammatical, spelling and punctuation errors and must be thoroughly edited*.

9. **HEADINGS**: All the headings should be in a 10 point Calibri Font. These must be bold-faced, aligned left and fully capitalised. Leave a blank line before each heading.

10. **SUB-HEADINGS**: All the sub-headings should be in a 8 point Calibri Font. These must be bold-faced, aligned left and fully capitalised.

11. **MAIN TEXT**: The main text should follow the following sequence:

    INTRODUCTION
    REVIEW OF LITERATURE
    NEED/IMPORTANCE OF THE STUDY
    STATEMENT OF THE PROBLEM
    OBJECTIVES
    HYPOTHESES
    RESEARCH METHODOLOGY
    RESULTS & DISCUSSION
    FINDINGS
    RECOMMENDATIONS/SUGGESTIONS
    CONCLUSIONS
    LIMITATIONS
    SCOPE FOR FURTHER RESEARCH
    REFERENCES
    APPENDIX/ANNEXURE
    It should be in a 8 point Calibri Font, single spaced and justified. The manuscript should preferably not exceed ***5000 WORDS***.

12. **FIGURES & TABLES**: These should be simple, crystal clear, centered, separately numbered & self explained, and **titles must be above the table/figure**. **Sources of data should be mentioned below the table/figure**. It should be ensured that the tables/figures are referred to from the main text.

13. **EQUATIONS/FORMULAE**: These should be consecutively numbered in parentheses, horizontally centered with equation/formulae number placed at the right. The equation editor provided with standard versions of Microsoft Word should be utilized. If any other equation editor is utilized, author must confirm that these equations may be viewed and edited in versions of Microsoft Office that do not have the editor.

14. **ACRONYMS**: These should not be used in the abstract. The use of acronyms is elsewhere is acceptable. Acronyms should be defined on first use in each section: Reserve Bank of India (RBI). Acronyms should be redefined on first use in subsequent sections.

15. **REFERENCES**: The list of all references should be alphabetically arranged. The author (s) should mention only the actually utilised references in the preparation of manuscript and they are supposed to follow **Harvard Style of Referencing**. Also check to make sure that everything that you are including in the reference section is cited in the paper. The author (s) are supposed to follow the references as per the following:

- All works cited in the text (including sources for tables and figures) should be listed alphabetically.
- Use (**ed.**) for one editor, and (**ed.s)** for multiple editors.
- When listing two or more works by one author, use --- (20xx), such as after Kohl (1997), use --- (2001), etc, in chronologically ascending order.
- Indicate (opening and closing) page numbers for articles in journals and for chapters in books.
- The title of books and journals should be in italics. Double quotation marks are used for titles of journal articles, book chapters, dissertations, reports, working papers, unpublished material, etc.
- For titles in a language other than English, provide an English translation in parentheses.
- Headers, footers, endnotes and footnotes may not be used in the document, but in short succinct notes making a specific point, may be placed in number orders following the references.

PLEASE USE THE FOLLOWING FOR STYLE AND PUNCTUATION IN REFERENCES:

BOOKS

- Bowersox, Donald J., Closs, David J., (1996), "Logistical Management." Tata McGraw, Hill, New Delhi.
- Hunker, H.L. and A.J. Wright (1963), "Factors of Industrial Location in Ohio" Ohio State University, Nigeria.

CONTRIBUTIONS TO BOOKS

- Sharma T., Kwatra, G. (2008) Effectiveness of Social Advertising: A Study of Selected Campaigns, Corporate Social Responsibility, Edited by David Crowther & Nicholas Capaldi, Ashgate Research Companion to Corporate Social Responsibility, Chapter 15, pp 287-303.

JOURNAL AND OTHER ARTICLES

- Schemenner, R.W., Huber, J.C. and Cook, R.L. (1987), "Geographic Differences and the Location of New Manufacturing Facilities," Journal of Urban Economics, Vol. 21, No. 1, pp. 83-104.

CONFERENCE PAPERS

- Garg, Sambhav (2011): "Business Ethics" Paper presented at the Annual International Conference for the All India Management Association, New Delhi, India, 19–23

UNPUBLISHED DISSERTATIONS

- Kumar S. (2011): "Customer Value: A Comparative Study of Rural and Urban Customers," Thesis, Kurukshetra University, Kurukshetra.

ONLINE RESOURCES

- Always indicate the date that the source was accessed, as online resources are frequently updated or removed.

WEBSITES

- Garg, Bhavet (2011): Towards a New Natural Gas Policy, Political Weekly, Viewed on January 01, 2012 http://epw.in/user/viewabstract.jsp

A SURVEY ON ONTOLOGY MEDIATION TOOLS

K. VASUMATHI
ASST. PROFESSOR
DKM COLLEGE FOR WOMEN
VELLORE

DR. L.RAVI
HEAD
DEPARTMENT OF COMPUTER SCIENCE
BHARATHIDASAN UNIVERSITY
TRICHIRAPALLI

ABSTRACT
Ontology mediation is enabled through interoperability of semantic data sources. It helps data sharing between heterogeneous knowledgebase and reuse by semantic applications. Ontology mediation includes operations such as, mapping, alignment, matching, merging and integration. After briefly describing these operations, this study selectively discusses set of methods, tools and data integration systems. It provides the researchers a comprehensive understanding of methods and tools intended for ontology mediation.

KEYWORDS
Ontology Mapping, Ontology Alignment, Ontology Merging, Ontology Integration and Ontology Mismatch.

I. INTRODUCTION
In any semantic solution, data is annotated using ontologies. Ontologies are shared specifications and therefore the same ontologies can be used for the annotation of multiple data sources, like web pages, XML documents, relational databases and so on. Their shared terminologies enable a certain degree of interoperability between the data sources using the same ontologies. To enable such an interoperation, mediation is required between the ontologies.

A. TERMINOLOGIES
An ontology mapping M is a declarative specification of the semantic overlap between two ontologies OS and OT. The correspondences between different entities of the two ontologies are typically expressed using some axioms formulated in a specific mapping language. Mapping can be unidirectional or bi-directional. The different phases in the generic mapping process as in[1] is shown in Fig. 1.

B. IMPORT OF ONTOLOGIES
Ontologies can be specified in different languages, which indicate a need to convert them to a common format so that the mapping can be specified. Furthermore, the ontologies need to be imported in the tool, which is used to specify the mapping.

Finding Similarities: Many systems use the match operator to automatically find similarities between ontologies. For any two-source ontology, the match operator returns the similarities between ontologies.

Specifying Mapping: After similarities between ontologies have been found, the mapping between the ontologies needs to be specified.

FIG. 1: MAPPING PROCESS

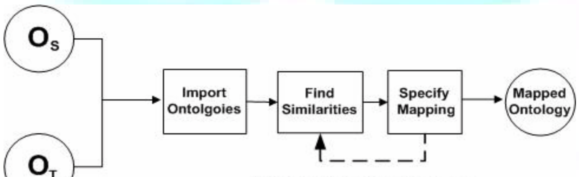


Fig. 1 Mapping Process

The automated or semi-automated discovery of correspondences between two ontologies is called ontology alignment. Ontology alignment is the task of creating links between two original ontologies. Ontology alignment is made, if the sources found to be consistent with each other, but are kept separate or when sources are from the complementary domains. Ontology matching is the process of discovering similarities between two source ontologies. The result of matching operation is a specification of similarities between two ontologies. Ontology matching is carried out through the application of match operator [2].

In ontology merging a new ontology is created which is the union of source ontologies in order to capture all the knowledge from the original ontologies. There are two different approaches in ontology merging. In the first approach, the input of the merging process is a collection of ontologies and the outcome is a one new merged ontology which captures the original ontologies, as given in Fig. 2.

In the second approach the original ontologies are not replaced, but rather a view called bridge ontology is created which imports the original ontologies and specifies the correspondence using bridge axioms as in Fig. 3.

Ontology integration is the process of generating a single ontology in one subject from two or more existing and different ontologies in different subjects. The different subjects of the different ontologies may be related. Some change is expected in a single integrated ontology [3].

FIG. 2: OUTPUT OF MERGING PROCESS (APPROACH 1)

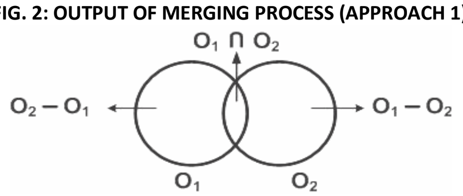
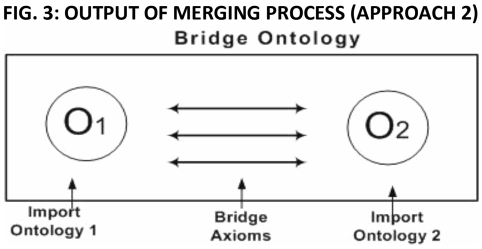


FIG. 3: OUTPUT OF MERGING PROCESS (APPROACH 2)



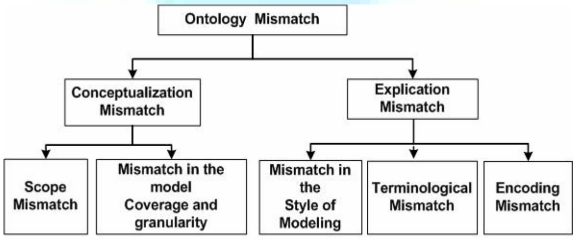
C. ONTOLOGY MISMATCHES

An important issue in the approaches of ontology mediation is the location and specification of the overlap and the mismatches between concepts, relations, and instances in different ontologies are conceptualization mismatches and explication mismatches. The hierarchy of ontology mismatch is given in Fig. 4.

Conceptualization mismatches are mismatches of different conceptualization of the same domain. Conceptualization mismatches fall in two categories; namely a scope mismatch and a mismatch in the model coverage and granularity. A scope mismatch occurs when two classes have some overlap in their extensions (the set of instances), but the extensions are not exactly the same. There is a mismatch in the model coverage and granularity, if there is a difference in (a) the part of domain that is covered by both ontologies (for example, the ontologies of university employees and the students), or (b) the level of detail with which the model is covered (for example, one ontology might have one concept 'person', whereas another ontology distinguishes between young person, middle-aged person and old person).

Explication mismatches are mismatches in the way of specifying a conceptualization. Explication mismatches fall in three categories namely mismatch in the style of modeling, terminology mismatch and encoding mismatch. A mismatch in the style of modeling occurs if either (a) the paradigm used to specify a certain concepts is different (for example, time specified in intervals is different from the time specified in points in time), or (b) the way the concept is described differs (for example, using subclasses versus attributes to distinguish groups of instances).

FIG. 4: HIERARCHY OF ONTOLOGY MISMATCH



A terminological mismatch occurs when two concepts are equivalent, but they are represented using different names (Synonyms) or when the same name is used for different concepts (Homonyms). An encoding mismatch occurs when values in different ontologies are encoded in a different way (for example, distance measure specified in kilometers and miles).

D. A COMPARISON ON ONTOLOGY MEDIATION TOOLS AND SYSTEMS

A specific framework does not exist for comparison of ontology mediation tools [5] nor direct comparison of ontology mediation tools be possible [6]. But the set of criteria to compare the ontology mediation tools is proposed as in [1,3]. The comparison of tools on ontology mediation is made on the following criteria, namely input and output requirements, level of user interaction, ontology language, mapping concepts, automation support, and the level of implementation.

II. MATERIALS AND METHODS

A. RDFT

RDFT[7] is an approach to the integration of product information over the web by exploiting the data model of RDF[8], which is based on direct labeled graphs. This approach assumed that the product catalogs from different organizations are specified in XML (eXtended Markup Language) document. Different organizations use different representations for their product catalogs and hence RDF triples are used to mediate between the different representations.

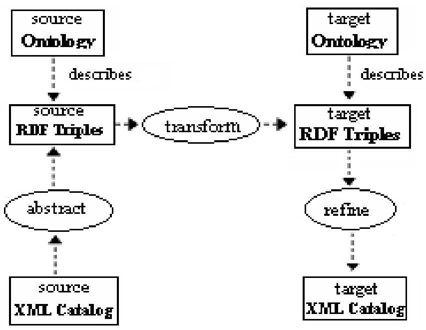
RDF triples consist of a subject, a predicate and an object. Subjects and objects form the nodes of the graph, where as predicates form the edges. An object in the triple can occur as a subject or an object of a different triple.

The approach to the integration of product catalogs is called two layered because the product information itself is represented in XML, whereas the transformation between different representations is done in RDF.

There are three transformation steps. In the first step the XML document whose structure is described by a DTD document (Document Type Definition) or XML schema is a*bstracted* to an RDF graph which is described by ontology in turn, that could be specified using RDF schema [9] ontology language. In the second step the RDF document is *transformed* into a target representation which is also described by ontology. In the third step the target RDF is *refined* to the target XML representation that can be used by the applications at the target vendor.

All the three transformation steps are performed by XML transformation language XSLT (XSL Transformation) [10]. The process of abstraction, transformation and refinement is illustrated in Fig. 5.

FIG. 5: TWO LAYERED INTEGRATION OF XML CATALOGS USING RDF



A mapping Meta-ontology is proposed in for describing the transformation between RDF documents. The mapping of Meta ontology is called RDFT (RDF Transformation) is specified using RDF schema[9] and used to describe the mapping between two RDFS ontologies. A technique for discovering semantic correspondence between different products classification schema based on a **Naïve-Bayes classifier** is described in[12]. The mapping between the different products classifications are represented using the bridges from the RDFT meta-ontology. RDFT can be used to express mappings between the arbitrary ontologies specified in the RDF schema ontology language. And also it is used to specify the transformation between XML document and the RDF representation.

B. OMEN

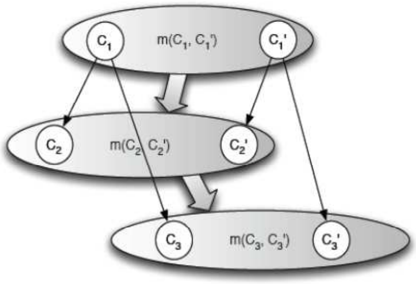
OMEN is an Ontology Mapping ENhancer which uses a set of meta-rules that captures the influence of the ontology structure and the semantics of ontology relations and matches nodes that are neighbors of already matched nodes in the two ontologies. The information sources from the same domain are heterogeneous in nature. To enable interoperation among heterogeneous information sources or to compose information from multiple sources, it is often needed to establish mappings between database schemas or between ontologies. These mappings capture the semantic correspondence between concepts in schemas or ontologies. Many tools are developed for mapping in a semi-automated fashion. There are interactive tools that enable experts to specify the mappings themselves. Once a particular set of mappings is established, the structure of ontologies is analyzed in the neighborhood of the mappings to produce additional mappings.

In OMEN if two properties and their domains match, then it can infer and its ranges can be related. Bayesian Net(BN) is built with the concept mappings. The BN uses a set of meta-rules that express the mapping affected by other related ontology mapping. The initial probability distribution for mapping can be done with existing automatic or semi automatic tools that is used to infer probabilistic distributions for other mappings.

OMEN contains Knowledge model and in this model it expresses the ontologies using the two components classes and properties. *Classes* are concepts in a domain, organized in a subclass–super class hierarchy with multiple inheritances. *Properties* describe attributes of classes and relationships between classes. Properties have one or more domains, which are classes to which the property can be applied; and one or more ranges, which restrict the classes for the values of property.

There are some important notations in the OMEN and in the first notation All concepts from O have no prime (’); all concepts from O' have a prime (’); in the second Upper-case C with or without a subscript is a class; in the third Lower-case q with or without a subscript is a property; in fourth P(C1 θ C2, *x*) indicates that the probability of the match (C1 θ C2) is *x*.

FIG. 6: SUB GRAPHS REPRESENTING CONCEPTS IN ONTOLOGIES O AND O' AND RELATIONS BETWEEN THEM



In the BN graph Nodes are individual pairs of matches between classes or property. Left hand tree are the classes in ontology O. Right hand tree are the classes in ontology O'. Thin arrows are the relationship between the classes in ontology. Solid arrows represent the influences in the BN Graph. The Conditional Probability Tables represent probability distribution in one node in BN-Graph affect the PD in another node.

The nodes in the Bayesian net can be selected as if the node is created for all parings of concepts in two ontologies the number of nodes in the BN-Graph grows with respect to the number of nodes in the source ontology quadratically. Factor that affects the size of the BN-graph is the number of parents that each node has. Thus the maximum number of parent nodes for a single node is restricted to 10. It is selected that the top 5 parents with the maximum a priori probability and the top 5 parents with the minimum a priori probability.

Two types of information’s are needed to run the Bayesians network, The first is the Evidence (obtained from the initial probabilities) describing what we already know with high confidence, and second is the Conditional Probability Tables, describing how the parent nodes influence the children in the BN-graph.

The OMEN algorithm contains the. Input that contains the source ontologies O and O', initial probability distribution for matches. There are few steps in the algorithm the first step contain the condition, if initial probability of a match is above a given threshold, create a node representing the match and mark it as evidence node. Second step Creates nodes in the BN graph representing each pair of concepts (C, C'), that C Є O and C Є O' as a node in the graph and the nodes are within a distance k of an evidence node. Third step creates the edges between the added nodes**.** The fourth step of the algorithm uses the meta-rules to generate CPT’s for the BN*.* Final step of the algorithm is to run the BN. Finally the output is produces as a new set of matches. There are some set of Meta rules in the OMEN they are

- There are two concepts C and C' that match and there is a relationship between C and another concept C_1 in the ontology O and a relationship between C' and C_1' in the ontology O'. If the two relationships match. Then, the probability of the match between C_1 and C_1' is increased and if they don’t match then the probability of the match between C_1 and C_1' is decreased.
- There are two properties that match and each of them has a single range. Then, the probability of a match between the classes that represent the ranges is increased.
- There are two properties that match and the first property has a range that is a union of the classes C_1 and C_2. The other property has a single range corresponding to C'. Then, the probability that C_1 is a specialization of C' can be increased. Analogously, the probability that C_2 is a specialization of C' can be increased.
- There are mappings between super concepts of two certain concepts, each belonging to a different ontology, and all the siblings. Then, the probability of a match between the remaining concepts is increased.

In OMEN the system probabilistic influences are combined with some conditions that is if a node in a Bayesian Network has two parents, the conditional probability tables are combined for the child using the assumption that the two parents are independent. i.e. $P(N \mid P_1,P_2)=P(N \mid P_1)\ P(N \mid P_2)$. When the match of two pairs of parents influences each other, this assumption is not true. The new mappings can be inferred by OMEN using RDF and RDF schema for expressing two ontologies.

***C.* S-MATCH**

Semantic Matching is also known as S-Match[13] and it is an approach for matching classification hierarchies. Each term in the classification hierarchy describes a set of documents. Semantic Matching is also seen as an implementation of the Match operator for purely tree-structures ontologies. The Match is defined as an operator that takes two graph-like structures (e.g. database schemas or ontologies) and produces a mapping between elements of the two graphs that correspond semantically to each other. Semantic Matching approach performs matching based on the nodes and the edges between the nodes in a graph. Semantic Matching has been mostly developed and tested for the task of matching hierarchies. Hierarchies are tree-structured graphs in which each node has only one parent. A property of hierarchies is that there is only one type of relationship, which is a *more-specific-term* relation which subsumes the *subclass-of* relationship.

The authors of [13] have argued that almost all earlier approaches to schema and ontology matching have been *syntactic* matching approaches, as opposed to *semantic* matching. In syntactic matching, the labels and sometimes the syntactical structure of the graph are matched and typically some similarity coefficient

[0,1] is obtained, which indicates the similarity between the two nodes. Semantic Matching computes a set-based relation between the nodes, taking into account the meaning of each node.

The possible relations returned by the Semantic Matching algorithm are equality (=), overlap (∩), mismatch ($\perp$), more general ($\subseteq$) or more specific ($\supseteq$). The correspondence of the symbols with set theory is not a coincidence, since each concept in the classification hierarchies represents a set of documents. In semantic matching algorithm for graph matching it contains two levels of granularity for matching, and this matching is distinguished as, element level matching and structure-level matching. At the element level, it is concerned with individual nodes. The authors distinguish techniques with weak semantics and techniques with strong semantics.

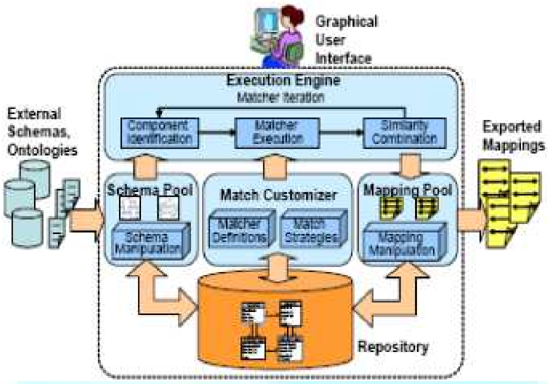
Element-level matching with strong semantics is done using thesauri, which typically contain synonym and hypernym relations between terms. These relations can be used to find semantic relations between nodes in the graphs. In the next phase, the structure-level matching, the matching problem, i.e. the two graphs together with the mapping query are translated into a propositional formula and then checked for validity. A mapping query is a pair of nodes and a semantic relationship between the pair of nodes. If the propositional sentence is valid, then the semantic relationship between the two nodes in the query holds and thus can be added to the mapping result. A potential problem with the algorithm is that the propositional satisfiability check which is known to have nondeterministic polynomial complexity has to be performed for every pair of nodes from the two graphs. This algorithm does not scale for large graphs.

D. COMA++

COMA++ is built on top of COMA by elaborating in more detail the alignment reuse operation. Also it provides a more efficient implementation of the COMA algorithms and a graphical user interface. COMA++ can be used as a platform to evaluate different match algorithms. In a comprehensive evaluation, we achieved high quality even on large real-world schemas and ontologies. Due to the highly optimized implementation of the matchers, in large matching problem COMA++ shows faster execution time than COMA. Without providing domain specific taxonomies or synonyms, COMA++ can solve many problems.

Figure 7 shows the underlying architecture of COMA++[14]. The GUI of COMA++ provides access to the five main parts. 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.

FIG. 7: ARCHITECTURE OF COMA++



To maximize the potential for reuse [15,16] the *Repository* centrally stores various types of data related to match processing, in particular imported schemas and ontologies, produced mappings, auxiliary information such as domain-specific taxonomies and synonym tables, and the definition and configuration of the matchers. A generic data model is used to implement in a relational DBMS to uniformly store the different kinds of schemas and ontologies and mappings between them.

Models are uniformly represented by directed graphs as the internal format for matching. The *Model Pool* provides different functions to import external schemas and ontologies, and to load and save them from/to the repository. Formats supported by COMA++ include XSD, XML Data Reduced (XDR), OWL, and relational schemas. From the Model Pool, two arbitrary models can be selected to start a match operation. The model pool also maintains all generated mappings and offers various functions to further manipulate them.

Automatic match processing is performed in the *Execution Engine* in the form of match iterations, which are the building blocks for match strategies such as fragment based matching. As indicated in Figure 7, match iterations take place in three steps, *component identification* to determine the relevant schema components for matching, *matcher execution* applying multiple matchers to compute component similarities, and *similarity combination* to combine matcher-specific similarities and derive the correspondences between the components. The obtained mapping can be used as input in the next iteration for further refinement. Each iteration can be individually configured using the alternatives supported by the *Match Customizer*, i.e. the types of components to be considered, the matchers for similarity computation, and the strategies for similarity combination.

COMA++ supports various methods to determine the components of a schema, such as nodes, paths, and fragments, and to determine the constituents of a single component, such as its name tokens, child nodes, etc., which can be considered to estimate the similarity between two components. Multiple matchers can be selected from the *Matcher Library* to compute the similarity between the identified components, resulting in a similarity cube. More than 15 matchers exploiting different kinds of schema and auxiliary information are available. COMA++ then employs the combination scheme developed in COMA with corresponding strategies for the sub-steps *aggregation*, *direction*, and *selection* [16] to derive a match result from the similarity cube. The obtained mapping is a set of correspondences specifying the matching components between two input models. Each pair of matching components is captured in a single correspondence, i.e. a 1:1 match.

The COMA++ has two higher-level match strategies to address complex match problems, in particular *Fragment-Based Matching and Reuse oriented Matching*. To cope with large schemas, COMA++ implements the fragment-based match processing framework proposed in [17]. Following the divide-and-conquer idea, it decomposes a large match problem into smaller sub problems by matching at the level of schema fragments. With the reduced problem size, it aims not only at better execution time but also at better match quality compared to schema-level matching. The framework encompasses two matching phases *Identifying Similar Fragments* and *Match Fragments* in the first step Depending on a specified fragment type, the step determines the fragments from the input schemas and compares them to identify the most similar ones worth to be fully matched later. In the Match fragment strategy each pair of similar fragments represents an individual match problem, which is solved in a single match operation to identify correspondences between their components. The result is a set of mappings containing correspondences between fragment components, which are then merged into a global match result.

In reuse oriented matching the reuse of existing schemas is addressed in [15], focusing on learning and using component statistics in a corpus of schemas for matching. In contrast, it pursues the reuse of previously determined match results. The main mechanism for this approach is the Match Compose operation [16], which performs a join-like operation on a mapping path consisting of two or more mappings, such as A-B, B-C, and C-D, successively sharing a common schema, to derive a new mapping between A and D. Thus different matchers and match strategies in COMA++ can be used to match between schemas and ontologies from several domains. Thereby, various interaction possibilities to influence the match process will be demonstrated, such as configuration of matchers and match strategies, step-by-step execution of match operations, verification and further manipulation of match results.

E. FALCON-AO

Falcon-AO is an automatic tool for aligning ontologies. The term Falcon-AO means **F**inding **A**ligning, **L**earning Ontologies, **C**apturing knowledge, **O**ntology-driven approach, and Automatic tool for **A**ligning **O**ntologies. There are two matchers integrated in Falcon-AO: one is a matcher based on linguistic matching for ontologies, called LMO; the other is a matcher based on graph matching for ontologies, called GMO. In Falcon-AO, GMO takes the alignments generated by LMO as external input and outputs additional alignments. Reliable alignments are gained through LMO as well as GMO according to the concept of reliability. The reliability is obtained by observing the linguistic comparability and structural comparability of the two ontologies being compared.

Falcon – AO Provides enabling technologies for finding, aligning and learning ontologies, and ultimately for capturing knowledge by an ontology-driven approach. It is still under development as a component of Falcon; Falcon- AO is an automatic tool for aligning ontologies. It is dedicated to aligning web ontologies expressed in OWL DL [18]. There are two matchers integrated in current version of Falcon AO (version 0.3). One is a matcher based on linguistic matching for ontologies, called LMO, and the other one is a matcher based on graph matching for ontologies, called GMO.

Linguistic matching plays an important role in matching process. Generally, linguistic similarity between two entities relies on their names, labels, comments and some other descriptions. **LMO** combines two different approaches to gain linguistic similarities: one is based on lexical comparison; the other is based on statistic analysis. In lexical comparison, the edit distances [19] is calculated between the names of two entities and use the following function to capture the string similarity (denoted by *SS*):

$$SS = 1/e^{\frac{ed}{|s1.len + s2.len - ed|}} \qquad (1)$$

Where *ed* denotes the edit distance between *s1* and *s2; s1.len* and *s2.len* denote the length of the input strings *s1* and *s2*, respectively. In statistic analysis, the VSM (Vector Space Model) [19] algorithm is used for implementation. Given a collection of documents, the *N* denotes the number of unique terms in the collection. In VSM, each document is represented as a vector in an *N*-dimensional space. The components of the vector are the term weights assigned to that document by the term weighting function for each of the *N* unique terms in the collection. The term weighting functions are defined as follows:

$$\text{Term Weighting} = TF * IDF \qquad (2)$$

$$TF = \frac{t}{T} \qquad (3)$$

$$IDF = \frac{1}{2} * (1 + \log_2 \frac{D}{d}) \qquad (4)$$

In equation (3), *t* denotes the number of times where one term occurs in a given document and *T* denotes the maximum number of times. In equation (4), *D* denotes the number of documents in collection and *d* denotes the number of documents where the given term occurs at least once. The cosine similarity between documents is gained (denoted by DS) by taking the vectors' dot product:

$$DS = N.N^t \qquad (5)$$

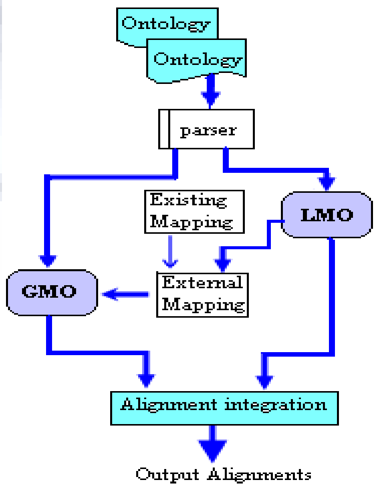
The above methods will take effect in ontology matching. In our implementation, by the combination of the equations the final equation for the linguistic similarity can be calculated.

$$\text{Linguistic Similarity} = 0.8 * DS + 0.2 * SS \qquad (6)$$

Another important component in Falcon-AO is **GMO**, which is based on a graph matching approach for ontologies. It uses directed bipartite graphs to represent ontologies and measures the structural similarity between graphs by a new measurement. Similarity of two entities from two ontologies comes from the accumulation of similarities of involved statements (triples) taking the two entities as the same role (subject, predicate, object) in the triples, while the similarity of two statements comes from the accumulation of similarities of involved entities of the same role in the two statements being compared. GMO takes a set of matched entity pairs, which are typically found previously by other approaches, as external mapping input in the matching process, and outputs additional matched entity pairs by comparing the structural similarity. The GMO are irreplaceable when there was little gain from lexical comparison. The GMO can also be integrated with other matchers. While using GMO approach to align ontologies, there should be another component to evaluate reliability of alignments generated by GMO. LMO and GMO are integrated in Falcon-AO. Alignments output by Falcon-AO come from the integration of alignments generated by LMO and GMO. The architecture of Falcon-AO is shown in Fig 8.

Due to heterogeneous ways in expressing semantics and the inference capability brought from ontology languages, two ontologies being matched may need to be coordinated by removing some redundant axioms from it or adding some inferred axioms. So coordination actions should be taken before using GMO approach. Several coordination rules are integrated in Falcon-AO. The *Parser* component is based on Jena [20] has the functionality of coordinating ontology models. The given external mapping as input, GMO can find additional mapping.

FIG. 8: SYSTEM ARCHITECTURE



The external mapping is made of two parts: one is the existing mapping pre-assigned by the system; the other comes from another matcher. The existing mapping is the mapping between built-in vocabularies of web ontology languages, data types, data literals and URIs used in both ontologies. And in Falcon-AO the alignments generated by LMO as the other part of external mapping. Entities involved in the alignments generated by LMO are set to be external entities and GMO will just output mapping between internal entities.

When the alignments generated by LMO and GMO are obtained, Falcon-AO will integrate the alignments by observing the linguistic comparability and structural comparability, following some rules. The first rule is that the linguistic similarity is more reliable than structural similarity, and that the alignments generated by LMO are always accepted by Falcon-AO. The second rule is that when the linguistic comparability is high and the structural comparability is low, only alignments generated by GMO with high similarity are reliable and accepted by Falcon-AO. And the final rule is that if the linguistic comparability is low, all of the alignments generated by GMO are accepted by Falcon-AO. In this case the information is not enough to measure the alignments can only be an assumption.

Falcon-AO is implemented in Java. The first implementation process is to Input two ontologies and parse them. Then Run LMO and obtain matched entity pairs. After matching the entity pair linguistic comparability and structural comparability are calculated. if the linguistic comparability is below a very low threshold (e.g.0.01) and the structural comparability of them is also low, we take that these ontologies are quite different and Falcon-AO exits with no alignment. External entities of the ontologies are set according to the matched entity pairs generated by LMO. Input matched entity pairs generated by LMO into GMO and form external mapping for GMO. In the current version of Falcon-AO, all the individuals of ontologies are specified as external entities and their similarities are computed by LMO. Then by running GMO matched entity pairs are obtained. Then integrate the alignments generated by LMO and GMO following the rules described above. Finally Exit with alignments as output.

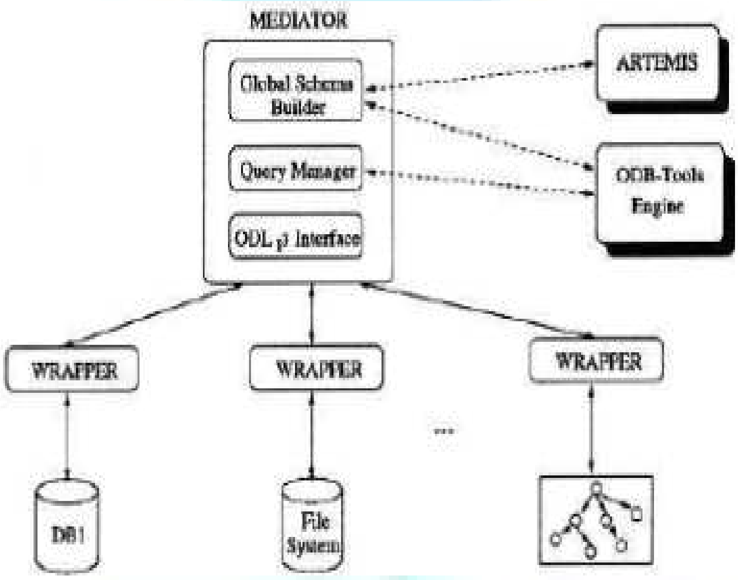
While aligning real ontologies, linguistic matching plays an important role in matching process. Therefore, GMO is integrated with LMO in Falcon-AO. Ontology matching is an important way to establish interoperability among (Semantic) Web applications using different but related ontologies. A practical system for ontology matching called Falcon-AO is implemented. And Falcon-AO (version 0.7) performs quite well and balancing on most of tasks.

G. MOIMS

MOMIS (**M**ediator envir**O**nment for **M**ultiple **I**nformation **S**ources) approach [21, 22] is an approach to the integration of heterogeneous data sources using a global ontology, which is the result of a merge of the local data schemas. The goal of MOMIS is to give the user a global virtual view [23] of the information coming from heterogeneous information sources. MOMIS creates a global mediation schema (ontology) for the structured and semi-structured heterogeneous data sources, in order to provide to the user a uniform query interface to these sources. The first step in the creation of the global mediation schema is the creation of the Common Thesaurus from the disparate data sources. First a wrapper is created for each data source in the ODL_I^3 languages. ODL_I^3 is an object-oriented language with an underlying Description Logic language OLCD, which enables making inferences (e.g. subsumption) about the classes expressed in that language.

Using the disparate schemas, a Common Thesaurus is created, which describes intra and inter-schema knowledge about ODL_I^3 classes and attributes of source schemas. The Common Thesaurus is built in an incremental process in which relationships (between classes) are added based on the structure of the source schemas, lexical properties of the source classes and attributes can be used to identify possible synonyms), relationships supplied by the designer, and relationships inferred by the inference engine.

FIG. 9: ARCHITECTURE OF THE MOMIS SYSTEM [22].



Once the Common Thesaurus has been created, a tree of affinity clusters is created, in which concepts are clustered based on their (name and structural) affinity. The name affinity coefficient is calculated based on the terminological relationships between two classes. The structural affinity coefficient between two classes is calculated based on the level of matching of attribute relationships in the Common Thesaurus. The sum of these two coefficients is the global affinity coefficient, which is used to construct the affinity tree, in which concepts with a high affinity are clustered together. For each cluster in the affinity tree, a global class is (interactively) created. For each global class a mapping (expressed in ODL) is maintained to all the source classes.

The architecture in Fig 9 shows the main tools used to support the overall architecture. A disadvantage is that there is no integrated tool environment. Any data source can be connected to the architecture, as long as an ODL_I^3 wrapper is created.

H. INFOMIX

INFOMIX[24,25,26] is a system that supports information integration by utilizing advanced reasoning capabilities. The INFOMIX system is built in cooperation with RODAN systems [27] which is a commercial database management system developer. It uses the DLV reasoning system [28] for the reasoning tasks. The DLV system is a disjunctive data log reasoning system and has been developed independently from INFOMIX. However, INFOMIX uses the DLV system for solving its reasoning tasks.

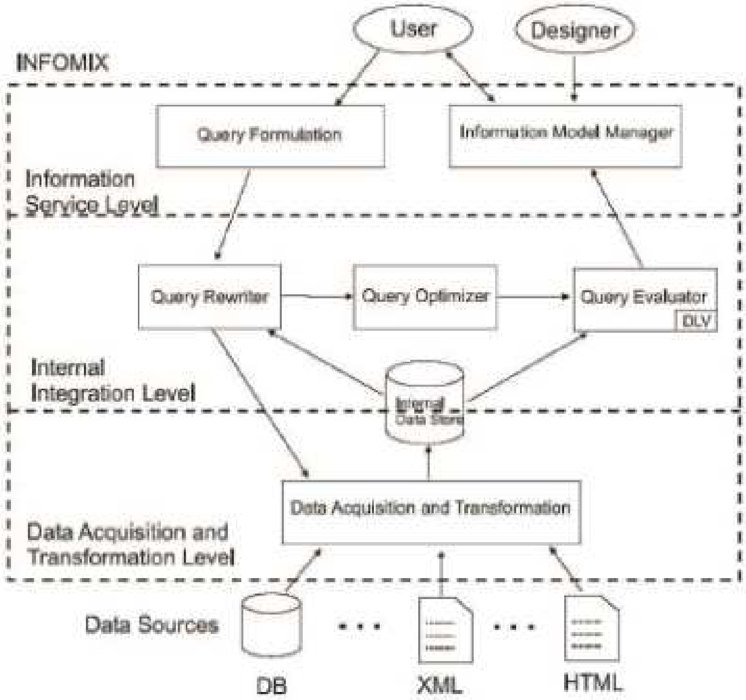
The INFOMIX architecture is depicted in Fig. 10. It is divided into three levels and supports two modes and they are design mode and query mode. In design mode the global schema, the source schema and the mapping between them are specified. Also, the wrappers for the data sources are created or imported. The data sources consist of relational and XML data. In query mode query answering facilities are provided at run time; including data acquisition integration, answer computation and presentation to the user. In both modes, the INFOMIX system is divided into three levels and they are Information Service Level, Internal Integration Level and Data Acquisition and Transformation Level.

The **Information Service Level** is a direct interface to the user is provided at run time and to the designer at design time. This level deals with global data and provides the interfaces that are necessary. It comprises two modules the Information Model Manager and the Query Formulation module The *Information Model Manager* handles the definition of the global schema and the local schemas, as well as the mapping. User-friendly interfaces for these tasks, including schema browsers, are provided. Automatic support for the verification of coherency, redundancy and adequacy of the application specification is given. Finally, this module presents query results in a suitable form to the user. The *Query Formulation module* provides a graphical, user-friendly interface for query formulation over the global schema and query validation facilities. The query validation facilities check the interactions between the user query and global integrity constraints to guarantee that query answering is always decidable.

The Internal Integration Level is based on computational logic and deductive database technology. It is composed by three modules Query Rewriter, Query Optimizer, Query Evaluator**.** The *Query Rewriter* reformulates the user query according to global integrity constraints. It makes use of a sub module to verify data consistency. This sub module exploits the mapping and unfolds the user query over the source relations and activates the corresponding wrappers to retrieve relevant data. Afterwards, the sub module checks whether there are integrity constraint violations.

If no violations occur, the reformulation produced by the rewriter is a simple (disjunction free) Datalog program. Otherwise, a suitable disjunctive Datalog program is generated that performs automatic repair of data, in a way such that cautious answers to this program evaluated over the data sources correspond to the certain answers to the query.

FIG. 10: ARCHITECTURE OF THE INFOMIX SYSTEM [24].



The *Query Optimizer* provides several optimization strategies which enhance the efficiency of the system. In particular, the module exploits some focusing techniques which are able to isolate the portion of the source database that is relevant to answer a user query.

The Data Acquisition and Transformation Level provides access to external data sources. INFOMIX has an architecture which allows for the integration of heterogeneous types of data sources. The primary types of data sources are relational, XML, HTML, and object-oriented data sources. However, it is claimed that arbitrary other types of data sources can be incorporated easily. All data sources are conceptually transformed into a uniform source data format, which is a fragment of XML Schema. Data encoded in this uniform source data format can be browsed. The acquisition and transformation of data is done by wrappers. A query plan for executing suitable wrappers is generated, which load data into the Internal Data Store. Constants are pushed to the query wrappers whenever it is possible in order to reduce the amount of data retrieved. Currently, INFOMIX offers three classes of wrappers, which provide different levels of support for query formulation and wrapper code generation. *Code wrappers* are basically a definition of an API and the code implementing it. The internals and characteristics of code wrappers are therefore inaccessible to INFOMIX. *Query wrappers* propagate queries to external data sources and treat the result as a logical data source. *Visual wrappers* support interactive development of wrappers at design time. Currently, there is support for developing LiXto wrappers and pipes as well as for Rodans Data Extractor.

APPROACH	INPUT	OUTPUT	USER INTERACTION	ONTOLOGY LANGUAGE	MAPPING LANGUAGE	AUTOMATION SUPPORT	IMPLEMENTATION
				METHODS & TOOLS			
MAFRA	Two Ontologies	Mapping of Two Ontologies	Semantic Bridging Modules and Graphical Interface	RDFS	Semantic Bridging Ontologies (SBO)	Lexical and Structural Matching & Semi-automatic Creation of Mappings	2 Prototype Implementation
RDFT				RDFS	RDFTA	Discovery of Similarities Based on Instance Data	Research Prototype
PROMPT	Two Ontologies	Merged Ontology	User Accepts, Rejects or Alters System's Suggestion	RDFS , OWL	Heuristic Bases Analyser	Name and Structural Matching	Version 2.1.1
GLUE	Two Taxonomies with Data Instances	Set of Pairs of Similar Concepts	User Defined Mappings, Similarity Measure & Analysing System's Match Suggestion	Taxonomies	Similarity Measures	Multi-strategy Machine Learning Approach	Research Prototype
S-Match	-	-	-	DAGs	Ser-based (Equal, Disjoint, Subset, Superset)	Matching Based on Synsets from Thesauri using SAT Solver	1st Prototype
OntoMap	-	-	-	Proprietary Language Similar to OWL Lite	OntoMapO	-	Prototype under Development Since 2001
RDFDiff	-	-	-	RDF	Changed, Added, Deleted	Detected Automatically	Research Prototype
OMEN	-	-	-	Proprietary Language Similar to RDF Schema	Bayesian Network with individual Pairs of Matches as nodes	Mappings that depend on other Mapping can be Inferred Automatically	1st Research Prototype
WSMX Mediation	-	-	-	Flora (WSML Based)	SEKT Abstract Mapping Language	String Matching, Wordnet Similarity & Real-time User Feedback	Research Prototype
DOME Mediation	-	-	-	WSML	SKET Abstract Mapping Language	No Specific Automation Support	Research Prototype
				INTEGRATION SYSTEMS			
InfoSleuth				OKBC	Wrappers	-	Project Prototype
ONION	Terms in Two Ontologies	Sets of Articulation Rules	Human Expert Chooses or Deletes or Modifies using GUI Tool	Directed Labeled Graphs and Horn Clauses	Articulation Rules	Term & Structural Matching Using SKAT Annotation Tool	Research Prototype for Unification of Heterogeneous Ontologies
OBSERVER	Two Ontologies	Inter-relationship Manager	Query Based Interface	Description Logics (CLASSIC)	Extended Relational Algebra for Mapping Ontology – DB & DL	Query Processing	Research Prototype for Access of Heterogeneous Data Source
INFOMIX	-	-	-	Set of Logical Implications in Disjunctive Datalog	Disjunctive Datalog	-	1st Project
AutoMed	-	-	-	HDM	HDM	Bidirectional Similarity Degrees	LSDI Project ISPIDER Project RoDEX Project

Comparison of Methods & Tools for Ontology Languages, Mapping Languages, Automation Support and Implementation

CONCLUSION

Ontology mediation is enabled through interoperability of semantic data sources. It helps data sharing between heterogeneous knowledgebase and reuse by semantic applications. Ontology mediation includes operations such as, mapping, alignment, matching, merging and integration. After briefly describing these operations, this study selectively discusses set of methods, tools and data integration systems. It provides the researchers a comprehensive understanding of methods and tools intended for ontology mediation.

REFERENCES

1. Erhard, R. and A.B Philip. 2001. A Survey of approaches to automatic schema matching, VLDB J. Very Large Databases, 10: 3334-350.
2. Namyoun, C., 1.1 Yeol Song and H.Hyoil, 2006. A survey on ontology mapping, SIGMOD Record, Vol 35, No 3, Sep 2006, pp: 34-41.
3. Kelvin, M., Combining and relating Ontologies: An Analysis of Problems and Solutions, In workshop on Ontologies and information sharing, IJCAI'01, August 4-5, 2001, Seattle, USA.
4. Yannis. K, and S. Marco, 2003, Ontology Mapping: The State of the art, The Knowledge Engineering Review, 18: 1-31.

5. Natalya. F.N, and A.M.Mark, 2002, Evaluating Ontology Mapping tools: Requirements and experience, Procedings of the workshop on evaluation of ontology at EKAW'02(EOEN 2002), Siguenza, Spain.
6. Borys Omelayenko and Dieter Fensel. A two-layered integration approach for product information in B2B e-commerce. In Proceedings of the Second Intenational Conference on Electronic Commerce and Web Technologies (EC WEB-2001), Munich, Germany, 2001. Springer-Verlag.
7. Ora Lassila and Ralph R. Swick. Resource description framework (RDF) model and syntax specification. W3c recommendation, W3C, 1999. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222.
8. Dan Brickley and Ramanathan V. Guha. RDF vocabulary description language 1.0: RDF schema. Recommendation 10 February 2004, W3C, 2004. Available from http://www.w3.org/TR/rdf-schema/.
9. James Clark. XSL transformations (XSLT) version 1.0. Recommendation 16 November 1999, W3C, 1999.
10. Borys Omelayenko. RDFT: A mapping meta-ontology for business integration. In Proceedings of the Workshop on Knowledge Transformation for the Semantic Web (KTSW 2002) at the 15-th European Conference on Artificial Intelligence, pages 76.83, Lyon, France, 2002.
11. Borys Omelayenko. Integrating vocabularies: Discovering and representing vocabulary maps. In Proceedings of the First International Semantic Web Conference (ISWC2002), Sardinia, Italy, 2002.
12. Fausto Giunchiglia and Pavel Shvaiko. Semantic matching. *The Knowledge Engineering Review*, 18(3):265.280, 2004.
13. Madhavan, J., P.A. Bernstein, A.H. Doan, A.Y. Halevy: *CorpusbasedSchema Matching.* Int. Conf. of Data Engineering (ICDE) 2005.
14. Do, H.H., E. Rahm: *COMA – A System for Flexible Combination of Match Algorithms.* VLDB 2002.
15. David Aumueller, Hong-Hai Do, Sabine Massmann, Erhard Rahm Schema and Ontology Matching with COMA++, *SIGMOD*, Baltimore, Maryland, USA, *2005*.
16. Rahm, E., H.H. Do, S. Massmann: *Matching Large XML Schemas.* SIGMOD Record 33(4), 2004.
17. P. Patel-Schneider, P. Hayes, I. Horrocks (eds.). OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation 10 February 2004. Latest version is available at http://www.w3.org/TR/owl-semantics/
18. V. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics - Doklady 10 (1966) 707-710
19. http://jena.sourceforge.net/
20. Sonia Bergamaschi, Silvana Castano, and Maurizio Vincini. Semantic integration of semi structured and structured data sources. SIGMOD Record Special Issue on Semantic Interoperability in Global Information, 28(1), March 1999.
21. Sonia Bergamaschi, Silvana Castano, Maurizio Vincini, and Domenico Beneventano. Semantic integration of heterogeneous information sources. Specia Issue on Intelligent Information Integration, Data & Knowledge Engineering, 36(1):215.249, 2001.
22. Richard Hull. Managing semantic heterogeneity in databases: A theoretical perspective. In ACM Symposium on Principles of Database Systems, pages 51.61, Tuscon, Arizona, USA, 1997.
23. N. Leone, G. Gottlob, R. Rosati, G. Greco, G. Ianni, V. Lio, V. Terracina, T. Eiter, W. Faber, M. Fink, D. Lembo, M. Lenzerini, M. Ruzzi, E. Kalka, B. Nowicki, and W. Staniszkis. The INFOMIX system for advanced integration of incomplete and inconsistent data. In Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 915. 917, 2005.
24. N. Leone, G. Gottlob, R. Rosati, G. Greco, G. Ianni, V. Lio, V. Terracina, T. Eiter, W. Faber, M. Fink, D. Lembo, M. Lenzerini, M. Ruzzi, E. Kalka, B. Nowicki, andW. Staniszkis. Data integration by logic programming: The INFOMIX system. In Proceedings of the Eighth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR), 2005.
25. Domenico Lembo, Maurizio Lenzerini, and Ricardo Rosati. Functional specification of the infomix system. Technical Report D2.1, University of Rome, 2002.
26. The web site of RODAN systems can be found at http://www.rodan.pl/en/.
27. N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, and F. Scarcello. The DLV system for knowledge representation and reasoning. In ACM Transactions on Computational Logic, 2005.

REQUEST FOR FEEDBACK

Dear Readers

At the very outset, International Journal of Research in Computer Application & Management (IJRCM) acknowledges & appreciates your efforts in showing interest in our present issue under your kind perusal.

I would like to request you tosupply your critical comments and suggestions about the material published in this issue as well as on the journal as a whole, on our E-mail**infoijrcm@gmail.com** for further improvements in the interest of research.

If youhave any queries please feel free to contact us on our E-mail **infoijrcm@gmail.com**.

I am sure that your feedback and deliberations would make future issues better – a result of our joint effort.

Looking forward an appropriate consideration.

With sincere regards

Thanking you profoundly

Academically yours

Sd/-
Co-ordinator

DISCLAIMER

ABOUT THE JOURNAL

In this age of Commerce, Economics, Computer, I.T. & Management and cut throat competition, a group of intellectuals felt the need to have some platform, where young and budding managers and academicians could express their views and discuss the problems among their peers. This journal was conceived with this noble intention in view. This journal has been introduced to give an opportunity for expressing refined and innovative ideas in this field. It is our humble endeavour to provide a springboard to the upcoming specialists and give a chance to know about the latest in the sphere of research and knowledge. We have taken a small step and we hope that with the active co-operation of like-minded scholars, we shall be able to serve the society with our humble efforts.

Our Other Journals