Research Journal of Engineering Sciences ISSN 2278 – 9472
Vol. **3(5),** 12-25, May **(2014)** *Res. J. Engineering Sci.*

A Survey on Tools and Algorithms of Ontology Operations

S. Chitra[1] and G. Aghila[2]
[1]Department of Computer Science and Engineering, Research and Development Center, Bharathiar University, Coimbatore, TN, INDIA
[2]Department of Computer Science and Engineering, National Institute of Technology Puducherry, Karaikal, Pondicherry, TN, INDIA

Available online at: www.isca.in, www.isca.me
Received 15th April 2014, revised 10th May 2014, accepted 23rd May 2014

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. Need to be access multiple ontologies from several applications. Ontology management is possible through interoperability of semantic data sources. Importance of ontology especially its application in the WWW leads to Semantic Web that is being standardized in various research and application. Different ontological tools have different data representation and operations of concepts with respect to their input. Their functionalities and information structures also vary across their tools and operations. There are at present very few survey papers that provide in-depth discussion of these techniques and their applications. In this paper we converse the various state-of-the-art ontological tools with their internal operations and various algorithms.

Keywords: Tools, algorithms, ontology operations, ontologies, techniques.

Introduction

Ontology's are playing an increasingly essential role in understanding management and the Semantic Web and to accomplish several purposes of cognitive informatics. They represent a shared conceptualization of a community, due to the fact they provide a shared acquaintance model for semantic driven application in the Internet[1]. Ontology represents knowledge as a set of concepts within a domain and the association between those concepts. It is a structural frame for organizing information and used in System engineering, Software engineering, Semantic Web and Artificial Intelligence. It is becoming popular because of following notion: a shared and general understanding of some domain that can be communicated between people and application system[2].

Ontology Operations: Despite current developments, the Web remains in essence composed with documents written in Hyper Text Markup Language, a language focusing on visualization and presentation of information to be consumed essentially by humans. Machines can interpret the information available on the Web, but it is essential to supply it in a language that both can understand. This is specifically the aim of the Semantic Web. The method of joining semantic concepts to natural language is referred as Semantic Annotation. The success of the Semantic Web depends essentially on the propagation of annotated Web content. Annotating data can help supply better search services, ever since queries will be based not only on fixed key words, but also well-defined concepts described by the ontology of the domain that we desire to search for information[3].

A basic operation performed while managing ontology's is the "Mapping" which interprets a set of correspondences between associated concepts through two or more ontology's of the same application domain or similar domains. Mappings can solve many forms of mismatches that may result from several ontology editors and environments used by different knowledge engineers in various settings and during various time intervals. There are three main forms of linguistic mismatches between ontology's, namely: the syntactic, semantic and lexical mismatches[4].

Ontology matching is far from being a fully automated task. In most cases where high exactness is required, manual intervention will be necessary to verify or fine-tune the matching formed by the automatic algorithms. As a significance, to create exact matching in a reasonable amount of time, users and tools must be matched together. This process, regularly referred to as semiautomatic ontology matching, normally follows an iterative process[5]. Ontology integration is the development of structure ontology in one subject reusing one or more ontology's in different subjects[6]. Scheming ontology alignment is a tedious process, in which many ongoing efforts to develop tools such as GUI and matching algorithms, in order to make it easier[7]. This paper is organized as follows: Next section discusses about the related work done on ontology management operations. Section 3 gives the detailed description of different ontology operations with their algorithms and tools. Section 4 depicts the tabular representation of comparative analysis done on these operations. Section 5 concludes the work done.

Related Work

In the literature, many Ontology Management Tools/systems and related works are available. Among them, Namyoun et al.,[8] have reported about the tools, systems, and related work of ontology mapping. Three ontology mapping categories are explained as i. mapping between local ontologies ii. mapping between an integrated global ontology and local ontologies, and iii. mapping on ontology merging and alignment. In their work, a comparison has been done on the evaluation criteria, input requirements, level of user interaction, type and the content of output, and also in the five dimensions called structural, lexical, domain, instance based knowledge, and type of result.

Natalya[9] has given a brief survey of the approaches to semantic integration developed by researchers in the ontology community. They have focused on differentiating the ontology research from other related areas. Also, they have discussed different techniques for finding correspondences between ontologies, declarative ways of representing and use of these correspondences in various semantic-integration tasks. Akrivi Katifori et al.,[10] 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.,[11] have provided a framework for analyzing the methodologies that compare them to a set of general criteria. A classification has been obtained based upon the bottom-up or top-down directions of ontology construction. 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.,[12] has given an article based on empirical evidence and real world findings of the methodologies 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.

Yannis et al.,[13] focuses the survey on current state of the art in ontology mapping. Ravi et al., does the analysis on ontology mediation tools. They review recent approaches, frameworks, techniques and tools. J L Hong et al,[14] have focused on the current ontological tools and their applications but they have not analyzed the existing tools. Siham et al, have done surveys on mapping, use of alignment and merging operations with mapping, and they are not specific about other ontological operations. Sean M. Falconer and Natalya F. Noyet al, describe the techniques for Ontology Matching and their application, process technique, exiting tools and its usage. Good surveys through the recent years are provided. However, none of the surveys provide a comparative review of the existing ontology management techniques and systems. In our previous work an attempt has been carried out by us for the classification of ontology management methodologies. Now we extend our survey for the all ontology management operations with algorithms. Finally we also have provided the comparative study for ontology management operation tools based on algorithms.

Ontology Operation Management

Many applications use multiple ontologies, particularly when using modular design of ontologies or when we need to combine with systems that use other ontologies. Ontology management will integrate all the operations together and make finest application for the easy ontology process, this all operations are essential for the preservation and integration of ontologies. Alignment is a method of mapping between ontologies in both orders whereas it is feasible to change original ontologies so that the proper translation exists. Thus it is possible to add new concepts and relations to ontologies that would structure proper equivalents for mapping. The condition of alignment is called articulation. Refinement is mapping from ontology A to another ontology B so that every concept of ontology A has comparable in ontology B, however primitive concepts of ontology A may communicate with non-primitive concepts of ontology B.]

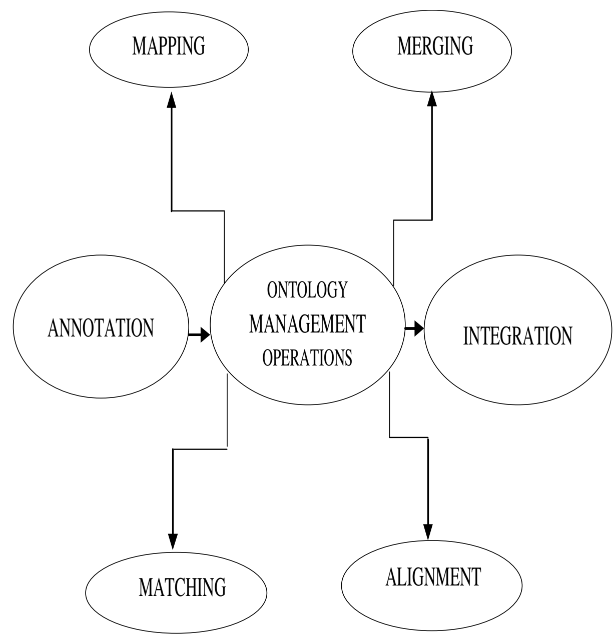

Figure-1
Ontology Operation Management

Modification defines limited ordering of ontologies. Integration is a method of looking for the similar parts of two different ontologies A and B while developing new ontology C that allows to transform between ontologies A and B and so allows interoperability between two systems where one uses ontology A and the additional uses ontology B. The new ontology C can replace ontologies A and B or can be used as an Interlingua for conversion between these two ontologies.

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

Algorithm: Quick-sort algorithm: Ontology mapping explore with the similarity that is being deliberated with the simple additive weighting with ontology concept with the aid of Quick-sort algorithm[15] that defines the ontology property relation which tends to adopt during the implementation of the algorithmic quick sort manipulation which is similarly very low and in order to get the accurate mapping result for solving the problems, It gives the ontology structure similarity in computing the lexical similarity.

Parsing graph-based algorithm: In consistent with the estimation of Parsing graph-based algorithm[16] similarity of the whole ontology, initially they set ontology basic correspondence degree as a small base value), and then add it to the parsing graph based algorithm. The problem is that fragmented data environments like the Semantic Web inevitably over mapping lead to data and information-quality problems. This is because the applications that process this data deal with the other comparisons of algorithms ill-defined, inaccurate, or inconsistent information about the domain.

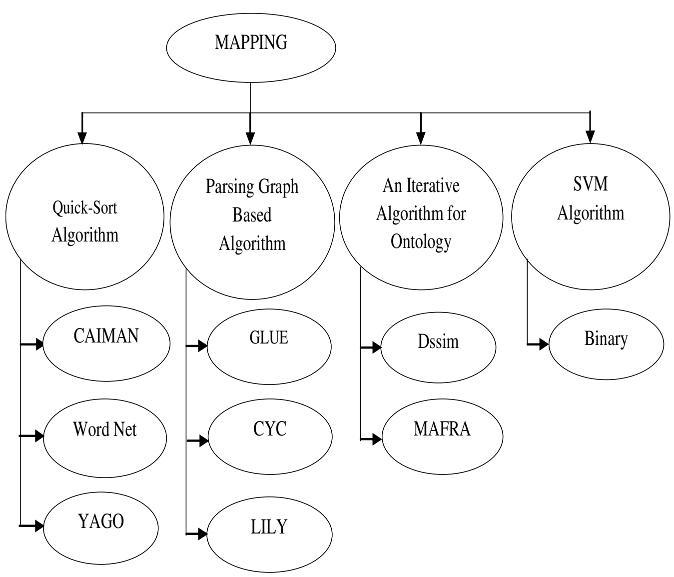

Figure-2
Ontology Mapping

QOM - Quick Ontology Mapping: The incomplete data over the quick ontology mapping can mean the different things to data consumer and data producer in a given application scenario. In traditional integration QOM - Quick Ontology Mapping [17] algorithmic implementation scenarios, resolving these data-quality issues requires a vast amount of time and resources for social amateurs before any integration in this quick ontology mapping which can take place.

SVM: The Support Vector Machine (SVM) algorithm was developed by Cortes and Vapnik, 1995. It is probably the most widely used kernel learning algorithm. It achieves relatively robust pattern recognition performance using well established concepts in optimization theory. Despite this mathematical classicism, the implementation of efficient SVM solvers has diverged from the classical methods of numerical optimization [18].

Tools: Binary Classification: Ming Mao dealt ontology mapping problem with machine learning techniques. His approach has five steps: i. generated various domain independent features, ii. randomly generates training and testing set for OAEI benchmark tests. iii. Train an SVM model on the training set. 4. Classify testing data on the trained SVM model. 5. Extract mapping results of testing data. Testing data are evaluated against ground truth. Steps 2 to 6 is repeated 10 times and the average evaluation result is used to eliminate bias.

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

CAIMAN: Martin Lacher 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 a 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.

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

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

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

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

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

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

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

The alignment process mainly contains three steps: i Preprocessing step parses the ontologies. ii. Match computing step uses suitable methods to compute the similarity between elements from different ontologies. iii. Post processing step is responsible for extracting, debugging and evaluating mappings.

Ontology Merging: Merge of ontologies means design of a new ontology by linking up the existing ones. Predictable necessity is that the new ontology contains all the knowledge from the novel ontologies, however, this requirement does not have to be fully fulfilled, since the original ontologies may not be together totally reliable. In that case the new ontology imports selected knowledge from the original ontologies so that the result is constant. The merged ontology may begin new concepts and relationships that serve as an association between terms from the original ontologies.

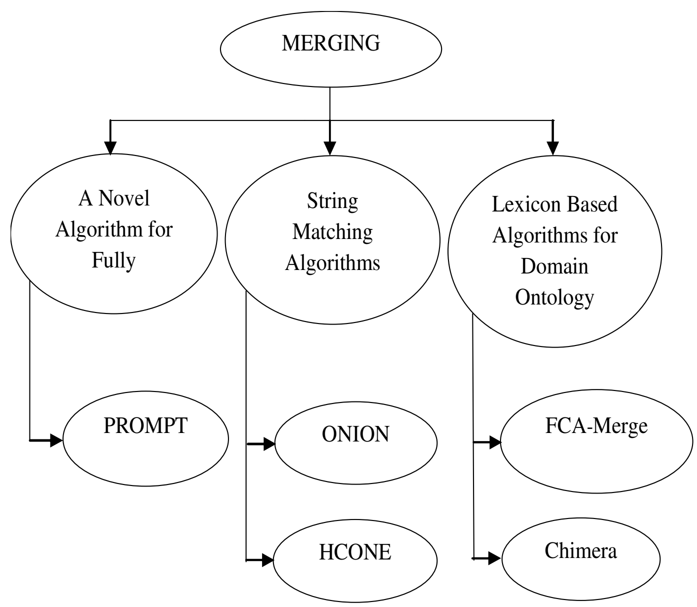

Figure-3
Ontology Merging

Algorithm: A Novel Algorithm for Fully Automated Ontology Merging: Ontology merging is the progression of crafting new solitary comprehensible ontology from two or more prevailing source ontologies related to the same domain. The new ontology will replace the source ontologies over this novel algorithmic approach. A merging task resembles a construction of a new ontology with this Novel Algorithm development that has its own comparisons with an algorithmic analysis implementation that allows starts with defining the domain and a lexicon of a typical, common vocabulary. It forms a base for a hierarchy of concepts – they are divided into classes and proper ontological relations that are attached.

String matching algorithms: Ontology merging, a new ontology generated heterogeneity delinquent which could be fully solved with the occasional string matching algorithms, which are an imperative class of algorithmic strategy that try to find a place where one or several strings (also called patterns) are found within a larger string or text. Ontology merging is a substantially effective way to solve the grim of ontology heterogeneity with this String matching algorithm. This indicates that the systems to merge ontology with a single-mapping result will cause the glitches of semantic intensity insufficiency, merging inefficiency, structure merging confusion, etc.

Lexicon based algorithm for domain ontology merging: Ontology similarity algorithms tend to greatly improve the accuracy in mapping-based ontology merging when compared to the other two algorithms. Here, WorldNet is utilized to manage over the Lexicon based algorithm. If so, it should be merged directly. If not, it should examine the relation between the classes to which the terms belongs. On condition that the classes are the relations between the merging conflict that ought to be treated as the corresponding merging hyponymy.

Tools: Prompt: The PROMPT suite 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, an ontology tool for finding additional points of similarity between ontologies like iPROMPT, an ontology versioning tool (PROMPT Diff) and a tool for factoring out semantically complete sub-ontologies (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.

Onion: Mitra has developed a scalable framework 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 these mantic 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 .

CA-Merge: Gerd Stumme 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.

Chimera: Deborah L. McGuinness developed tool called Chimera for ontology merging. Chimaera is aimed to support: i. merging multiple ontologies and ii. 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 .

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

Ontology Annotation

Adaptive Information and Extraction Algorithm: Proprietary algorithm: A proprietary algorithm is a series of steps or rules performed to complete a specific goal, belonging to a commercial group as it has been trademarked or patented by its owner. A search engine ranking algorithm is an example for one of the commercial web search engines - some of the information may be readily available to the public except the source to avoid misuse.

TBD: Track-before-detect (TBD) is a concept according to which tracked before declaring it a target. In this approach, the data about a tentative target are integrated over time and may yield detection in cases when data from any user and detected a target. The TBD approach may be applied both for pure detection when the unsure target displays a very small amount of apparent motion and for actual motion tracking.

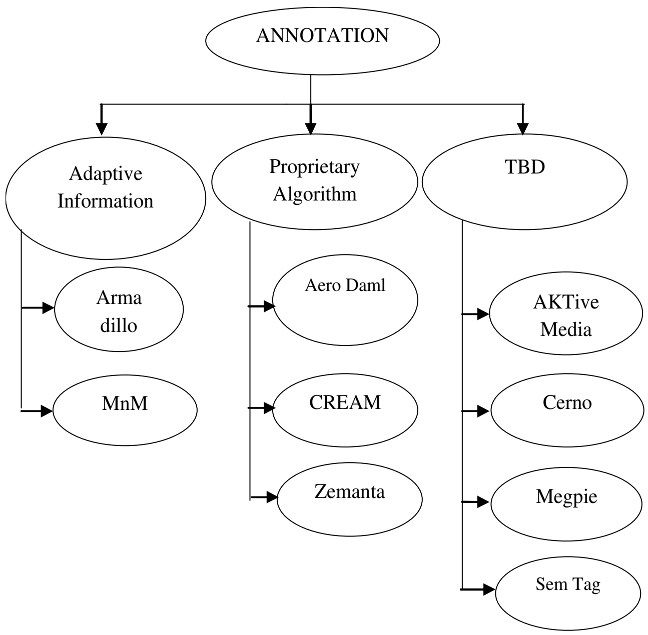

Figure-4
Ontology Annotation

Tools: AeroDAML: AeroDAML is a knowledge markup tool that automatically generates DAML annotations on web pages after the application of natural language extraction techniques. AeroDAML maps proper nouns and common relationships with classes and properties in DAML ontologies. AeroDAML has two different modes of utilization. The web-enabled version of AeroDAML supports annotation with a default generic ontology of commonly found words, classes and relationships. The user enters a URI and AeroDAML responds with the DAML annotation for the URI normally associated with a web page. The client server version of AeroDAML supports annotation with customized ontologies. In this version the user must enter a file name and AeroDAML returns the DAML annotation for the text document.

Aktive Media: AKTive Media is a user centric system for annotating documents with support of text, images and HTML documents with ontology-based and free-text annotations. Both author and reader can perform annotations allowing the utilization of different ontologies. The annotations are not stored in the document but separately with authorship allowing users to share comments and annotations with other members of the community using a centralized server. Most annotations are done manually but various techniques are available to reduce the effort of annotating. AKTiveMedia is the successor of AKtiveDoc.

Armadillo: Armadillo is a system for creating automatic domain-specific annotation on large repositories in an unsupervised way. This tool implements an adaptive information extraction algorithm using a pattern-based approach to find entities from a handful of seed examples that must be provided by the user and discovers new facts using examples. Learning is seeded for information extraction from redundant information repositories, such as databases and digital libraries, or from a user-defined lexicon. The information that is retrieved is in part used to annotate a set of new documents. Then, the new annotated documents are used to bootstrap learning. The user can repeat this process until the annotations reach the quality expected .

Cerno: Cerno is a framework for semi-automatic Semantic Annotation of text documents according to a domain-specific semantic model. Cerno uses lightweight techniques and tools for code analysis and markup, requiring limited human effort for adaptation to a new domain. The Cerno framework comprises a process for defining keywords and grammar-based rules so that it can identify instances of concepts in a textual document and an architecture that applies the rules to annotate and extract instances that are identified in a document.

Cream: CREAM (Creation of Metadata) is a framework for creating annotations, in particular relational metadata. CREAM supports metadata creation during authoring and after authoring Web Pages. CREAM includes inference services, crawler, document management system, ontology guidance/fact browser, document editors/viewers, and Meta ontology. Onto Annotate and Onto Mat Annotizer are two different implementations of the CREAM framework.

Magpie: Magpie is presented within a web browser, avoiding the need of manual annotation by automatically annotating web resources by associating text strings that exist in the web page with an ontology chosen by the user. It is similar to Thresher using wrappers to produce RDF in "real-time" as users explore web pages.

MnM: MnM is an annotation tool, which provides the possibility of annotating pages in an automatic or a semiautomatic way with semantic metadata. MnM is an ontology-based information extraction engine based on unsupervised learning. It can learn extraction rules from a training corpus applying these rules on unseen news articles to populate ontology. MnM has a web browser integrated with an ontology editor. It provides open APIs for the connection to ontology servers and for the integration with information extraction tools. MnM is a good example of an ontology editor, because it is web-based, facilitates semantic tagging and mechanisms for large-scale automatic tagging with convenient metadata on web pages.

Zemanta: Zemanta is an online annotation tool that offers word processing of formless documents recommending applicable links to diverse contents on the Web. The content is analyzed with a proprietary algorithm for natural language and semantic processing and it combines machine-learning techniques to regularly refine its recommendations.

Ontology Matching

SVM algorithm[19]: **GAOM: Genetic Algorithm based Ontology Matching:** GAOM algorithm is used as a backbone of multi-level matching technique and performs a neighbor search, to find the correspondences between the entities in the given ontologism. A main feature of this algorithm is the high quality of the matches it finds. Besides, as the result of the initial search introduced, this algorithm converges fast, making it comparable to existing SVM and Iterative ontology techniques.

An Iterative Algorithm for Ontology Matching: This combines standard string match metrics with a structural similarity measure that is based on Iterative Algorithm in the vector representation. After all pair wise similarities between concepts the algorithm will have been calculated to apply well-known graph algorithms to obtain an optimal matching criteria. It is also capable of using existing mappings to a third ontology as training data to improve accuracy.

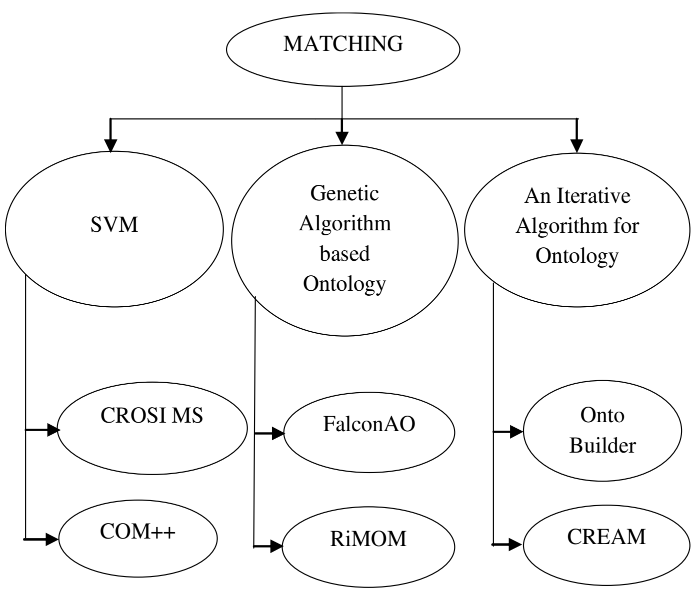

Figure-5
Ontology Matching

Tools: Falcon AO: Wei Hu has 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 method to help enable interoperability between (Semantic) Web applications using different but associated ontologies. It consists of five components: the Repository to temporarily store the data through the matching process; the Model Pool to control ontologies and to build different models for different matchers; the Alignment set to generate and to assess exported alignments; the Central Controller to configure matching strategies and to execute matching operations.

CROSICMS: Yannis Kalfoglou has proposed architecture CROSI CMS, which is a 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 resulting similarity values are compiled by multiple comparison aggregators running in parallel or repeated order.

RiMOM: Xiao Zhang proposed a structure 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 getting 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 finement refines the alignment results from the previous steps.

Anchor-prompt: Natalya F. Noy has 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.

COM++: Erhard Rahm et al., developed COMA++ as a representation 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 completing Engine to perform match operations. Automatic match processing is performed in the Execution Engine as a three step process, component identification, matcher execution and similarity combination. The obtained mapping can be used as input in the next iteration for additional modification.

OntoBuilder: Avigdor Gal has proposed tool OntoBuilder for ontology matching. The OntoBuilderproject 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.

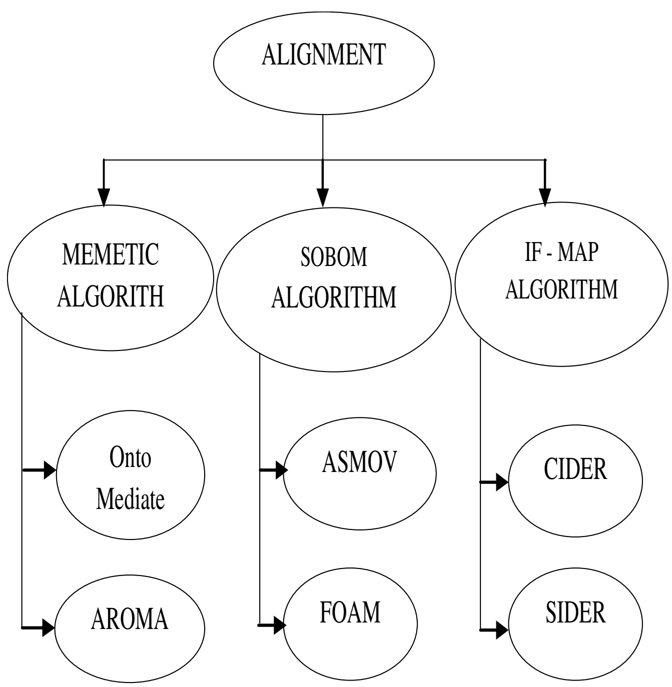

Figure-6
Ontology Alignment

Algorithm: Memetic: Recurrently, ontologies in accordance with the alignment based domain incorporate with the Memetic algorithm have many differences, which are also known as heterogeneity. The reason behind heterogeneity is rooted in diversity in ontology modeling based on different views which have its own consideration with the domain analysis. Heterogeneity cannot be avoided in ontological alignment with the distributed and open systems as, for instance, with its Semantic Web. Moreover, an ontology may follow different modeling goals.

SOBOM: Modeling lexical constrains over the IF-map algorithm with the SOBOM algorithm can occur at various levels of semantic scrutiny bases on a different meaning whereas the semiotic alignment over the ontological alignment will tends to analyses the bases on the individual interpretation of ontology entities and the various relationships among them regarding certain context that tends to specify the usage of the ontology techniques for the SOBOM algorithm for comparative analysis IF-map algorithm.

IF-MAP: In order to tackle the need of sharing knowledge within and across organizational boundaries over the ontological constrain , the last probabilistic observation has saw many of the estimated both in academia and industry advocating for the use of ontology's with the implementation of the IF-map algorithm as a means for providing a shared understanding of common domains. But with the generalized use of large distributed environments such as the World Wide Web came the proliferation of many different ontologism, even for the same or similar domain, hence setting forth a new need of sharing–that of sharing ontologism.

Tools: ASMOV: Yves R. Jean-Mary et al.,[20] 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 the object to data type 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), and 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.

CIDER: Jorge Gracia et al.,[21] proposed an alignment service called CIDER (Context and Inference base D align ER) 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 the 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 modeling 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.

SPIDER: Marta Sabou had given a system SPIDER, which provides alignments with a variety of map types this system, combines two concrete subsystems. First, the CIDER algorithm is to derive equivalence mappings. Second, the alignment is extended with non-equivalence mappings derived by Scarlet. CIDER is briefly explained in the 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.

Onto Mediate: Gianluca Correndo proposed a project Onto Mediate 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 Dataset 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.

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

FOAM: Marc Ehrig has 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 a 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, as the similarity of one alignment influences the similarity of neighboring entity pairs.

Ontology Integration

Algorithm: Greedy algorithm: Ontology integration's goal is to establish independent connection between different ontology's. So if we use the Portage with the greedy algorithm, it tends to construct the top shimmering ontology and enrich the relation between ontology's with the query transforming algorithm. The approach, based on the discussion of various custom relations from different levels of algorithmic domains classify the generalization to integrated relation, to further improve the ontology's storage, retrieval and design and laid the foundation for knowledge reasoning.

A Multi-level Matching Algorithm: Various similarity measures have been proposed for ontology integration to identify and suggest possible matches of components in a semi-automatic process. A (basic) Multi Match Algorithm (MMA) can be used to combine these measures effectively, thus making it easier for users in such applications to identify "ideal" matches found. A multi-level extension matching ontological integration, which is similarly partitioned by the user and that there is a partial order on the partitions, also defined by the user.

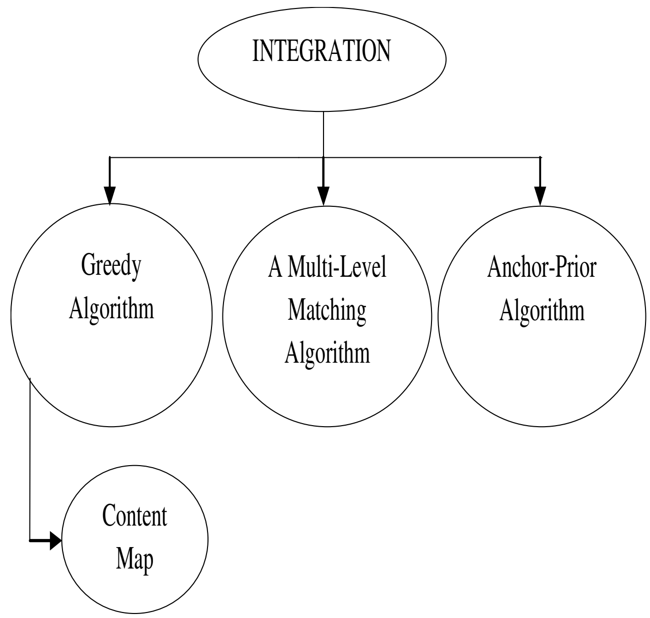

Figure-7
Ontology Integration

Anchor-Prior algorithm: The main contribution of the approach is to reduce the computational complexity and to enhance the accuracy of ontology integration with the implementation of Anchor-Prior algorithm. The key idea of this approach is to start from an Anchor (two matched concepts) to work towards a collection of matched pairs among its neighboring concepts by computing similarities between the "priorly" collected concepts across the ontology's starting from the anchor.

Content Map: Ernesto Jim_enez-Ruiz et al., developed a system called Content Map A logiC-based ONtology inTEgratioN Tool using MAPpings. The Content Map evaluates and repairs the logic consequences of merging two independent ontologies using mapping. The method is as: i. Compute mapping M between O1 and O2 using a mapping algorithm, and filter those using the criteria. ii. Compute logic difference and evaluate the impact by comparing the entailments holding before and after the integration. iii. Detect unintended entailments and select them. iv. Compute repair plans and execute the best one according to the user requirements.

Table-1
Comparison

Methodology	Operation	Tools Name	Algorithm	Input/ Output	Language Used	Technology Used	Automation Level	Application
Machine Learning	Ontology Mapping	Binary classification	SVM	Two ontologies and Evaluation Result	*	CROSS TASK	*	Structural and web Feature
Machine Learning	Ontology Mapping	GLUE	Parsing graph-based algorithm	Concepts in Taxonomy and similarity Measure	*	Jaccard Coefficient and MSP	Semi-automated	Used To Meata Learning
Machine Learning	Ontology Mapping	CAIMAN	Quick-sort algorithm	Two Ontologies and Mapping result	*	*	Semi-automated	It used for Text Classification
Machine Learning	Ontology Integration using Mapping	Content Map	Greedy algorithm	Pre-Computed mapping ontologies and visualizing the mapping results	OWL2	*	Semi-automated	Description of logic and heuristics
Machine Learning	Ontology Merging and Alignment	Prompt	Novel Algorithm for Fully Automated Ontology Merging	Two DAML Ontologies and Merged Ontology	OWL	OKBC	Automated	Linguistic Similarity
Structure Based	Ontology Matching	Falcon AO	GAOM	RDFS or Owl Ontology and RDF/XML Format	JAVA	Jena	Semi-automated	Capturing Knowledge From Web for the process
Structure Based	Ontology Alignment	ASMOV	SOBOM	OWL-DL	*	Jena ARP	Semi-automated	Increase accuracy
Structure Based	Ontology Alignment	CIDER	IF-MAP	RDF/OWL input andRDF document	JAVA	Alignment API	Automated	Easy for Similarity measure to compare
Structure Based	Ontology Alignment	SPIDER	SOBOM	RDF/OWL ontologies and RDF document	JAVA	CIDER and Scarlet	Semi-automated	Provides Alignments with variety of mapping types
Structure Based	Ontology Merging	ONION	String matching algorithms	RDF file and plain text	JAVA	SKAT	Automated	Checks input Mc of users choices
Structure Based	Ontology Matching	CROSICMS	SVM	OWL Ontologies and matched result	JAVA	Jena JWNL	*	Easy for Structure matching system
Structure Based	Ontology Matching	OntoBuilder	An Iterative Algorithm for Ontology	OWL Ontologies and produce matching	JAVA	*	Semi-automated	Find best mapping between two Ontologies

			Mapping					
Structure Based	Ontology Mapping	DSSim	QOM	Two Ontologies andMapping result	OWL	SKOS parser	Semi-automated	Assess quantitative similarity values
Structure Based	Ontology Merging	FCA-merge	Lexicon based algorithm	Two ontologies + DL and produce merged ontology	*	Concept lattice	Semi-automated	Merge employ from bottom up
Semantic Based	Ontology alignment and mapping	OntoMe diate	A Multi-level Matching Algorithm	OWL ontologies and produce result	J2EE and AJAX	Jean API	Semi-automated	Social internet
Semantic Based	Ontology Matching	RiMOM	GAMO	RDF and OWL ontologies and produce result	JAVA	OWL API	Automated	To find minimal risk for each sample
Semantic Based	Ontology Matching	Anchor-prompt	SVM	Pair of anchors and generates new pairs with close similarity	OntoVi z	Protege 2000 plugin	Automated	Used to find the anchor's
Semantic Based	Ontology Mapping	MAFRA	QOM	RDF files and produce output	JAVA and KALO N	MAFRA Service Interface API	Semi-automated	Setup Consensus Between various of proposals
Semantic Based	Ontology Merging and diagnosing	Chimera	Lexicon based algorithm	RDF and DAML and merged output	*	*	Semi-automated	Merge Multiple Ontologies and diagnesing
Semantic Based	Ontology Merging and alignment	HCONE	String matching algorithms	Two ontologies and result	JAVA	Neo Classic Descripti on Logic	Semi-automated	Validate used to find the minimum set
Semantic Based	Ontology Mapping	Word Net	Quick-sort algorithm	Different words and categorized words	*	Light Weight Ontologic al Tech	Automated	Hyernyms synonyms
Semantic Based	Ontology Mapping	CYC	Parsing graph-based algorithm	Concepts, Facts and result	JAVA	Relations hip between Tree and leaves	Semi automated	Terrorism knowledge application
Semantic Based	Ontology Mapping	BabeNet	Parsing graph-based algorithm	Integrate domain and knowledge and high level ontology	*	Light Weight Ontologic al Tech	Automated	High level information provided domain
Semantic Based	Ontology Mapping	YAGO	QUICK Sort Algorithm	High data and Accuracy Ontology	JEAN API	Light Weight Ontologic al Tech	Automated	Artificial Intelligence System
Semantic Based	Ontology Annotation	Aero DAML	Proprietary algorithm	URI and DAML Annotation result	OWL	NLET	Automated	Automaticall y generates

	and mapping							DAML annotation
Semantic Based	Ontology Annotation	AkTive Media	TBD	Text and annotation result	HTML RDF	Various Tech	Automated /Manual	User centric system for annotating
Semantic Based	Ontology Annotation	Armadill o	Adaptive information Extraction	Redundant information and Annotation Result	RDF	Diverse Tech	Automated	Creating automatic domain for specific annotation
Semantic Based	Ontology Annotation	Cerno	TBD	Text and Annotation Result	RDF	Light weight Tech	(semi)Auto mated	Code annalysis and markup
Semantic Based	Ontology Annotation	CREAM	Proprietary algorithm	Relational metadata and annotation Result	HTML OLML	Manually	Automated /Manual	Document editors/view ers
Semantic Based	Ontology Annotation	Magpie	TBD	Text Strings And Annotation Result	HTML	RDF	Automated	Avoiding manual annotation
Semantic Based	Ontology Annotation	MnM	Adaptive information and extraction algorithm	Text Strings and annotation Result	RDF OIL	Extractio n Rules	(Semi) Automated	Mechanisms for large scale automatic tagging
Semantic Based	Ontology Annotation	Zemanta	proprietary	Unstructured Document And Annotation Result	OWL	Machine Learning Tech	(Semi) Automated	Combine machine learning tech for recommenda tion
Semantic Based	Ontology Annotation	SemTag	TBD	Lexical and taxonomic information Annotation Result	RDF	Structural Analysis	Automated	Operate centralized application
Hybrid	Ontology Matching	COM++	SVM	XML,OWL files as input and generates	*	OWL API	Semi Automated	Store all related data And mapping tools
Hybrid	Ontology Alignment	AROM A	A Multi-level Matching Algorithm	OWL ontologies and produce output	*	OWL Libraries	Semi Automated	Asymmrtric matching find out relatives
Hybrid	Ontology Mapping	Lily	Parsing graph-based algorithm	RDF ontologies and generates output in text	JAVA	LOM.GO M and SOM	Automated	Preliminary alignment are carefully analayzed
Hybrid	Ontology Alignment and mapping	FOAM	SOBOM	OWL ontologies and generates aligned text file	JAVA	Google API And KAON2	Semi-auto mated	Selects the ontology for specific domain

Conclusion

Ontology operations management is one of the essential processes in any Semantic Web and all kinds of applications. Based on our reviews and findings, we positively hope that ontological tools will have multilingual support in the ever. The ontology tool's automated process will allow people to work more efficiently and effectively, and afford them the time to concentrate on difficult tasks that are not easily automated. Our aim of this survey is describing the importance of ontological ontology operations management and applications of ontology tolls and its algorithms. This process clearly depicts the importance of ontology management operations keeping in view of 37 tools/frameworks collected. The comparison table shows the essential features of the tools that are being discussed. The result of this survey and analysis provides complete understanding of ontology management operations, algorithm and tools developed in the ontology operations. This work can be extended with other tools/frameworks for the further ontology management operations.

References

1. S. Shakkottai, T.S. Rappaport and P.C. Karlsson., Cross-Layer Design for Wireless Networks, *WNCG Tech. Rep. TR-2003-04-00001, Wireless Networking and Commun. Group, Dept. of Elec. and Comp. Eng.,* **(2003)**

2. Vineet Srivastava and MehulMotani., Cross-Layer Design: A Survey and the Road Ahead, *IEEE communication magazine*, **43(12),** 112-119, **(2005)**

3. Ishizaki F. and Gang UkHwang., Cross-Layer Design and Analysis of Wireless Networks Using the Effective Bandwidth Function, *IEEE Transaction on Wireless Communications*, **6(9)**, 3214-3219 **(2007)**

4. V. Srivastava and M. Motani., The Road Ahead for the Cross-Layer Design, *in Proc. of IEEE 2nd International Conf. on Broadband Networks*, **1,** 551-560, **(2005)**

5. Fotis Foukalas, Vangelis Gazis, And Nancy Alonistioti., Cross-Layer Design Proposals for Wireless Mobile Networks: A Survey And Taxonomy, *IEEE Communication Surveys and Tutorials.,* **10(1)**, 70-85, **(2008)**

6. S. Toumpis and A. J. Goldsmith., Capacity Regions for Wireless Ad Hoc Networks, *IEEE Transactions on Wireless Communications.*, **2(4)**, 746-748, **(2003)**

7. S. Pollin, B. Bougard and G. Lenoir., Cross-Layer Exploration of Link Adaptation in Wireless LANs with TCP Traffic, *Proc. IEEE Benelux Chapter on Communications and Vehicular Technology.*, **(2003)**

8. F. Aune., Cross-Layer Design Tutorial, *Norwegian University of Science and Technology, Dept. of Electronics and Telecommunications*, Trondheim, Norway, **(2004)**

9. A.J.Goldsmith and P.P.Varaiya., Capacity of Fading Channels with Channel Side Information, *IEEE Trans. Inform. Theory.*, **43(6)**, 1986-1992, **(1997)**

10. D. Wu and R. Negi., Effective Capacity: A Wireless Link Model for Support of Quality of Service, *IEEE Trans. Wireless Commun.*, **2(4)**, 630–43 **(2003)**

11. Xi Zhang and Jia Tang, Hsiao-Hwa Chen, Song Ci, Mohsen Guizani., Cross-Layer-Based Modeling for Quality of Service Guarantees in Mobile Wireless Networks, *IEEE Comm. Mag.*, **44(1),** 100-106, **(2006)**

12. Q. Liu, S. Zhou, and G. Giannakis., Queuing with adaptive modulation and coding over wireless links: cross-layer analysis and design, *IEEE Trans. Wireless Commun.*, **4(3)**, 1142–1152, **(2005)**

13. Jie Gong, Sheng Zhou, ZhishengNiu., Queuing on Energy-Efficient Wireless Transmissions with Adaptive Modulation and Coding, *IEEE International Conference on Communications* (ICC)., **1(5)**, 5-9, **(2011)**

14. Tin-Yu Wu, Chi-Yuan Chen, Kai-Di Chang, Diego Chung and Han-Chieh Chao., Pave the Way to Future Smart Living Space-Cross-layer Enhanced AAA for 4G Core Network, *Multimedia and Ubiquitous Engineering, 2007. MUE '07, International Conference on.*, **1(2)**, 191-203, **(2007)**

15. M.S. Alouini and A.J. Goldsmith., Adaptive modulation over Nakagami fading channels, *Kluwer Journal on Wireless Commun.*, **13(1–2)**, 119-143, **(2000)**

16. E. Malkamaki and H. Leib., Performance of truncated type-II hybrid ARQ schemes with noisy feedback over block fading channels, *IEEE Trans. Commun.*, **48(9)**, 1477-1487, **(2000)**

17. Q. Liu, S. Zhou, and G. B. Giannakis., Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links, *IEEE Trans. Wireless Commun.,* **3(5)**, 1746-1755, **(2004)**

18. Reuven Cohen, Guy Grebla and LiranKatzir., Cross-Layer Hybrid FEC/ARQ Reliable Multicast With Adaptive Modulation and Coding in Broadband Wireless Networks, *IEEE/ACM Transactions on Networking.*, **18(6)**, 1908–1920, **(2010)**

19. Ashok K. Karmokar and Vijay K. Bhargava., Adaptive Coding and Modulation for Hybrid ARQ Systems over Partially Observable Nakagami-m Fading Channels, *IEEE Global Telecommunications Conference Publications.*, 1-6, **(2006)**

20. Feijin Shi and Dongfeng., Cross-Layer Performance Analysis of TCP over Wireless Link with AMC and Type-III HARQ, *Spread Spectrum Techniques and*

Applications, ISSSTA '08. IEEE 10th International Symposium on., 133-137, **(2007)**

21. X. Lin and N. B. Shroff., The Impact of Imperfect Scheduling on Cross Layer Rate Control in Wireless Networks, *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE.*, **3,** 1804-1814, **(2005)**