Ontology Views for Ontology Change Management: a state of the art

Perrine Pittet, Christophe Cruz, Christophe Nicolle

Le2i, University of Burgundy, France

INTRODUCTION

In the literature, ontology change management systems (OCMS) are direct implementation of the concept of "change management" stated by reference [Klein]. Ontology change management combines ontology evolution and versioning features to manage ontology changes and their impacts. Since 2007, many works have combined ontology evolution and versioning into ontology change management systems (OCMS). The evolution subject has been massively studied in these works. They especially addressed the consistence issue for the application of changes on the ontology. These proposals constituted a consequent background for ontology change management but they did not take into account certain specificities of ontologies.

One of them is the fact that ontologies are decentralized data [Rajugan]. It means that multiple versions of the same ontology evolution are bound to exist over the Web and must be supported. It implies that ontology chronological evolution is not enough to manage ontologies. Actually, managing different parallel versions of a same ontology would bridge this gap.

Another characteristic is that ontologies are meant to grow during their lifecycle and may become too large to be used in its original scale by potential applications. Indeed, ontology development implies a dynamic and incremental process starting from the creation of a brute ontology, which has to be revised and refined [Djedidi]. Refinement often leads to the improvement of the ontology level of detail corresponding to the addition of new elements to its conceptualization. Therefore the ontology size may increase after each refinement iteration, with no guaranty that the ontology is still manageable by applications and understandable by humans.

In the literature, ontology views have been defined to bridge this ontology size issue and improve ontology reusability. Several definitions and implementations of ontology views have been studied in the Ontology View Management specific research field. However no agreement was found. Nevertheless, a view generally is a subset specification on an ontology, which allows to extract a manageable portion of the ontology capable to be used and queried by applications like the whole ontology. The resulting sub-ontology can be generated not only as a sub-graph of the ontology but also as an independent ontology, itself being a new interpretation of the domain. It can be considered as a new parallel version of the actual ontology validating the decentralized quality of ontologies. From the different approaches studied in this paper, can be deduced four types of view specification and implementation: query language based, subset extraction based , rule based and other views specifications based on hybrid techniques.

This article aims at giving an overall state of the art on ontology views, their objectives, their different implementations and use, the corresponding advantages and lacks, and finally defining the future research directions to take in the context of Ontology Change Management.

BACKGROUND

Ontology views objectives

An ontology is a common representation of the meanings of knowledge of an application domain.

Reference ontologies are intended to support the domain knowledge requirements of multiple disparate applications. They are often too large or too complex, however, for any specific application [Detwiler]. The problem is the same for large domain ontologies reused by specialized business applications. Automated or semi-automated evolution of these large ontologies is often impossible to realize because of their complexity.

In addition, the “world view(s)” provided by such domain ontologies may not match exactly the views required by particular applications. Also, potential applications will not access to the entire ontology [Noy & Musen]. Access to unnecessary ontology whole can be slow if the ontology is complex. In order to utilize these ontologies, therefore, applications often require custom ontology views tailored for use within their specific context. In this case, views can optimize the access time and query processing of ontology by only loading a small portion of this ontology. In recent years, many academic and industrial research directions have focused on these issues and some of the notable works. Displaying portions of ontologies is then crucial to allow OCMS supporting large ontologies evolution.

In user-centred applications, usually, users just need to use a small portion of their resources or may not have the right to access certain parts of the ontology. Views are used then to manage access policies, profile, context and data security for users [Rajugan]. Moreover adaptation of the ontology to several contexts or uses is one of the change management objectives.

Finally, the overall understanding of a complex ontology by the community may be impossible [Rajugan]. Evolution of such ontologies cannot then neither be realized by a human.

To resolve these issues, views should produce an **understandable and manageable portion of an ontology for local applications and users** as a means of enabling them to use standard well-developed ontologies [Orbst]. This requirement groups all the objectives of views cited above. This is the unique guideline we will retain all along the paper, especially to complete change management foundations.

Nevertheless, the definition of an ontological view is still an open question in the literature and its specification has been studied through several approaches.

Ontology view specifications in the literature

Major works technically define the view as a specification of a sub-graph of the ontology. Two main approaches exist in the literature to deal with ontology views and are presented below: **ontology query language based approaches,** which use queries to select subgraphs of the ontology, and **subset extraction approaches**, which uses subset extraction techniques to provide sub-portions of the ontology. Other view specifications approaches based on rules, pipes and named graphs are also exposed.

Ontology Query Languages Approaches

General description:

In databases, a view is specified as a query: any instances satisfying this request constitute the view. Researchers from the database field have been working on using views directly responding to requests. Since the views are themselves queries, this field of research reformulated the user query result to express it as physical and searchable data as a database table: the case of materialized views. So it is within the same perspective that ontology views defined by queries are specified on an ontology. Ontology query languages are used and extended for this purpose and provide several features to manage and update the views.

Existing Proposals:

Until now, several different RDF query languages have been proposed for querying within the semantic web, including RQL, RDQL and SPARQL. A number of proposals have been made for extending SPARQL's functionality, including Sparklers, ARQ, SPARQL++, CPSPARQL, nSPARQL, NetworkedGraphs and vSPARQL. Below are detailed RQL, RVL and vSPARQL proposals.

RQL is a declarative query language for RDF that allows querying over both resource descriptions and schemas. In this framework, a view represents a new class or property of the ontology [Magkanaraki et al.] have chosen to deepen this approach to define views based on queries.

RVL is a view definition language for creating virtual resource descriptions and schemas; RVL uses RQL as its query language. In RVL mechanisms to restructure the original class hierarchies and properties are proposed, enabling the creation of new resources, property values, classes or properties. From there, the view definition includes not only the result of the new structuration itself but also a set of predicates that define these new structures, linking them to the query results. RVL and Lightweight Ontologies [Volz et al.] both leverage RQL to provide declarative mechanisms for defining views over ontologies.

The use of vSPARQL is a general solution for RDF information set reuse inspired by database views in order to enhance the reusability of views specified by queries. The view definition language, vSPARQL, created here allows applications to specify the exact content that they are interested in and how that content should be restructured or modified. A Query Manager tool has been constructed in order to allow view queries to be edited, executed, and stored for reuse even materialized views. Subsequent queries can still be answered against the original ontology, ensuring that information set updates are reflected in the query results.

Subset Extractions Approaches

General description:

Three approaches have been developed for materialization of a subset of an information set for reuse: manual, structural, and logical. Traditionally, manual techniques have been used to derive a relevant subset of an information set for an application's use thanks to a partial mapping or other mapping techniques. A developer acquires a copy of the information set, which can be called ”sub-ontology” and then manually deletes, modifies, and adds facts until the application's requirements are met. This definition of view as extraction of a sub-ontology is different from the vision of databases here because the view contains the knowledge of the sources selected by the ontology mapping, whereas the database view approach only store a query in a new concept of the ontology. Below are detailed the main relative works of the literature.

Existing Proposals:

[Orbst] suggests creating application views of ontologies to allow applications to use standard ontologies without questioning the entire ontology. This approach combines views or "contexts" with mappings between concepts in the views and in the ontology. A mapping is generally used to associate each element of a given with one or more elements of a second one without impacting them. By extension it can be used to specify the relations between an ontology and its sub-ontology. Because it has no direct impact on them, it will preserve their terminological (schema) and assertional (instances) levels. So the problem of preserving coherence and consistency does not arise before further evolution of the view if the source ontology is known as consistent.

Structural techniques such as PROMPT's Traversal Views and Web ontology segmentation require a set of concepts and properties to be specified for inclusion in the result, which is a self-contained subset of the ontology. Starting from these key concepts, the output set is grown by recursively adding the specified properties and concepts until a fixed point is reached. In a view traversal, a user specifies departures, relations concerned with these concepts and the maximum distance to cross from the departure of these relationships (depth of transitive closure of relation). This mechanism allows users to extract portions of an ontology related to a particular concept or set of concepts (such as parts and various components of the heart).

In contrast, logical techniques have been developed for deriving modules from OWL-DL ontologies. A signature is specified containing all of the key concepts that should be contained within a selected subset of the ontology; leveraging the high-level semantics of OWL-DL, the extractor grows the set of concepts and axioms needed in the output module. For concepts in the signature, the ontology module is guaranteed to capture the meaning of the concepts, such that an application that performs reasoning after importing the module would give the exact same results as an application that performs reasoning after importing the entire ontology. However, it is not suitable for all applications' needs. The current approaches are limited to DL languages such as OWL-DL.

Other View Specification Approaches

In this paragraph, we describe several other approaches for manipulating or transforming RDF data.

Another approach to specify ontology views is to use rule languages. In logic, a rule is the act of drawing a conclusion based on the form of premises interpreted as a function which takes premises, analyzes their syntax, and returns a conclusion (or conclusions). Rules are mainly based on subsets of First Order Logic and possible extensions. Several different rule languages like TRIPLE and SWRL have been developed for the semantic web. General rule languages, such as Jena's general rule language, can be used to derive new facts without SWRL's semantic restrictions. Rule languages allow new facts to be inferred from existing facts in an information set of the ontology. They also offer more expressiveness (e.g. constructor for composite properties) than ontological query languages with efficient reasoning support. However, based on logic programming, they face the difficulties of decidability.

Several projects are building upon the notion of pipes to provide easy-to-use mechanisms for transforming and aggregating RDF data in mash-ups. A mash-up is a web application that uses and combines data, presentation or functionality from two or more sources to create new services. Semantic Web pipes are powerful data level mash-up tools based on RDF. They fetch RDF graphs on the Web, operate on them, and produce an RDF output. Operators are provided for extracting data from Web content, performing SPARQL queries over data, and inference. Similarly, SPARQLMotion provides a GUI editor for pipelining RDF data sources and transformation operations together. Banach provides a set of operators that can be pipelined inside Sesame to transform RDF data.

NRL uses named graphs as views. Named graphs are a key concept of Semantic Web architecture in which a set of RDF statements (a graph) are identified using a URI allowing descriptions to be made of that set of statements such as context, provenance information or other such metadata. Graph views are used to define specialized semantics and assumptions over RDF named graphs. Graph roles declaratively assign meaning to named graphs. However graph views are procedural, specifying the application or query/rules that need to be used to generate an output graph from an input graph.

MAIN FOCUS OF THE ARTICLE

Issues, Controversies, Problems

The Change Management requirement for ontology views given in the previous section, namely **producing a understandable and manageable portion of an ontology for local applications and users**, seems to almost be reached by the different approaches we have presented. Some technical and functional issues, depending on the type of specification, still hinder the emergence of an optimal specification for ontology view for ontology change management. In this section we focus on subset extraction and query-based approaches, which are the main techniques used today and for which feedbacks have arisen.

First of all, the **extraction of a portionof an ontolog**y is partially realized. Concerning query language based tehcniques, some problems arise with the use of URIs when dealing with ontology blank nodes. An RDF blank node is an RDF node that itself does not contain any data, but serves as a parent node to a grouping of data. From an RDF/XML syntactical standpoint, a blank node is an rdf:Description element that does not have an rdf:about attribute assigned to it. Intermediate results with blank nodes are then not possible for query languages except vSPARQL. Also, the fact that ontology query languages are all based on SPARQL, which was designed for RDF queries, is a brake to easily extract sub-ontologies from OWL ontologies. Indeed the syntax can be complicated to adapt when selecting specific OWL constructors. More generally, the definition of a view with a query is much more difficult than the definition of a mapping with a subset extraction approach. Concerning subset extraction approaches, the precision of the extraction is more or less high depending on the tools provided. For example, a traversal specification will allow to specify a starter concept in the ontology and a depth for the traversal function, so the precision will be limited to this depth. But for a manual subset extraction based on the aggregation of one by one element extractions, the precision level will always be maximal. Inversely a module extraction may produce modules that contain data that the user is not interested in. Precision level is not a problem with a query-based approaches as the query allows to precisely select the elements wanted in the ontology.

Second, the **manageable quality of the portion extracted** depends a lot considering the two main approaches. By manageable, we mean that the sub-graph can be easily accessed and queried (like the ontology), updated (like database views) and evolved (like ontologies). On one hand, updating a view specified by a query, in response to changes in the ontology, just implies updating the query. On the other hand, updating a view specified by a subset extraction approach implies redefining the manual extraction since the beginning and then reapplying the extraction operation on the updated ontology. Subset extraction approaches require significant user effort and must be repeated whenever the information set is updated.

Third the question of the **nature of the portion of ontology extracted** is still open. Does the view have to extract only sub-graphs or sub-ontologies ? Logical and module extraction approaches, because they are based on description logics (DL), guaranty the extraction of a DL sub-ontology on which knowledge can be inferred which is not always the case in query based approaches. Indeed the query techniques are not specifically for deriving subsets of the original ontology that have the same meaning as the original for a specified signature. Some users may want to simply extract lists of terms or small connected sub-graphs and may not need to extract ontologies.

Solutions and Recommendations

The questions of an optimal extraction of portion of an ontology, its management and its nature should be dealt before any agreement is found on the definition of a view.

As we can see issues concerning query based approaches can be resolved by subset extraction approaches, and inversely. An idea could then be to merge the two techniques in a hybrid approach. For instance depending on the level of simplification wanted for the definition of the view, the user could choose to directly compose a query (if he wants a simple sub-graph) or use a graphical subset extractor or module extractor (if he wants a sub-ontology) on the ontology, which would transcript the extraction mapping into a query that could be stored. The result would however depend on the expressivity of the query language chosen. In this case, the definition of the view could be edited and updated directly with the query if required. The sub-graph, if corresponding to a sub-ontology, could nevertheless be updated, evolved and enriched if the user want to consider it as an independent ontology and apply it to a specific application domain. In this case, the link between the source ontology and the evolved sub-ontology would be broken but not the link between the source and the previous one. However, the question of the view dependence to the ontology has to be studied with respect to the use perspectives of each project. To fully cope with the Ontology Change Management requirements, a view should ideally be extracted, managed and evolved as an ontology to allow parallel or branch evolutions of ontologies that can be managed at the same time, reinforcing ontology versioning features.

The recommendations given here are not exhaustive and would deserve more attention namely in this context. We present in the following section the future research directions that should be taken to reach these objectives.

FUTURE RESEARCH DIRECTIONS

From this state of art on ontology views, several future research directions can be envisioned to complete the analysis.

First, the major works in the literature present an ontology view as an extraction of a sub-graph from the source ontology. However this extraction produces a semantic loss because the data are removed from their original context. Also this extracted sub-graph, if considered as sub-ontology, is not just a portion of the ontology extension, like a view in the database, but also a collection of concepts, relationships and is itself a new semantic interpretation of ontology [Rajugan & al.]. So, to bridge this semantic loss, a view needs to be semantically enriched by a new context (custom) or from multiple ontologies. It has to evolve in its own new context.

Second, the views definition, their specifications, the question of their completeness and their evolution are current topics in research that should be further studied. Additionally, ontology change management including maintenance of consistency in particular ([Noy & Musen], [Rajugan & al.]), and ontology versioning is a new question for the sub-ontology views [Noy et al.]. Also, the propagation of changes occurred in the source ontology to the view specification and the extracted sub-graph has been rarely addressed and would complete ontology evolution advances. The question of the propagation of view sub-graphs changes to the source ontology is also open.

Finally, the combination of views as a mechanism to extract subsets of an ontology allowing renaming or reorganization of concepts is crucial in order to encourage the reuse of the ontology. This reuse, in turn, may facilitate the integration problem of ontology (which is a kind of mapping). The aggregation of extracted subsets from different ontologies in order to build a new ontology is another subject to explore in the same context [Rajugan & al.]. Indeed ontologies have to resolve semantic interoperability and can be powered by several sources of distributed data (ontologies). Similarly a view can be an aggregation of data from these sources-ontologies.

CONCLUSION

This document provides a state of the art on ontology views, namely for their introduction in ontology change management methodologies. In the latter, an overview of the corresponding issues and objectives is introduced. The main objective is that views should produce a understandable and manageable portion of an ontology for local applications and users as a means of enabling them to use standard well-developed ontologies. A classification of existing proposals is then proposed respecting their approach type: query based, subset extraction and other techniques. For each approach, a description of the method and an overview of existing work are given. A critical point of view is given on the lacks of the two main approaches and advices are given for their resolution. Problems remain on the extraction the view update capabilities, leading us to present future research directions. The three chosen directions invite researchers to think about the nature of the extracted subset, the change management issues of views and the use of views for ontology integration purposes.

REFERENCES

Klein, M. C. A. (2004). *Change management for distributed ontologies*.

Rajugan, R., Chang, E. and Dillon, T., 'Ontology Views: A Theoretical Perspective', Springer, 2006), pp. 1814-24.

Djedidi, R. (2009). *Approche d'évolution d'ontologie guidée par des patrons de gestion de changement* (Doctoral dissertation, Université Paris Sud-Paris XI).

Detwiler, L.T. Shaw, M. and Brinkley, J.F., (2010) Ontology View Query Management. In *Proceedings, Annual Symposium of the American Medical Informatics Association*, pages 1023, Washington, DC.R.

Noy, N., Kunnatur, S., Klein, M. and Musen, M., 'Tracking Changes During Ontology Evolution', The Semantic Web–ISWC 2004 (2004), 259-73.

ADDITIONAL READINGS

OCMS:

- Luong, P.H. and Dieng- Kuntz, R., 'A Rule- Based Approach for Semantic Annotation Evolution', Computational Intelligence, 23 (2007), 320-38.

- Jaziri, W., Sassi, N. and Gargouri F., 'Approach and Tool to Evolve Ontology and Maintain Its Coherence', International Journal of Metadata, Semantics and Ontologies, 5 (2010), 151-66.
- Rogozan, D.; Paquette, G., "Managing ontology changes on the semantic Web," *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on* , vol., no., pp.430,433, 19-22 Sept. 2005.
- Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D. and Antoniou, G., 'Ontology Change: Classification and Survey', The Knowledge Engineering Review, 23 (2008), 117-52.
- Stojanovic, L. Maedche, A., Motik, B. and Stojanovic, N., 'User-Driven Ontology Evolution Management', Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web (2002), 133-40.
- Castano, S., Ferrara, A. and Montanelli, S., 'Evolving Open and Independent Ontologies', International Journal of Metadata, Semantics and Ontologies, 1 (2006), 235-49.

Views for large ontologies:

- Volz, R., Oberle, D. and Studer, R., "Views for light-weight Web ontologies," Proceedings of the ACM Symposium on Applied Computing (SAC '03), USA, 2003. 1824 R. Rajugan, E. Chang, and T.S. Dillon
- Do, H. H. and Rahm, E., "Flexible integration of molecular-biological annotation data: The genmapper approach," Proceedings of the 9th Int. Conf. on Extending Database Technology (EDBT '04), Heraklion, Crete, Greece, 2004.
- Hadzic, M. and Chang, E. "Role of the Ontologies in the Context of Grid Computing and Application for the Human Disease Studies," Semantics for Grid Databases, First Int. IFIP Conf. on Semantics of a Networked World (ICSNW '04), Paris, France, 2004.
- Ceravolo, P., Damiani, E., Elia, G. and Viviani, M . "Bottom-up Extraction and Maintenance of Ontology-based Metadata," in Fuzzy Logic and the Semantic Web: Elsevier, 2006.
- Wouters, C. "A Formalization and Application of Ontology Extraction," School ofEngineering and Mathematical Sciences Faculty of Sciences, Technology and Engineering, La Trobe University, Melbourne, Australia, Melbourne, Doctor of Philosophy (Ph.D) thesis, 2006. pp. 460.

Ontology Query Languages Views Proposals:

- Karvounarakis G., Alexaki S., Christophides V., Plexousakis D. and Scholl M.. RQL: a declarative query language for RDF; Proceedings, 11th International Conference on World Wide Web; 2002.pp. 592–603.
- RDQL - A Query Language for RDF. Jan 9, 2004. http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/
- SPARQL Query Language for RDF. Jan 15, 2008. http://www.w3.org/TR/rdf-sparql-query/
- Kochut K. J. and Janik M., SPARQLeR: Extended Sparql for Semantic Association Discovery; Proceedings, 4th European conference on The Semantic Web; 2007.pp. 145–159.
- ARQ - A SPARQL Processor for Jena. http://jena.sourceforge.net/ARQ.

- Polleres A., Scharffe F., Schindlauer R., SPARQL++ for Mapping between RDF Vocabularies; Proceedings, 6th International Conference on Ontologies, Databases, and Applications of Semantics; 2007.
- Alkhateeb F., Baget J.F. and Euzenat J.. Constrained Regular Expressions in SPARQL. INRIA; 2007. Research Report 6360.
- Perez J., Arenas M., and Gutierrez C. nSPARQL: A Navigational Language for RDF; Proceedings, International Semantic Web Conference; 2008.pp. 66–81.
- Schenk S. and Staab S. Networked graphs: A declarative mechanism for SPARQL rules, SPARQL views and RDF data integration on the web; Proceedings, 17th International Conference on the World Wide Web; 2008.
- Shaw, M., Detwiler, L.T., Noy, N., Brinkley, J. and Suciu, D., vSPARQL: A view definition language for the semantic web. Journal of Biomedical Informatics. 2011;44:102–117.
- Magkanaraki, A., Tannen, V., Christophides, V. and Plexousakis. D. Viewing the semantic web through RVL lenses. In Second International Semantic Web Conference, volume 2870, pages 96–112, Sanibel Island, FL, 2003. Springer-Verlag.

Ontology Subset Extraction Approaches:

- Obrst, L., Liu, H. and Wray, R., 2003. Ontologies for Corporate Web Applications. Artificial Intelligence Magazine, special issue on Ontologies, American Association for Artificial Intelligence, Chris Welty, ed., Fall, 2003, pp. 49-62. http://portal.acm.org/citation.cfm?id=958676.
- Noy, N. and Musen, M., 'Specifying Ontology Views by Traversal', The Semantic Web–ISWC 2004 (2004), 713-25.
- Seidenberg, J. and Rector, A., Web ontology segmentation: analysis, classification and use; Proceedings of the 15th international conference on World Wide Web.2006. pp. 13–22.
- Kontchakov, R., Pulina, L., Sattler, U., Schneider, T., Selmer, P., Wolter, F. and Zakharyaschev, M., Minimal Module Extraction from DL-Lite Ontologies using QBF Solvers; Proceedings 21st Int. Joint Conf. on Artificial Intelligence (IJCAI); 2009.pp. 836–841.
- Grau B. C., Horrocks I., Kazakov Y. and Sattler U., Just the right amount: extracting modules from ontologies; Proceedings of the 16th international conference on World Wide Web; 2007.pp. 717–726.
- Grau B. C., Horrocks I., Kazakov, Y. and Sattler, U., Modular Reuse of Ontologies: Theory and Practice. Journal of Artificial Intelligence Research (JAIR) 2008;31:273–318.

Other View Specification Approaches

- [Zoltan et al.]
- SWRL: A Semantic Web Rule Language Combining OWL and RuleML. May 21, 2004. http://www.w3.org/Submission/SWRL/
- Miklos, Z., Neumann, G., Zdun, U. and Sintek M., Querying Semantic Web Resources Using TRIPLE Views; Proceedings, International Semantic Web Conference; 2003.pp. 517–532.

- Le-Phuoc, D., Polleres, A., Hauswirth, M., Tummarello, G. and Morbidoni, C., Rapid Prototyping of Semantic Mash-Ups through Semantic Web Pipes; 2009. Proceedings, International Conference on the World Wide Web.
- Jena - A Semantic Web Framework for Java. http://jena.sourceforge.net.
- SPARQLMotion. http://www.topquadrant.com/products/SPARQLMotion.html.
- Banach http://simile.mit.edu/wiki/Banach.
- Sintek M., Van Elst, L., Scerri, S. and Handschuh, S., Distributed Knowledge Representation on the Social Semantic Desktop: Named Graphs, Views and Roles in NRL; Proceedings of the 4th European Semantic Web Conference; 2007.

KEY TERMS & DEFINITIONS

Database View: In database theory, a view consists of a stored query accessible as a virtual table in a relational database or a set of documents in a document-oriented database composed of the result set of a query or map-and-reduce functions. Unlike ordinary tables (base tables) in a relational database, a view does not form part of the physical schema: it is a dynamic, virtual table computed or collated from data in the database.

Materialized Database View: A materialized view is a database object that contains the results of a query. For example, it may be a local copy of data located remotely, or may be a subset of the rows and/or columns of a table or join result, or may be a summary based on aggregations of a table's data.

Ontology: represents knowledge as a set of concepts within a domain, and the relationships among those concepts. It can be used to reason about the entities within that domain and may be used to describe the domain.

Ontology Change Management: process of performing the changes as well as to the process of coping with the consequences of changes.

Ontology Evolution: timely adaptation of an ontology to the arisen changes and the consistent management of these changes.

Ontology Mapping: task of relating the vocabulary of two ontologies that share the same do main of discourse in such a way that the mathematical structure of ontological signatures and their intended interpretations, as specified by the ontological axioms, are respected.

OWL : The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies. The languages are characterised by formal semantics and RDF/XML-based serializations for the Semantic Web.

OWL-DL : OWL DL is designed to provide the maximum expressiveness possible while retaining computational completeness, decidability and the availability of practical reasoning algorithms. OWL DL includes all OWL language constructs, but they can be used only under certain restrictions.

RDF: The Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web. It is particularly intended for representing metadata about Web resources, such as the title, author, and modification date of a Web page, copyright and licensing information about a Web document, or the availability schedule for some shared resource.SPARQL

URI: a Uniform Resource Identifier is a string of characters used to identify a name or a web resource. URI references are used for naming all kinds of things in RDF.

IGI Global Microsoft Word 2007 Template

Reference **templateInstructions.pdf** for detailed instructions on using this document.