Published in final edited form as:

Proc IEEE Int Conf Inf Reuse Integr. 2008 August 5; 2008: 351–356. doi:10.1109/IRI.2008.4583056.

Modular Ontology Techniques and their Applications in the Biomedical Domain

Jyotishman Pathak, Thomas M. Johnson, and Christopher G. Chute

Division of Biomedical Informatics, Mayo Clinic College of Medicine, Rochester, MN 55905, USA

Jyotishman Pathak: pathak.jyotishman@mayo.edu; Thomas M. Johnson: johnson.thomas@mayo.edu; Christopher G. Chute: chute@mayo.edu

Abstract

In the past several years, various ontologies and terminologies such as the Gene Ontology have been developed to enable interoperability across multiple diverse medical information systems. They provide a standard way of representing terms and concepts thereby supporting easy transmission and interpretation of data for various applications. However, with their growing utilization, not only has the number of available ontologies increased considerably, but they are also becoming larger and more complex to manage. Toward this end, a growing body of work is emerging in the area of modular ontologies where the emphasis is on either extracting and managing “modules” of an ontology relevant to a particular application scenario (ontology decomposition) or developing them independently and integrating into a larger ontology (ontology composition). In this paper, we investigate state-of-the-art approaches in modular ontologies focusing on techniques that are based on rigorous logical formalisms as well as well-studied graph theories. We analyze and compare how such approaches can be leveraged in developing tools and applications in the biomedical domain. We conclude by highlighting some of the limitations of the modular ontology formalisms and put forward additional requirements to steer their future development.

1 Introduction

The ability to design and maintain ontologies requires expertise in both the domain of the application and the ontology language used for modeling. However, both these responsibilities become difficult to support, not only cognitively, but also due to lack of tool support as ontologies become larger and more complex. For example, none of the existing reasoners can handle the original OWL version of GALEN¹ and none of the ontology editing environments can load the original version of FMA² within reasonable memory/processing power limits. This severely restricts the re-use, maintainability, evolution, and personalization of ontologies.

Toward this end, research in modular ontologies has drawn significant attention in the recent past. Informally, a module can be considered to be a subset of a “whole” that makes sense (i.e., is not an arbitrary subset randomly built) and can somehow exist separated from the whole, although not necessarily supporting the same functionality as the whole. Thus, in the context of ontologies, a module is a sub-ontology that “makes sense” either from an application (e.g., answering certain queries) or systems (e.g., improving performance) perspective [19]. Modularization can be perceived in two orthogonal ways: independently developing modules that can be integrated coherently and uniformly (*ontology composition*)

¹<http://www.co-ode.org/galen>

²<http://fma.biostr.washington.edu>

or extracting such modules from an integrated ontology for supporting a particular use case (*ontology decomposition*). Addressing both these requirements, in general, is challenging because traditional ontology formalisms such as description logics were primarily designed for single/centralized ontologies rather than multiple/decentralized ones [23]. Furthermore, support for dynamically handling interconnected modules is lacking: if $M_i(O)$ and $M_j(O)$ are modules of an ontology O that are interconnected, how will the updates in $M_i(O)$ reflect changes in $M_j(O)$? Finally, traditional reasoner implementations (such as Pellet³) were developed to operate on a single ontology as opposed to multiple ontologies.

It is to be noted that these issues are highly relevant in the biomedical domain since most of the widely used ontologies such as SNOMED CT⁴ or NCI Thesaurus⁵ are large and complex, and will therefore benefit from the development of tools and techniques that enable proper development and management of multiple, distributed ontologies and corresponding reasoning support. In this paper, we investigate a representative set of approaches for modular ontology languages and formalisms as well as provide use cases from the biomedical domain and additional requirements to steer their future development. While other work [13,19,23] in this topic has focused on rigorously evaluating modular ontology techniques from a theoretical perspective, our objective is not meant to do a comprehensive study of a handful of approaches with rigorous logical proofs. Instead, we survey a wider range of modular ontology proposals and analyze their potential for supporting novel applications to advance biomedical research.

Organization

The rest of the paper is organized as follows: Section 2 formulates the definition of an ontology module within the scope of this paper, Section 3 introduces various modular ontology formalisms, Section 4 discusses use cases and applications in biomedicine where such formalisms can be leveraged, and finally Section 5 concludes the paper with a summary and discussion.

2 Background

Although the notion of a module is well-understood in the software engineering community, discussions on ontology modularization become murky since the concept can be understood in rather different ways. For the purposes of this paper, we adopt the following definition of an ontology module [7]: *An ontology module is a reusable component of a larger or more complex ontology, which is self-contained but bears a definite association to other ontology modules, including the original ontology.* This definition, in addition to implying that modules can be reused “as-is” or extended by introducing new concepts and relationships, asserts that they are not isolated entities or disjoint from each other. For example, assuming that a module is entirely based on subsumption relationships, if a concept A has sub-concepts B and C , such that B and C are disjoint siblings of A , creating a module centered on A would include all three concepts, whereas creating a module on B would only include B , thereby making both the modules not entirely disjoint. However, the general expectation is that modules developed by an ontology engineer will be comprised of distinct concepts rather than concepts that are closely related. In particular, we expect an ontology module to be *self-contained* (i.e., given a set of relations, the ontology module should be transitively closed with respect to those relations) and *logically consistent* (i.e., given an ontology that is logically consistent, every module extracted from it should be logically consistent as well).

³<http://pellet.owldl.com>

⁴<http://www.snomed.org>

⁵<http://nciterns.nci.nih.gov>

More formally, based on [7], assume that $O = (C, R)$ represent an ontology where $C = \{c_1, c_2, \dots, c_n\}$ and $R = \{R_1(C_i, C_j) \dots R_m(c_x, c_y)\}$ is the set of concepts and relationships, respectively. We say that $O_M = (C_M, R_M)$ corresponds to a module of an ontology O such that $O_M \subseteq O$, where $C_M \subseteq C$ and $C_M \neq \emptyset$ and $R_M \subseteq R$. It can be observed that at least $n + 1$ modules can be extract from O , where n corresponds to the number of concepts in C and $n > 1$; for an ontology with only one concept ($n = 1$), a single module can be extracted.

3 Techniques in Modular Ontologies

3.1 Distributed Description Logics

Distributed Description Logics (DDL) [4] is a knowledge representation formalism for representing sets of ontologies and semantic relations between them. DDL captures the idea of importing and reusing concepts between ontologies and enables reasoning with multiple ontologies interconnected by directional semantic mapping (called the *bridge rules*). In particular, if I is a set of non-empty indices, such that $\{O_i\}_{i \in I}$ is a set of ontologies, in DDL, $i: C$ denotes a concept in ontology O_i and $j: C \sqsubseteq D$ represents that concept C is a sub-concept of D in ontology O_j , such that $i: C$ and $j: C$ are different concepts. A bridge rule from i to j is an expression of the form:

- $i: C \xrightarrow{\sqsubseteq} j: D$ (into-rule)
- $i: C \xrightarrow{\sqsupseteq} j: D$ (onto-rule)

where, $i: C$ and $j: D$ are concepts and essentially provide a subjective point of view: an into-bridge rule $i: C \xrightarrow{\sqsubseteq} j: D$ intuitively says that from j -th point of view, the individuals in concept C in i correspond to a subset of the individual in its local concept D . For example, consider fragments of class hierarchies from two simple ontologies shown in Figure 1. The following are some bridge rules that can be defined:

- $\text{Publication} \xrightarrow{\sqsubseteq} \text{Publication}$
- $\text{InProceedings} \xrightarrow{\sqsubseteq} \text{ConferencePaper} \cup \text{WorkshopPaper}$
- $\text{InBook} \xrightarrow{\sqsupseteq} \text{BookArticle}$

3.2 ε -Connections

As can be observed, DDL allows only one type of domain relations. The ε -Connections approach [12], on the other hand, allows multiple “connections” between modules. Informally, an ε -Connection is a set of “connected” ontologies, wherein each ε -Connected ontology, similar to OWL, contains informations about classes, properties and their instances, and additionally new kind of properties called *link properties*. The link properties are logically interpreted as binary relations, where the first element belongs to its source ontology and the second to its target ontology. Thus, in essence, the link properties are used to relate individuals belonging to different ontologies in combination. For example, the concept `GradStudent` from ontology in Figure 1(b) could be defined as a student enrolled in at least one graduate course by using a *someValuesFrom* OWL restriction on the link property *enrolledIn* with the value `GraduateClass` which would be a class in a different ontology about “academic courses”: $\text{GradStudent} \sqsubseteq \exists \text{enrolledIn. GraduateClass}$.

However, ε -Connections require that the ε -Connected ontologies be disjoint from each other thereby enforcing strong restrictions in some applications [23]. For example, a concept cannot be declared as subclass of another concept in a foreign module thereby ruling out the

possibility of asserting inter-module subsumption. Similarly, it does not allow specification of relationships between roles belonging to different ontologies.

3.3 Package-based Description Logics

Package-based Description Logics (P-DL) [2] introduces the notion of “importing relations” to connect local modules. The main idea behind P-DL is the ability to import only commonly shared terms that are interpreted with respect to the overlapping parts of the local module. This is in stark contrast with OWL which only allows the model of an imported ontology to be completely embedded in a global model, thereby hindering *partial reuse* of ontologies. P-DL also enables transitive reuse of knowledge in the sense that if a module (or package as referred in P-DL) $M_i(O)$ asserts that $X \sqsubseteq Y$, and $M_j(O)$ directly imports that axiom from $M_i(O)$, then $X \sqsubseteq Y$ is also valid from $M_j(O)$'s perspective. Furthermore, P-DL facilitates contextualized interpretation of knowledge by ensuring that the interpretation of assertions in each ontology module is constrained by their context. That is, when knowledge in a particular module is reused by another module, the interpretation of the reused knowledge should be constrained by the context in which the knowledge is being reused. As an example, in western countries, the concept weekend typically refers to the days saturday and sunday of a week, whereas in islamic countries it is friday and saturday [1]. Thus, depending on the context, the concept weekend has different interpretations, and it is vital to preserve it. Additionally, P-DL semantics for less expressive logics ensure that both distributed reasoning on modular ontologies and classical reasoning on an integrated ontology comprised of the respective modules will yield the same conclusion; although for more expressive logics this has not been investigated.

3.4 Conservative Extensions

Grau et al. [10] propose the notion of conservative extensions to support partial reuse of ontologies where the objective is to extract from a “foreign ontology” a small fragment that captures the meaning of terms used in a “local ontology”. For example, when building an ontology about research projects, one may use terms such as `CysticFibrosis` and `HeartFailure` from a separate ontology in the description of medical research projects. Conservative extensions ensures that the nature of the modules extracted are as small as possible (ideally), yet guarantee that querying using the modules provide the same answers as if the entire foreign ontology was imported. In particular, the main intuition behind conservative extension is to ensure local completeness of the modules such that the knowledge contained in each individual module is not altered even after their integration. That is, integrating modules cannot induce new relationships between existing concepts in any module. More formally, if O corresponds to the union of ontology modules $\{M_1(O), \dots, M_n(O)\}$, we say that O is a conservative extension of $M_i(O)$ if $O \models \alpha \Leftrightarrow M_i(O) \models \alpha$, such that α is an axiom of the form $X \sqsubseteq Y$, where X and Y are concepts in $M_i(O)$. However, determining whether a particular extension is a conservative extension or not is computationally unsolvable and various approximation techniques, such as the one proposed in [11], have to be employed in practice.

3.5 Graph-based Ontology Segmentation

GALEN Segmentation Service—Seidenberg and Rector [20] developed a methodology for extraction of related concepts from GALEN based on one or more classes given as input by the user. Figure 2 (adopted from [20]) gives a pictorial representation of the approach where starting from a “target” concept, the algorithm extracts all the concepts in the paths to the root and all the leaf concepts. Additionally, any links across the hierarchy from any of the previously traversed classes are followed, and the hierarchy is traversed upwards (but not downwards) from any of these classes that the cross-links point to. Links pointing at

other classes from these newly traversed classes are also included. This continues until there are no more links left to follow. [20] shows that using their approach, the size of GALEN was reduced by a factor of 20 and the ontology segments extracted could be classified within seconds. However, it is unclear how the algorithm can be generalized and applied to other ontologies apart from GALEN.

ModTool: A Module Extraction Tool—Doran et al. [7] propose another technique for extracting ontology modules based on a graph/hierarchy traversal approach similar to the GALEN segmentation service. However, unlike [20], the algorithm does not allow upward navigation of the subclass hierarchy from the target concept based on the justification that such a traversal will substantially increase the probability of extracting modules that are as large as the whole ontology. Furthermore, the technique ensures that the modules extracted are transitively closed with respect to the relations that are traversed. A preliminary evaluation based on the NCI Oncology ontology showed that this approach generated modules of smaller size compared to those generated by ε -Connections [12].

Modularization via Dynamic Selection—Similar to above approaches, [5] develops another mechanism for modularization that relies on exploiting the hierarchical relationships in an ontology. Given a set of input terms, the objective is to extract the smallest part of the ontology covering those terms via a fixed-point algorithm. In particular, the algorithm recursively inspects the ontology expressions to include elements that participate in the definition of the input terms. Although, unlike [20], instead of including all super-concepts of a selected concept, only the most specific common super-concepts are included. Furthermore, the algorithm removes from the hierarchy all the intermediate concepts that do not participate in the semantic definition of the considered terms, and only retains the hierarchical structure. While no rigorous evaluation was presented in [5], the authors claim satisfactory results.

PATO: Partitioning Tools for Ontologies—Stuckenschmidt and Klein [22] propose an approach for partitioning light-weight ontologies (primarily class hierarchies) into disjoint and covering sets of concepts. Their approach is based on extracting a weighted dependency graph, $D_G = \langle C, D, w \rangle$, from the class hierarchy, where C represents the set of concepts, and links D between the concepts represent different types of dependencies that can be derived from the ontology and weighted according to the strength of the dependency. The weights, in particular, give an estimate of connectivity between concepts and is used to determine sets or clusters of concepts that are strongly inter-connected. The authors perform experiments on real-world ontologies such as NCI Thesaurus and generated modules with an average size of 57 concepts, although with a larger variance as module sizes ranged between 4 and 268 concepts.

Prompt Traversal Views—The Prompt Traversal View [18] is a Protégé plugin that allows a user to select target concept(s) of interest, and extract additional concepts based on relationships that the user wants to explore until a particular traversal depth is reached. It allows a user to tweak various additional traversal parameters manually and is essentially a tool that is targeted to assist users in manually extracting and inspecting smaller parts of the ontology. While useful for this specific use case, it is unclear how the tool can be applied for (semi-) automatic ontology module extraction. Also, experimental results were not presented in [18] which makes it harder to judge the performance of Prompt compared to other tools, especially when dealing with large ontologies.

4 Applications of Modular Ontologies in Biomedicine

The domain of life sciences, biology and medicine in particular, have a strong tradition of structuring their terminological knowledge in terms of controlled vocabularies, thesauri or classifications. Such scientific vocabularies have been successfully and widely used in various applications ranging from clinical trials to patient billing. However, at the same time, it has become apparent that the challenges for proper construction, management and usage of biomedical ontologies is far from trivial as initially expected, in part due to their sheer size and complexity. We believe that research in modular ontologies is poised to address some of these challenges and can be leveraged for building novel applications. We outline a few of them in the following.

Ontology Reuse

Knowledge-based systems that support applications such as decision support in healthcare typically depend on large amounts of domain knowledge. However, capturing domain knowledge, in the form of ontologies, is expensive and reuse is always encouraged. In particular, the ability to *partially* reuse ontologies is vital because for many applications only specific ontology segments are relevant as opposed to the entire ontology. However, existing ontology languages such as OWL only support importing the whole ontology. As illustrated earlier, modular ontology techniques such as P-DL natively support partial ontology import with contextualized semantics, thereby ensuring the inferences are drawn from the point of view of the (local) ontology importing (foreign) ontologies.

Ontology Alignment

Ontology alignment is the process of determining correspondences between multiple concepts in multiple ontologies. Over the last several years, enormous amounts of effort have been invested in aligning and harmonizing biomedical ontologies. However, manually aligning ontologies is a labor intensive process and often semi-automatic approaches [6,15] are sought. Nevertheless, in spite of many improvements, such systems generate incorrect mapping to varying degrees. Consequently, to avoid incorrect utilization of mappings (e.g., for query answering) it is vital to detect and fix defective mapping elements. Recently, techniques [16] based on DDL have been proposed to address this problem where logical reasoning is used to analyze the impact of mapping to ontologies it connects where the assumption is that a mapping, if incorrect, will cause inconsistency (unsatisfiability) in mapped ontologies.

Value Set Construction

In the context of vocabularies, a value set is a uniquely identifiable set of valid values that can be resolved at a given point in time to an exact set (collection) of codes. The main objective of modeling value sets is to specify a *concept domain* with certain *attributes* of interest such that the attribute-values can be obtained from one or more vocabulary of interest. An example of a concept domain could be “world countries”, and the representative value set will include countries such as USA and UK. In practice, these value sets are constructed manually from pre-existing ontologies such as SNOMED and ICD by constraining the value selection based on logical expressions (e.g., all sub-concepts of the concept ColonCancer) which, arguably, is a tedious and cumbersome process. We believe that all the techniques for ontology segmentation introduced earlier can be leveraged for semi-automatic value set construction. In particular, given a set of target concepts, the algorithms in [5,7,20] can act as a guide in traversing the hierarchy and select super-, sub-, sibling-concepts etc., and extract an initial value set that can be validated and refined by a domain expert.

Secure Information Exchange

Addressing issues related to privacy and selective sharing of information in the biomedical domain is of utmost importance. In particular, the ability to develop techniques for privacy-preserving query answering with ontologies, where the objective is to enable “selective” answering of queries against ontologies, will be of immense benefit in a healthcare setting. Toward this end, recent research in privacy-preserving reasoning with hidden knowledge using P-DL [3] has shown promising results. The crux of the technique is to divide an ontology into two mutually exclusive parts, visible and hidden, and answering queries by inferencing on both the visible and hidden parts, however at the same time ensuring that the hidden knowledge is never revealed. The approach is based on an Open World Assumption, and hence, a query that cannot be answered without disclosing the hidden knowledge to the reasoner will be answered such that the reasoner is lacking the complete information to answer the query. While there are other approaches [8] that also prevent unwanted inference, however they are based on a Closed World Assumption which make them overly restrictive for querying purposes.

Collaborative Authoring

The process of building large ontologies such as SNOMED is often collaborative and involves a team of experts that communicate and reconcile their changes. Since their objective is to follow principled approaches to generate, manage, and integrate multiple components of the ontology, there is a requirement to minimize the impact of changes performed by a modeler on other parts of the ontology developed by other modelers. We believe that this is a major application area for modular ontologies where individual ontology modules can be extracted (ontology decomposition), modified and updated, and then combined to form an integrated ontology (ontology composition). While other techniques for collaborative ontology authoring, such as Collaborative Protégé [17] and Semantic Wiki [14] exists, they cannot support modular representation of an ontology, and hence lack a principled way for controlling conflicts in editing or propagation of editing errors.

Distributed and Incremental Reasoning

In practice, various biomedical ontologies are autonomously created, maintained and interlinked/reused within different application contexts. This results in the formation of a distributed information resource thereby necessitating the support for distributed reasoning (as opposed to global reasoning) to address scalability and context-specific nature of the ontologies involved. Recently, distributed reasoning techniques such as DRAGO [21] based on DDL have been proposed where reasoning with multiple ontologies is accomplished by leveraging the bridge rules.

On a different note, modular ontology formalisms are playing a key role in providing support for incremental reasoning with changing knowledge bases [9]. The idea behind incremental classification is to persist the reasoner state such that in the event of an update, the previous classification tree can be reused. Arguably, if the classification tree can be partitioned into modules, the updates can be applied to the targeted module(s) and changes propagated to other relevant module(s). Such an approach is highly relevant in dealing with ontologies such as the Gene Ontology (GO) which are revised constantly since loading a new version of GO can easily be handled as an incremental update to the previous version.

Scalable Querying

As more and more instance data gets tagged with concepts from ontologies (e.g., in biomedical grids), the ability to efficiently and effectively support ontology-based querying

is pivotal. Since reasoning is required for instance-level querying (e.g., if $X \sqsubseteq Y$ and $x \in X$, where X, Y are classes and x is an instance of X , querying for all instances of Y should also return x), modularized ontology reasoning support becomes important. Specifically, the ability to compute modules on-demand based on the queries asked and classifying such modules on-the-fly will allow one to efficiently reason about the relationships that may exist among the instance data resident in the data nodes. Such reasoning could be used for example to determine if data tagged with a given concept code is also a member of the set of instances belonging to other concept(s). However, to the best of our knowledge, none of the tools can provide such a support for querying based on modularized ontology reasoning.

5 Discussion

As new ontologies in the biomedical and life sciences domain begin to populate and existing ones continue to grow and become more complex, there is an increasing need to leverage, develop and expand modular ontology formalisms for a wide variety of applications. In this paper, we have briefly introduced a representative set of such approaches and illustrated key potential applications which could benefit from such advances. With their increasing maturity, we believe that modular ontology languages and tools are poised to address various challenges in the development, management and usage of large biomedical ontologies.

However, at the same time, there are many potential areas of future development that have to be pursued for modular ontology approaches to become mainstream. For instance, even though the techniques for ontology segmentation based on graph traversal algorithms [5,7,20] are efficient and useful in practice, they have the risk of generating incomplete results since the hierarchy traversal is done without considering the semantics of the underlying ontology language. This is in particular true for OWL ontologies because depending on the presence or absence of OWL defined and primitive classes in a particular ontology, its asserted class hierarchy will be different compared to an inferred hierarchy (obtained after classification). Since the traversal algorithms cannot make a distinction between defined and primitive classes, they ignore classes and relationships between them. Also, as mentioned earlier, all the modularization techniques focus on generating modules from the ontology and reasoning on them (known as TBox modularity). It is unclear what role modular ontology formalisms will play when dealing with instance data (known as ABox modularity). Addressing this issue is important because many applications (e.g., distributed data retrieval in a grid) require interaction with multiple data nodes storing instance information. Another shortcoming of modular ontology formalisms is the lack of a query language, unlike SPARQL which is the query language for RDF graphs. Just like SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware, it should be feasible to query distributed ontology modules (TBoxes) and corresponding instance data (ABoxes) in a principled way. Finally, since many modular ontology languages such as DDL and ε -Connections leverage mappings or bridge rules, the ability to detect, and potentially rectify, inconsistent mappings is vital. While approaches such as [16] have been proposed for debugging DDL bridge rules, there is a requirement to expand such work in a more general setting that will enable resolution of inconsistencies between ontology modules that are interlinked.

References

1. Bao, J. PhD thesis. Department of Computer, Iowa State University; Ames, Iowa, USA: 2007. Representing and Reasoning with Modular Ontologies.

2. Bao, J.; Slutzki, G.; Honavar, V. A Semantic Importing Approach to Knowledge Reuse from Multiple Ontologies. 2nd AAAI Conference on Artificial Intelligence; 2007. p. 1304-1309.
3. Bao, J.; Slutzki, G.; Honavar, V. Privacy-Preserving Reasoning on the Semantic Web. IEEE/ACM/WIC International Conference on Web Intelligence; 2007. p. 791-797.
4. Borgida A, Serafini L. Distributed description logics: Assimilating information from peer sources. *Journal of Data Semantics*. 2003; 1:153–184.
5. d'Aquin, M.; Sabou, M.; Motta, E. Modularization: A Key for the Dynamic Selection of Relevant Knowledge Components. 1st International Workshop on Modular Ontologies; 2006.
6. Dinakarpandian D, Tong T, Lee Y. A Pragmatic Approach to Mapping the Open Biomedical Ontologies. *International Journal of Bioinformatics Research and Applications*. 2007; 3(3):341–365. [PubMed: 18048196]
7. Doran, P.; Tamma, V.; Iannone, L. Ontology Module Extraction for Ontology Reuse: An Ontology Engineering Perspective. 16th ACM International Conference on Information and Knowledge Management; 2007. p. 61-70.
8. Farkas C, Brodsky A, Jajodia S. Unauthorized Inferences in Semistructured Databases. *Information Science*. 2006; 176(22):3269–3299.
9. Grau, BC.; Halaschek-Wiener, C.; Kazakov, Y. History Matters: Incremental Ontology Reasoning Using Modules. 6th International Semantic Web Conference; 2007. p. 183-196.LNCS 4825
10. Grau, BC.; Horrocks, I.; Kazakov, Y.; Sattler, U. A Logical Framework for Modularity of Ontologies. 20th International Joint Conference on Artificial Intelligence; 2007. p. 298-303.
11. Grau, BC.; Horrocks, I.; Kazakov, Y.; Sattler, U. Just the Right Amount: Extracting Modules from Ontologies. 16th International Conference on World Wide Web; 2007. p. 717-726.
12. Grau BC, Parsia B, Sirin E. Combining OWL Ontologies using epsilon-Connections. *Journal of Web Semantics*. 2006; 4(1):40–59.
13. Grau, BC.; Parsia, B.; Sirin, E.; Kalyanpur, A. Modularity and Web Ontologies. 10th International Conference on Principles of Knowledge Representation and Reasoning; 2006. p. 198-209.
14. Krötzsch M, Vrandečić D, Völkel M, Haller H, Studer R. Semantic Wikipedia. *Journal of Web Semantics*. 2007; 5(4):251–261.
15. Lambrix P, Tan H. SAMBO: A System for Aligning and Merging Biomedical Ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*. 2006; 4(3):196–206.
16. Meilicke, C.; Stuckenschmidt, H.; Tamin, A. Repairing Ontology Mappings. 22nd AAAI Conference on Artificial Intelligence; 2007. p. 1408-1413.
17. Noy, NF.; Chugh, A.; Liu, W.; Musen, MA. A Framework for Ontology Evolution in Collaborative Environments. 5th International Semantic Web Conference; 2006. p. 544-558.LNCS 4273
18. Noy, NF.; Musen, MA. Specifying Ontology Views by Traversal. 3rd International Semantic Web Conference; 2004. p. 713-725.LNCS 3298
19. Rector AL, Napoli A, Stamou G, et al. Report on Modularization of Ontologies. Technical report, Knowledge Web Deliverable D2.1.3.1. 2005
20. Seidenberg, J.; Rector, AL. Web Ontology Segmentation: Analysis, Classification and Use. 15th International Conference on World Wide Web; 2006. p. 13-22.
21. Serafini, L.; Tamin, A. DRAGO: Distributed Reasoning Architecture for the Semantic Web. 2nd European Semantic Web Conference; 2005. p. 361-376.LNCS 3532
22. Stuckenschmidt, H.; Klein, MCA. Structure-Based Partitioning of Large Concept Hierarchies. 3rd International Semantic Web Conference; 2004. p. 289-303.LNCS 3298
23. Wang, Y.; Bao, J.; Haase, P.; Qi, G. Evaluating Formalisms for Modular Ontologies in Distributed Information Systems. 1st International Conference on Web Reasoning and Rule Systems; 2007. p. 178-193.LNCS 4524

Figure 1.
Sample Class Hierarchies

Figure 2.
GALEN Segmentation Algorithm [20]