Partitioning of ontologies driven by a structure-based approach

F. Amato, A. De Santo, V. Moscato, F. Persia, A. Picariello and S.R.Poccia
Dip. di Ingegneria Elettrica e Tecnologie dell'Informazione
University of Naples "Federico II"
via Claudio 21, 80125, Naples, Italy
Email: {flora.amato, aniello.desanto, vmoscato, fabio.persia, picus, silvestroroberto.poccia}@unina.it

***Abstract*—In this paper, we propose a novel structure-based partitioning algorithm able to break a large ontology into different modules related to specific topics for the domain of interest. In particular, we leverage the topological properties of the ontology graph and exploit several techniques derived from Network Analysis to produce an effective partitioning without considering any information about semantics of ontology relationships. An automated partitioning tool has been developed and several preliminary experiments have been conducted to validate the effectiveness of our approach with respect to other techniques.**

I. Introduction

Nowadays, the pursuit of fast progresses in many branches of science requires increasingly detailed descriptions of large amounts of concepts and relationships. This desire for comprehensive information related to extended real world domains has led to large ontologies (e.g., medical and bioinformatics ontologies or other domain-specific applications, FOAF, DBpedia and other examples from Semantic Web, etc.) increasing importance for knowledge engineering and information processing. However, the growing size and monolithic nature of these ontologies originate new and previously unexplored problems, especially when it comes to the development of an ontology related to complex domains, to the difficulty of designing adequate quality control procedures, and to scalability problems and reasoning complexity [1].

Since the origins of such problems seem to be reducible to the fact that domain-comprehensive ontologies are just too large to be handled effectively, recent works [2] have suggested to dissemble the overall models into a subset of smaller *modules*, each focused around a specific sub-topic of interest. Interestingly, while maintaining knowledge of its connection with the other sub-parts of the ontology, each module can easily be used independently from the others, thus providing obvious benefits to the information processing and ontology maintenance burden.

Exploiting such an idea, in this work we present a method for the structure-based partitioning of a large ontology into a set of topic-centered sub-modules. Intuitively a well-built module will contain information about a sub-topic that can stand coherently by itself. This requires the concepts within a module to have strong semantic connections to each other while lacking strong dependencies with information outside the module.

We start from converting an ontology into a *weighted graph*, where certain elements (e.g. subjects, verbs and objects) are nodes. Links between these nodes are derived from the definitions and axioms existing in the ontology. While previous works rely on a sub-class hierarchy in order to differentiate between the relationships inside the ontology, our method is an attempt to make the partitioning more generic and completely automated, without the need of pre-assigning weights to the hierarchical relationships typical of each ontology. Working on the ontology graph, we exploit techniques derived from network analysis to identify important concepts in the ontology, evaluate the degree of dependency between these concepts, and therefore find sets of both related and unrelated concepts and finally identify unrelated parts of the original ontology. In the end, we want to obtain a partitioning of an ontology into a number of disjoint sets of concepts which - if considered as a whole - would result semantically isomorphic to the original ontology.

The paper is organized as in the following. Section 2 reports the related work on the ontology partitioning problem. Section 3 describes the proposed approach and illustrates the developed partitioning tool. Finally, Sections 4 and 5 discuss the preliminary experimental results and some conclusions and future work.

II. Related Work

Due to their extensive use for information processing and knowledge engineering in different domains, ontologies have grown into large, complex collections of thousands of concepts and ontology definition techniques are drawing considerable attention from researchers world-wide [3], [4]. In order to support maintenance and reusability, it has been recently proposed that the structure of a large ontology should be based on the combination of self-contained, independent and reusable knowledge components.

This way of structuring ontologies following *modularity* principles should come easy to ontology engineers, since the implicit idea on which large ontologies are built is to relate several sub-domains. However, at the time being most ontologies are not structured in a modular way, and therefore modularization techniques able to identify and extract significant modules from existing ontologies are becoming

IEEE ICSC 2015, February 7-9, 2015, Anaheim, California, USA
978-1-4799-7935-6/15/$31.00 ©2015 IEEE

essential not just to ontologies' management, but also to their exploration [5].

Moreover, a distributed computing environment could leverage the obtained modules to perform parallelized search or reasoning tasks on the ontology [6].

In this section, we describe recent works related to ontology modularization. Although existing approaches can be divided into several categories [7]–[9], here we chose to focus on *module extraction* and *ontology partitioning* techniques.

The first kind of approach is based on the idea of reducing an ontology (i.e., *segmentation* or *traversal view extraction*) to the sub-part that covers a particular sub-vocabulary, related to a specific topic. Following this idea, the authors in [10] use a set of classes of the input ontology and extract related elements on the base of specific properties and restrictions; optional filters can also be activated to reduce the size of the resulting module.

Noy *et al.* [11] present *Traversal Views* as a way of defining an ontology view: a user specifies a subset of an ontology to include in the view by defining the starter concepts, the links to follow from those concepts, and how deep into the ontology the search should go on. Then, starting from the elements of the input sub-vocabulary, relations in the ontology are recursively *traversed* to gather relevant elements.

Similarly to [10] and [11], the authors in [12] define a method for the dynamic selection of relevant modules from on-line ontologies. Here, the input sub-vocabulary can contain either classes, properties, or individuals. The mechanism is fully automatized and designed to work with different kinds of ontologies (from simple taxonomies to rich and complex OWL ontologies), and relies on inferences during the modularization process. In [9], users can extract a module from the original ontology according to a semantic query.

In the second kind of approach, partitioning an ontology corresponds to the process of splitting up the set of axioms into a set of modules $\{M_1, ..., M_k\}$ such that each M_i is an ontology and the union of all modules is semantically equivalent to the original ontology O. In [1] the partitioning is accomplished through the ontology graph structure enriched by assigning different weights to the different relationships. The weighting is performed according to the priority and meaning of the relationships and the random walk algorithm is then adopted to obtain final partitions.

Stuckenschmidt and Klein [13] use the previous assumption that dependencies between concepts can be derived from the structure of the ontology. In their approach, an ontology graph is built though the extraction of dependencies resulting from the subclass hierarchy and some additional domain restrictions; then, they exploit connections among nodes to assign the weights. Finally, in order to obtain the final partitioning, they define a modularization algorithm called *island* based on the minimum cut principle that was implemented in *PATO* [1].

In all these approaches, the representation of the ontology is central to the modularization method. Interestingly, the authors in [14] propose a graph representation alternative to directed labeled graphs that scales well even for complex cases, particularly regarding the central notion of connectivity of resources.

They introduce the concept of *RDF Bipartite Graph* and show its advantages as intermediate model between the abstract triple syntax and data structures used by applications. In the light of this model they explore the issues of transformation costs, data/schema- structure, and the notion of RDF connectivity.

III. The proposed Ontology Partitioning approach

The aim of this work is to develop a new partitioning technique based on graph representation of an ontology exploiting the related topological properties. Although there exist several possible representations for an ontology graph, we chose to treat an ontology as a network, where each element of an *RDF triple* represents a separate node in the graph.

Links between nodes are then weighted by computing the related frequency in the graph. We can identify three consecutive steps in our method: *ontology-graph transformation*, *edges' weight computation* and *graph partitioning*.

A. Ontology Graph

Let us consider an RDF description of a generic ontology. We suppose that a weighted and directed graph $G = (V, E, \omega)$ can be extracted from the ontology, where:

- V is the finite set of the graph vertices - each vertex v represents an element of the ontology;
- $E \subseteq V \times V$ is the set of directed edges - two vertices are connected by an edge if the corresponding elements are related within one or more triples in the ontology;
- $\omega : E \to \mathbb{N}^+$ is a function assigning a weight w_{ij} to each edge $e_{ij} = \langle v_i, v_j \rangle$.

Intuitively, the weight of a connection between two nodes v_i and v_j describes the importance of the link from one node to other. Thus, the weight w_{ij} for the edge connecting two nodes v_i, v_j is computed as the number of direct relationships c between the two related ontology elements divided by the maximum number of global relationships shared by each of the two nodes with every other node in the ontology:

$$\omega_{ij} = \frac{c_{ij} + c_{ji}}{max(\sum_k c_{i,k} + c_{k,i}, \sum_k c_{j,k} + c_{k,j})} \qquad (1)$$

We computed the degree of relations between concepts focusing just on the structure of the graph. Note that, according to Equation 1, we need no prior knowledge of the semantics of the relationships between nodes, nor of the strength of the dependencies between concepts in the specific ontology we are partitioning.

We can also observe that if in the original ontology there is more than one directed link between two nodes, then these links are combined into a single weighted edge of the graph.

Figure 1 shows how a part of an ontology graph can be generated using the set of triples (in the *turtle* format) related to the *Kennedy's Family Ontology* [1].

[1] The Kennedys Ontology is available at http://topbraid.org/examples/kennedys

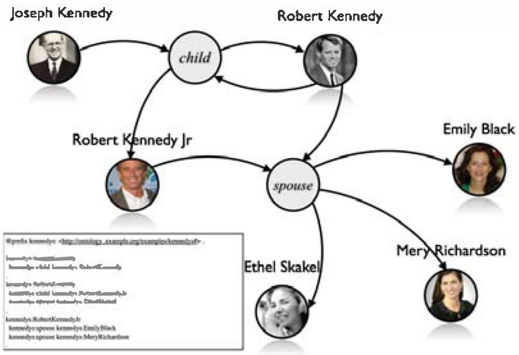

Figure 1: Example of a graph ontology.

B. Graph Partitioning

The weighted graph we just built provides us with a foundation for detecting sets of strongly related concepts. To this goal, we decide to use a multilevel *k-way* partitioning schema: the related aim is to compute k partitions minimizing the *edge-cut*, meaning that we search for a partitioning such that the number of edges (or, in case of a weighted graph, the sum of their weights) crossing different partitions is minimized. The k-way partitioning is well suited for our needs, determining sets of concepts that present strong internal connections and weak external ones.

For the implementation, we leverage the **METIS**[2] libraries. However, while METIS expects the number of modules the graph has to be partitioned into to be known, we want to obtain such number based on ontology features. Thus, we implemented a procedure recursively using METIS partitioning method to iteratively increase the number of modules (starting from an initial value k which depends from the number of the ontology's concepts).

The partitioning procedure aims at minimizing *bulkiness* of each module M_i and maximizing the related *connectedness* as suggested in [15] using a proper heuristics.

Exploiting these optimization criteria, we define the bulkiness value for each module M_i as:

$$bulk_i = \frac{1}{2} - \frac{1}{2}\cos(\pi \cdot \frac{n_i}{n}) \qquad (2)$$

n being the number of nodes and n_i the number of nodes of a module M_i.

Similarly, we define the connectedness of a module M_i as the number of edges connecting M_i to other modules divided by the total number of edges in that module:

$$conn_i = \frac{\#\{(v,v') \in M_i \mid M(v) \neq M(v')\}}{\#\{(v,v') \in M_i\}} \qquad (3)$$

where (v, v') is an edge of the graph connecting nodes v and v', and $M(v)$ returns the module which the vertex v is assigned to.

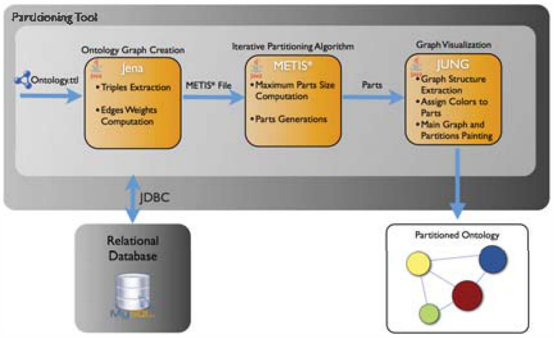

Figure 2: Partitioning Tool Architecture

The number of actual elements in each module can be deduced by the number of the corresponding vertices. Eventually, a *label* can be associated to a module considering the label of the vertex having the highest *betweenness*:

$$betw(v) = \sum_{s,t \in V \, s \neq v \neq t} \frac{e^*_{st}(v)}{e^*_{st}} \qquad (4)$$

where $s, t, v \in V$, $s \neq v \neq t$ and e^*_{st} is the number of shortest paths between s and t, passing through v.

C. The Partitioning Tool

The partitioning procedure described in the previous section was implemented through a proper tool using JAVA. Figure 2 shows an outline of the overall system architecture and of the related workflow.

Starting from an ontology input file, the first step is to generate a graph structure which can be used as input for METIS. To such a goal, we exploit **Jena**[3] facilities to convert ontologies represented in the form of RDF, OWL or other RDF serialization formats in a set of triples (stored in a RDBMS) and successively into the related ontology graph.

The graph generated by Jena is then weighted and partitioned into modules following the iterative approach previously described. Finally, to efficiently analyze the results of the partitioning, we need to visualize the original graph and modules into which it was divided. To this aim, we used both **Jung** API[4] and **Pajek** [16] visualization tool.

IV. Preliminary Experimental Evaluation

In order to evaluate the quality of our partitioning method, we present in this section a preliminary experimental set-up conducted on the Kennedy's Family ontology.[5]. Given the ontology and the modules derived from it through the partitioning procedure, we adopted both an *empirical*[6] and a *criteria-based*[7] evaluation to determine the quality of partitioning.

[2] http://glaros.dtc.umn.edu/gkhome/metis/metis/overview

[3] http://jena.apache.org/

[4] http://jung.sourceforge.net

[5] The ontology consists of 619 triples, amounting to a total of 282 nodes and 748 edges

[6] A partitioning generated using our automatic tool is evaluated against a ground truth partitioning built by human experts, in terms of *recall* and *precision*.

[7] The partitioning quality is evaluated according to some *criteria* which can be classified as logic-based, structural and application-dependent.

First of all, we collected a number of students and we made them analyze the *Kennedys* ontology in order to build what we will consider the *optimal* partitioning of the ontology [8]. Then, we defined three similarity measurements: *precision*, *recall* and *F-Measure*. These measures are based on the numbers of *intra-pairs*, which are pairs of concepts (subject-verb or verb-object) belonging to the same module.

More formally:

- **Precision**: is the ratio of intra-pairs in the generated partitioning that are also intra-pairs in the optimal partitioning.
- **Recall**: is the ratio of intra-pairs in the optimal partitioning that are also intra-pairs in the generated one.
- **F-Measure**: is a value used to to point out the overall quality results

In Table I, we compared the obtained results with a partitioning performed by *PATO* [9].

	Precision	**Recall**	**F-Measure**
PATO	53.84%	50%	52%
Proposed Approach	88.2%	91.4%	89%

Table I: Precision and Recall Comparison

The problem of the empirical evaluation is to obtain a reliable optimal partitioning to be used as comparison when we face the analysis of large and complex ontologies. However, it is possible to rely on an alternative method exploiting criteria descriptive of the quality of the given partitioning. In our evaluation, we used the parameter of *global connectedness* again defined in [15][10].

Again, we compared the results with those obtained by *PATO*. The modules produced by PATO vary significantly in size, the *connectedness* values of the modules are heavily variable and the value of *global connectedness* is significantly higher than our method's one as specified in the Table II.

	Proposed Method	**PATO**
Number of Modules	3	4
Number of Nodes	282	147
Smallest Module Size	16	8
Largest Module Size	65	83
Global Connectedness	**1.01**	**10.3**

Table II: Structural Comparison

V. Conclusions and Future Work

In this paper we described a method for structure-based ontology partitioning. The main idea of our approach is to translate the structure of an ontology into a weighted graph and to break it into a set of modules which have stronger internal connections than external ones. A preliminary experimental evaluation was conducted on the *Kennedy's Family* ontology. The results were validated by comparing them both with a ground truth generated by humans and with the results obtained by *PATO* partitioning tool. We obtained encouraging results, both in terms of precision and recall, and of internal coherence of the obtained modules. Future work will be devoted to improve the quality of the partitioning, for example employing other graph partitioning techniques, and to extend our experiments using larger ontologies.

References

[1] H. Stuckenschmidt and A. Schlicht, "Structure-based partitioning of large ontologies," in *Modular Ontologies* (H. Stuckenschmidt, C. Parent, and S. Spaccapietra, eds.), vol. 5445 of *Lecture Notes in Computer Science*, pp. 187–210, Springer Berlin Heidelberg, 2009.

[2] G. Asieh and A. Hassan, "Partitioning large ontologies based on their structures," *International Journal of Physical Sciences*, vol. 7, no. 40, pp. 5545–5551, 2012.

[3] V. Moscato, A. Penta, F. Persia, and A. Picariello, "Mowis: A system for building multimedia ontologies from web information sources.," in *IIR*, pp. 89–93, 2010.

[4] A. Chianese, V. Moscato, F. Persia, and C. Sansone, "A framework for building multimedia ontologies from the web," in *A Framework for Building Multimedia Ontologies from Web Information Sources*, pp. 83–90, SEBD 2012, 2012.

[5] M. d'Aquin, A. Schlicht, H. Stuckenschmidt, and M. Sabou, "Ontology modularization for knowledge selection: Experiments and evaluations," in *Database and Expert Systems Applications* (R. Wagner, N. Revell, and G. Pernul, eds.), vol. 4653 of *Lecture Notes in Computer Science*, pp. 874–883, Springer Berlin Heidelberg, 2007.

[6] C. Molinaro, V. Moscato, A. Picariello, A. Pugliese, A. Rullo, and V. Subrahmanian, "Padua: Parallel architecture to detect unexplained activities," *ACM Transactions on Internet Technology (TOIT)*, vol. 14, no. 1, 2014.

[7] T. Özacar, Ö. Öztürk, and M. O. Ünalır, "Anemone: An environment for modular ontology development," *Data & Knowledge Engineering*, vol. 70, no. 6, pp. 504–526, 2011.

[8] M. Abadi and K. Zamanifar, "Producing complete modules in ontology partitioning," in *Semantic Technology and Information Retrieval (STAIR), 2011 International Conference on*, pp. 137–143, June 2011.

[9] S. Ghafourian, A. Rezaeian, and M. Naghibzadeh, "Graph-based partitioning of ontology with semantic similarity," in *Computer and Knowledge Engineering (ICCKE), 2013 3th International eConference on*, pp. 80–85, Oct 2013.

[10] J. Seidenberg and A. Rector, "Web ontology segmentation: Analysis, classification and use," in *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, (New York, NY, USA), pp. 13–22, ACM, 2006.

[11] N. Noy and M. Musen, "Specifying ontology views by traversal," in *The Semantic Web Conference ISWC 2004* (S. McIlraith, D. Plexousakis, and F. van Harmelen, eds.), vol. 3298 of *Lecture Notes in Computer Science*, pp. 713–725, Springer Berlin Heidelberg, 2004.

[12] M. d'Aquin, M. Sabou, and E. Motta, "Modularization: a key for the dynamic selection of relevant knowledge components," in *Workshop on modular ontologies, WoMO 2006, Athens*, 2006.

[13] H. Stuckenschmidt, "Network analysis as a basis for partitioning class hierarchies," 2006.

[14] J. Hayes and C. Gutierrez, "Bipartite graphs as intermediate model for rdf," in *The Semantic Web–ISWC 2004*, pp. 47–61, Springer, 2004.

[15] A. Schlicht and H. Stuckenschmidt, "Towards structural criteria for ontology modularization," in *In: Proc. of the ISWC 2006 Workshop on Modular Ontologies*, 2006.

[16] V. Batagelj and A. Mrvar, "Pajek – analysis and visualization of large networks," in *Graph Drawing Software*, pp. 77–103, Springer, 2003.

[8] Humans identified three main sub-topics around which the analyzed ontology is focused: Professional Career, Vital Statistics and Degree if Kinship.

[9] http://web.informatik.unimannheim.de/anne/Modularization/pato.html

[10] The global connectedness is defined as the fraction of inter-modules edges compared to the total number of edges in the ontology graph.