STAG: Smart Tools & Apps for Graphics (2015)
Silvia Biasotti and Marco Tarini and Andrea Giachetti (Editors)

# Grontocrawler: Graph-Based Ontology Exploration

A. Agibetov[1], G. Patanè[1] and M. Spagnuolo[1]

[1]Consiglio Nazionale delle Ricerche, Genova, Italy

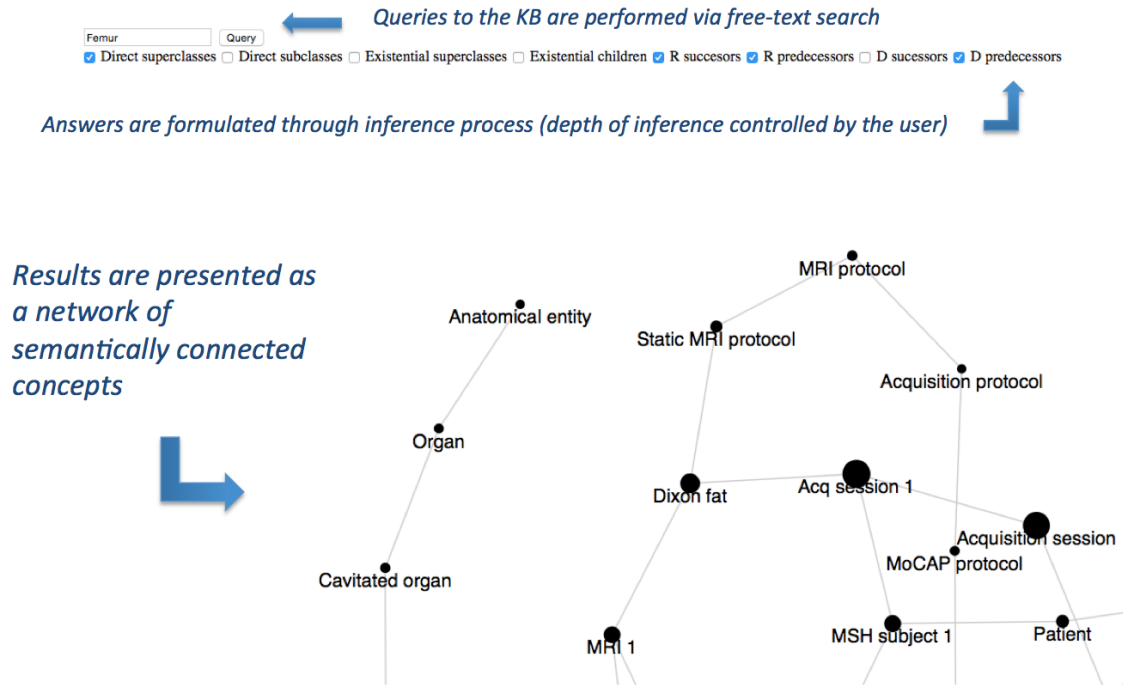


**Figure 1:** *User interface of Grontocrawler*

**Abstract**

*Biomedical ontologies helps discover hidden semantic links between heterogeneous and multi-scale biomedical datasets. Computational methods to ontology analysis may provide a semantic flavor to data analysis of biomedical mathematical models and help discover hidden links. In this paper we present Grontocrawler - a framework for visual ontology exploration applied to the biomedical domain. We define an OWL sublanguage - $\mathcal{L}$ and we present a methodology for transformation of $\mathcal{L}$ ontologies into directed labelled graphs. We then show how Social Network Analysis techniques (e.g., centrality measures, graph partitioning, community detection) can be used to i) filter the information presented to the user, and ii) provide a summary of knowledge encoded in the ontology. Finally, we show the application of ontology exploration in the biomedical domain to help discover hidden links between the biomedical datasets.*

Categories and Subject Descriptors (according to ACM CCS): H.3.3 [Information Search and Retrieval]: Information filtering—H.5.2 [Information Interfaces and Presentation]: User interfaces—Graphical user interfaces (GUI) I.2.4 [Computing Methodologies]: Artificial Intelligence—Knowledge Representation Formalisms and Methods J.3 [Computer Applications]: Life and Medical Sciences—Medical information systems

© The Eurographics Association 2015.

DOI: 10.2312/stag.20151293

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

## 1. Introduction

Physiological processes inside a human body may be represented with mathematical models. Computational methods may then be used in order to get further insight about these processes. Consider the two datasets on Figure 2. On the left a CT acquisition which produces discrete medical images that represent a human brain. From these images we can build a 3D model of a human crane. We start with the isolation of pixels corresponding to the boundary of a human crane in each image by thresholding the grayscale values. We then apply a surface reconstruction algorithm to obtain a 3D representation. On the right of the Figure 1 we have another type of acquisition, describing digitally the human motion. Motion capture markers as well EMG markers are placed on human body and record the spatial displacement of anatomical landmarks and muscle activity throughout one gait (motion) cycle.

The two acquisition scenarios produce multi-scale biomedical data, which are highly heterogeneous. They have different spatial domains and they represent information coming from different biological scales (organ, behavior). Though, these data may be represented as a vector space and similarity measures may be applied to them, the interpretation of the similarity may not be trivial. Without a medical background knowledge, relating the two datasets, it is not clear how the two scenarios are related. When *distance* measures are not enough to assess the semantics of similarities, we need to have background knowledge (e.g., same patient has underwent two different acquisition sessions, human brain activity might have influenced the gait pattern of the patient).

One way to formalize this knowledge is to use ontologies [Gru93], which conceptualize the domain of application by representing it as a set of concepts and relations among them. In fact, in the biomedical domain it is a common practice to formalize the medical background knowledge in biomedical ontologies to increase the interoperability between the medical applications [SAR*07]. Ontologies provide a semantic layer which facilitates data management and browsing. Concepts of ontologies drive query formulation over the content stored in repositories or knowledge bases and help in indexation of data and information [Mä05]. Biomedical knowledge bases and repositories, that use ontologies as their semantic backbones, can be explored by navigating interactively and visually ontologies that they rely on. Computational methods to ontology analysis may enhance data analysis of biomedical mathematical models stored in the knowledge bases, by reasoning on the semantic links.

In this paper the three aspects of ontology analysis are considered: i) Ontology segmentation or module extraction from ontologies [PJC09], ii) Ontology visualization or Ontology exploration [KHL*07], and iii) Structural Semantic Analysis of ontologies [HHJ*06] via SNA (Social Network Analysis) [CSW05]. We present Grontocrawler, a framework to combine the three facets of ontology analysis applied in the biomedical domain. Ontocrawler relies on graph representation of OWL (Web Ontology Language) [BvHH*04] ontologies and uses graph analysis algorithms to address these aspects. The contributions of this paper to the state of the art are as follows: i) while most of the OWL ontology to graph transformations are based on *intuitive notions* [NM00, SR06, HHJ*06, MMP*11], we propose a method based on the OWL's theoretical foundation - Description Logic [BCM*03], ii) apart for some exceptions [SK04, MMP*11], the three aspects of ontology analysis were studied separately, whereas we demonstrate connections between them and how they can be used together for ontology exploration, iii) and we show the application of ontology exploration in the biomedical domain to help discover hidden links between the biomedical datasets.

Grontocrawler demonstrates novel possible connections between the SNA methods, graph visualization and formal methods to knowledge modelling, in the vein of previous works [HHJ*06, MMP*11, Mik11, MRW14] which motivate hybrid approaches to ontology analysis.

## 2. Related work

Current ontology engineering tools (e.g. Protégé [NM01]) provide various functionalities for ontology analysis and interactive visual exploration. Ontology analysis techniques can be divided into two categories, depending on which theoretical model is chosen. The first one relies on formal representation of ontologies in a knowledge representation language (e.g., Description Logic [BCM*03]); it includes services such as: consistency checking and ontology classification [Abb12]. The second category treats an ontology as a (labeled, directed) graph and relies on graph analysis techniques for ontology analysis (ontology segmentation [PJC09], ontology visualization [KHL*07]).

In literature ontology segmentation is known under different names: subontology extraction, ontology modularization, ontology decomposition. First algorithms to *extract* a module of an ontology, satisfying certain user requirements, were proposed in PROMPT tool [NM00]. A similar approach based on graph traversal was outlined under the name of Web Ontology Segmentation in [SR06]. Both represent graph-based approaches to ontology modularization and provide only *intuitive notions* of what an ontology module is. Pathak et al [PJC09] provide a good overview of the application of ontology modularization in the biomedical domain. Since Pathak's survey ontology modularizaiton domain has known many results. First of all the formal underpinning of what an *ontology module* is was defined in [GHKS08]. Based on this formal notion Del Vescovo et al, have defined *Atomic decomposition of ontologies* [DVPSS11] by studying *chains* of modules extracted from one ontology, thus defining its atomic structure. Later, the same group has stud-

© The Eurographics Association 2015.

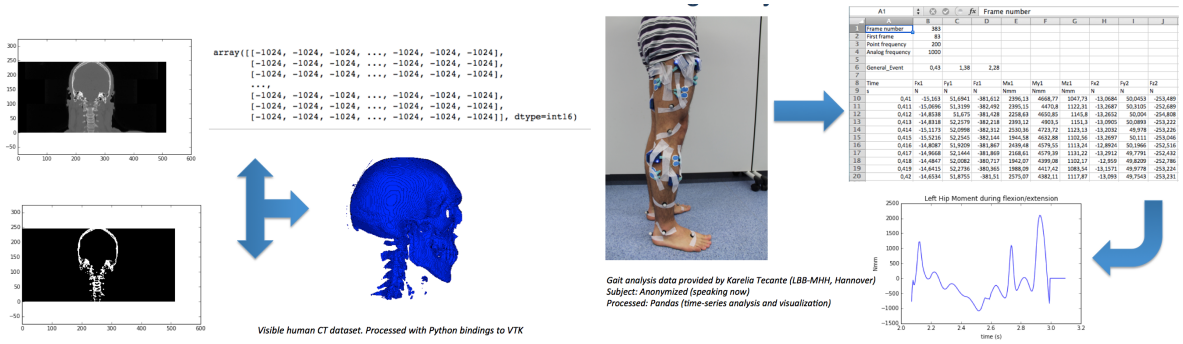


**Figure 2:** *Heterogeneity of acquired and processed biomedical data*

ied the level of decomposability of open biomedical ontologies available at BioPortal [DVGK*11]. More recently, a linear algorithm for computing atomic modular structure of an OWL ontology by using hypergraph representation of axiom dependance has been presented in [MRW14].

Visual exploration of ontologies is usually obtained through the adaptation of InfoVis techniques (Hierarchical or Network Visualization techniques) on graph (network) representation of ontologies [KHL*07]. Some techniques are better at detailing topological information of an ontology; some are better at specific tasks of showing a particular class instance with a certain constraint [MMP*11]. Most popular visualization algorithms rely on network layout calculation and belong to a class known as force-directed algorithms [Tam07].

Graphs have been also used to tackle other (i.e., other than *extraction* of modules and/or visualization of ontologies) problems of ontology analysis. Lembo et al. [LSS13] propose ontology classification algorithm, which transforms OWL QL ontologies into directed graphs, and computes subsumption relations via transitive closure computation. Social Network Analysis techniques application to ontology analysis has been pioneered by Hoser et al. [HHJ*06], where standard in SNA community graph metrics based on: node degree, node betweeness and on eigenanalysis of the adjacency matrix, were used to study properties of ontologies. The connection between SNA and Ontology Analysis have also been studied in a highly cited paper by Mika [Mik11], bridging Social Networks and Semantics. Network partitionning algorithms have been used in [SK04] to identify islands of ontology, a notion comparable to a module of ontology (as used by the graph-based modular extraction community), with the applications to Visual Analytics.

In most cases, whenever graph representation of ontologies is used, the process of identification of edges takes into account mostly RDFS (Resource Description Framework Schema) [BG14] axioms, targeting direct hierarchical relations of concepts (taxonomies). For example, the most frequent OWL ontology to graph transformation treats *named OWL concepts and/or OWL individuals and/or OWL Object/Datatype properties* as nodes and *TBox* (`rdfs:subClassOf`, `rdfs:domain` and `rdfs:range`) Abox (`rdf:type`) and RBox (`rdfs:subPropertyOf`) axioms as edges. Such a transformation is sufficient for lightly axiomatized linked-data collections, relying on ontologies having mostly taxonomical structure, but does not cover the whole spectrum of biomedical knowledge encoded in biomedical ontologies (see Section 3.1 for more details).

For biomedical ontologies model parthood and functional relations of anatomical entities by using, for instance, *existential restriction* on properties [Boe12]. These ontologies require a more expressive language than RDFS to capture biomedical relations. OWL language [HPSvH03] provides a rich set of constructors to model complex relationships between the concepts and for that reason is a *de-facto* standard for modelling complex biomedical knowledge. Consequently, the need for computational analysis of OWL ontology axioms to support interactive ontology exploration and segmentation has arisen.

Graph analysis algorithms have been used separately for ontology visualization and ontology segmentation in literature, apart for some exceptions [MMP*11, SK04], where the focus was mostly on visualization and the ontology axiom processing was based on *intuitive notions* rather than on Description Logic [BCM*03]. The two problems (visualizaiton and segmentation), can however be linked in a common framework where knowledge is represented as directed and labelled graphs, with the identification of nodes and edges guided by the theoretical model to knowledge modeling (DL). Social Network Analysis techniques, can then provide more intuitive ways of querying linked data backed up by highly axiomatized OWL ontologies to support Visual Analytics, Decision making tools and general Intelligent Systems focused on hypothesis testing in various biomedical domains as in [GBM*08, AVF*14].

## 3. Notation

Before we proceed with the presentation of Grontocrawler, we would like to introduce the: i) OWL sublanguage $\mathcal{L}$

© The Eurographics Association 2015.

which we consider in our work and the notation to represent its axioms, ii) notation for directed labeled graphs, and iii) connection of $\mathcal{L}$ to the biomedical ontologies. We also discuss the assumptions we make in our methodology about the structure and the content of OWL ontologies.

**Considered sublanguage for OWL ontologies** We present the subset of OWL 2 [GHM*08] language - $\mathcal{L}$ considered in this work during the OWL ontology to graph transformation, roughly it corresponds to OWL2-EL [BBL05] with a restriction of having atomic concepts only in the left hand side of the *concept inclusion* $A \sqsubseteq C$. We use the German notation for describing its constructs and axioms, similarly to the one found for OWL2 QL profile in [LSS13].

$$B \rightarrow A|\delta(U) \qquad C \rightarrow B|\exists P.A|\exists P.D$$
$$D \rightarrow B \sqcup B|B \sqcap B \qquad R \rightarrow P|U.$$

where: $A, P, U$ are symbols denoting respectively an *atomic concept*, an *atomic role*, and an *atomic attribute*. $B$ - set of basic concepts, $B(a)$ denotes that $a$ is an individual of $B$. $C$ set of concepts formed by using a qualified restriction on atomic concepts or concepts from set $D$. $D$ - set of concepts constructed using *conjunction* or *disjunction* of basic concepts, $\delta(U)$ - the domain of $U$, i.e., the set of objects that $U$ relates to values. $R$ - set of properties.

**Notation for Graphs** In this work we use directed graphs $G = (V, E)$, nodes and edges are labelled and may have attributes attached to them. For convenience $L_V, L_E$ denote labels and $A_V, A_E$ denote attributes for nodes and edges, respectively. We use $\texttt{predecessors}(n, G)$ to denote the set of nodes $p_n \in V$ such that there exists in $E$ an edge $(p_n, n)$. Similarly we use $\texttt{successors}(n, G) = \{s_n \,|\, \exists e = (n, s_n), e \in E\}$. For ease of the notation every edge is represented as a tuple (source, target, label, attributes).

**Adjacency matrix for graphs.** We use adjacency matrix $A$, with 1 row and 1 column for each node, defined as follows:

$$A = \begin{cases} a_{ij} & := 1, (v_i, v_j) \in E \\ a_{ij} & := 0, otherwise \end{cases}.$$

Adjacency matrices are used in the computation of *centrality* measures as well as in the construction of the graph Laplacian matrix. Please note that for the graph Laplacian we transform $G$ into an undirected graph, i.e. $\forall e_{ij}, a_{ij} = a_{ji} = 1$. This is a strong assumption and we discuss the consequences of it in the discussion section.

### 3.1. $\mathcal{L}$ and biomedical ontologies

Whereas RDFS ontologies mainly define taxonomical relationships between the concepts as well as domain and range restrictions for properties, Web Ontology Language provides several enhancements to represent more complex relationships between the concepts. In the biomedical domain, the existential axioms are used to model parthood, spatial, causal and functional relationships.

As an example, consider the following axioms, which define what *Femur* terminologically is (expressed in Description Logic (DL) [BCM*03]). The *Femur subontology* (set of DL axioms) is an excerpt of the FMA (Foundational Model of Anatomy) ontology [RM03] and visualized in Protégé ontology editor on Figure 3 (bottom right part of the GUI).

$$Femur := \{$$
$$\overbrace{Femur}^{A} \sqsubseteq \overbrace{Long\,bone}^{A} \tag{1a}$$
$$\underbrace{Femur}_{A} \sqsubseteq \underbrace{\exists constitutional\,part\,of.Thigh}_{\exists P.A} \tag{1b}$$
$$\}$$

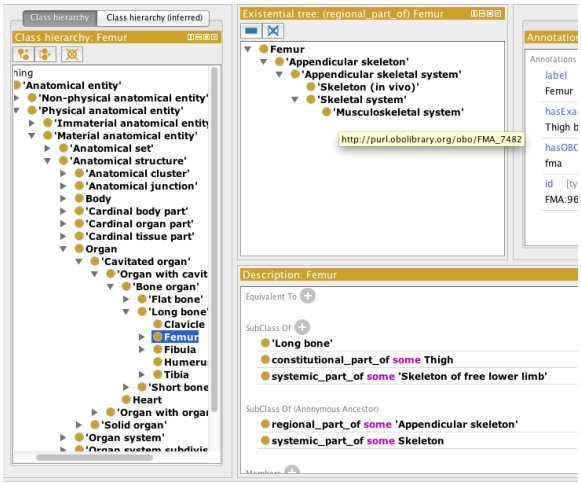


**Figure 3:** *OWL axioms (beyond RDFS) as employed in FMA*

Note that concept inclusion 1a models the subsumption relation $A \sqsubseteq A$ between two atomic concepts $\{Femur, Long\,bone\} \in A$. It is the usual direct taxonomical or hierarchical relationship between the two concepts. Such axioms constitute the main taxonomical skeleton of an ontology and is visualized as a *rooted tree* or a *hierarchy* in most of the ontology editors (cf. left part of Figure 3). Usually graph-based approaches to Ontology Analysis consider only such kind of semantic relations between the concepts (i.e., DL concept inclusion axiom between two atomic concepts).

DL concept inclusion axiom 1b of type $A \sqsubseteq C$, where $A = Femur$ (atomic concept) and $C = \exists P.A$ is formed with a existentially qualified restriction on property $constitutional\,part \in P$, is an example of a semantic relation between two concepts $\{Femur, Thing\} \in A$, as we call *beyond RDFS*. Note that FMA definition of *Femur* uses several

© The Eurographics Association 2015.

*concept inclusions* of type $A \sqsubseteq C$. These semantic relations between the concepts are too important to be neglected, yet state-of-art graph-based approaches to Ontology Analysis do not seem to be taking them into account.

**Assumptions for OWL ontologies** $\mathcal{L}$ was designed to support the DL axioms common to the biomedical ontologies. We assume that ontologies are expressed in OWL and have many axioms which can be captured by $\mathcal{L}$. The ontologies which contain axioms not taken into account by the language constructs of $\mathcal{L}$ are accepted, but only the supported axioms will be used in OWL ontology to directed labelled graph transformation.

## 4. Methodology and algorithms

Grontocrawler may be used as an ontology segmentation tool as well as a visual ontology exploration tool. In both cases it relies on OWL ontology to graph transformation, which we refer to as $\mathcal{L} \mapsto G$ (i.e., the procedure to transform ontology into graph). Nodes in $G$ represent concepts ($B$) or individuals ($B(a)$), attributes are used to keep track of the specific type. Edges represent a semantic relation between concepts (e.g., direct taxonomical, existential taxonomical as in concept inclusions 1a, 1b) (see 4.1 for implementation details). As in the case of nodes, edge attributes help us identify the specific type of semantic relation.

### 4.1. $\mathcal{L}$ axioms and RDF graphs

We provide one detailed example of one edge production corresponding to a $A \sqsubseteq \exists.P.A$, $\mathcal{L}$ axiom 2.

Consider the following $\mathcal{L}$ axiom:

$$\underbrace{\texttt{Cartilage\_thinning}}_{\texttt{rdfs:subClassOf}} \sqsubseteq \overbrace{\exists causes.\texttt{Joint\_stiffness}}^{\text{blank node construction}} \tag{2}$$

It is encoded as a set of RDF triples forming an RDF graph (see official W3C specification for RDF graph patterns for DL axioms [BvHH*04]) as depicted on the following Listing 1. Notice how the *right hand side* of the concept inclusion is realized through a *blank node* construction.

**Listing 1** RDF graph encoding of a DL existential axiom

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix msh: <http://example.org/msh/1.0/> .

msh:Cartilage_thinning rdf:type    owl:Class .
msh:Joint_stiffness    rdf:type    owl:Class .
msh:causes             rdf:type    owl:ObjectProperty .
msh:Cartilage_thinning
    rdfs:subClassOf [ # blank node construction
        rdf:type owl:Restriction;
        owl:onProperty msh:causes;
        owl:someValuesFrom msh:Joint_stiffness
    ] .
```

We analyse this RDF graph and infer a relationship between `Cartilage thinning` ($v_i$) and `Joint stiffness` ($v_j$) as depicted on Figure 4. Finally, we produce edges $(v_i, v_j, \text{"causes"}, \text{"existential superclass"})$ and $(v_j, v_i, \text{"causes"}, \text{"existential child"})$.

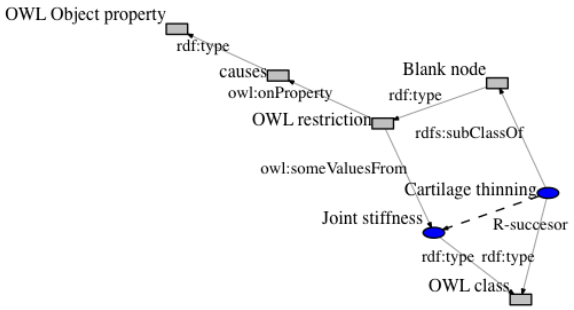


**Figure 4:** *RDF graph corresponding to Listing 1*

Other $\mathcal{L}$ axioms are treated in the similar manner during the edge production procedure. The Table 1 summarizes the edge production rules that we support in Grontocrawler.

As a result $\mathcal{L} \mapsto G$ allows us represent an ontology as a network of interrelated concepts, where relations are semantically consistent up to the $\mathcal{L}$ language.

### 4.2. Graph traversal and $\mathcal{L}$ segmentation/exploration

Ontologies with over ten thousand classes suffer severely from scaling problem [SR06]. Segmentation of ontologies by choosing application-specific subparts of an ontology (or modules) is a way of overcoming these difficulties. As an example consider a biomedical application which requires a formalization of *knee* anatomy. Developers may choose to re-use a comprehensive ontology, such as FMA [RM03], for that purpose. However, the original FMA ontology covers the whole human body anatomy and is too complex and broad. The developers may extract a relevant subpart of FMA ontology focusing on the human knee. They provide the seed nodes, for instance the bones participating in the knee joint articulation Figure 4.2. By processing the axioms, with which this information is encoded, we can extract the relevant subpart of what was deemed to be a subpart of formalization of information on the knee joint.

In Grontocrawler the User starts the segmentation/exploration by providing a *focus entity* and as a result Grontocrawler presents the infered *semantic context* around the *focus entity*. Ontology segmentation/exploration is thus performed through a graph traversal of $G$ starting from the *seed node*. Specifically, we adopt the iterative breadth-first search algorithm for graph traversal in Grontocrawler (cf. Algorithm 1).

© The Eurographics Association 2015.

| CI Rule | Pattern $\alpha := seed \sqsubseteq \beta$ | Production |
|---|---|---|
| $R_1$ | $\beta := A_1$ | $E \leftarrow E \cup (seed, A_1, \sqsubseteq, \texttt{"superclass"})$ |
| $R_2$ | $\beta := \exists P.A_1$ | $E \leftarrow E \cup (seed, A_1, R, \texttt{"existential superclass"})$ |
| $R_3$ | $\beta := \exists P.C, \forall A_i \in C$ | $E \leftarrow E \cup (seed, A_i, P, \texttt{"existential superclass"})$ |
| ABox Rule | Pattern | Production |
| $R_4$ | $\alpha := seed(a)$ | $E \leftarrow E \cup (a, seed, \texttt{"is a"}, \texttt{"instance of"})$ |
| $R_5$ | $\alpha := P(seed, b)$ | $E \leftarrow E \cup (seed, b, P, \texttt{"R-successor"})$ |
| DBox Rule | Pattern | Production |
| $R_6$ | $\alpha := D(a, Literal(seed))$ | $E \leftarrow E \cup (a, \phi(seed), D, \texttt{"D-successor"})$ |

**Table 1:** *Production rules $\mathcal{L} \mapsto G$ for Algorithm 2 (produce_edges)*

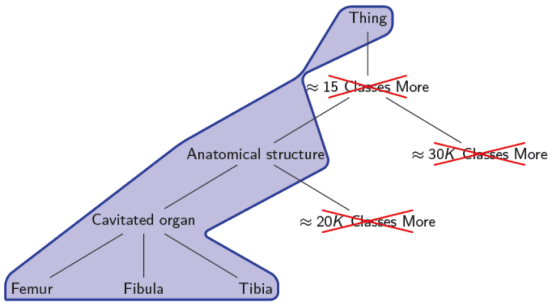


**Figure 5:** *FMA module focused on Femur*

**Algorithm 1** Graph traversal algorithm (Iterative Breadth-first search)

**Require:** $seed, visited, start\_queue, crawl\_options$
  $G \leftarrow \emptyset$
  $to\_crawl \leftarrow start\_queue \bigcup seed$
  **while** $to\_crawl \neq \emptyset$ **do**
    $u \leftarrow pop(to\_crawl)$
    **if** $u \notin visited$ **then**
      $visited \leftarrow visited \bigcup u$
      $successors \leftarrow get\_successors(u, crawl\_options)$
      $to\_crawl \leftarrow to\_crawl \bigcup successors$
      $G \leftarrow connect(G, u, successors)$
    **end if**
  **end while**

**Algorithm 2** *get_successors* for Algorithm 1

**Require:** $seed, crawl\_options$
  $E \leftarrow \emptyset$
  **for all** $R_i \in crawl\_options$ **do**
    $E \leftarrow E \cup produce\_edges(seed, R_i)$
  **end for**

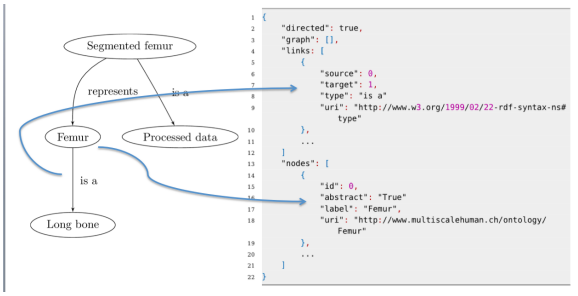


**Figure 6:** *Structured graph representation as JSON object*

#### 4.2.1. Visualization of $G$ representation on a computer

Rule-based transformation of ontologies into graphs detailed previously, produces smaller subgraphs which are merged into one final labelled graph with every edge having an attribute describing the type of an edge (name of the rule). In Grontocrawler, we use JSON to represent $G$ (cf. Figure 6) for exchange of information over the Web as well as the input to the graph visualization frameworks.

### 4.3. Social Network Analysis on $G$

The resultant (labelled, directed) graph $G$ can be analysed by using SNA techniques (centrality measures, graph partitioning and community detection). We employ the centrality measures to identify the most important nodes (concepts) according to their *strategic* position in the network. The measure of importance is then used to limit the number of information presented to the user by thresholding only *important* nodes. Graph partitioning and community detection algorithms are used to identify clusters of ontology according to a specific *topicality*.

**Centrality metrics on Graphs.** We consider primarily two *centrality measures* for nodes in a graph; based on the degree of a node (*degree centrality*); calculated from $A$ as a row or column sum $c_k = \sum_l^n a_{kl}$, and (*betweenness centrality*): based on the proportion ($g(v)$) of all the shortest paths from node $s$ to node $t$ (any two nodes in a graph) that pass through a node $v$ (denoted $\sigma_{st}(v)$) to the total number of number of shortest paths ($\sigma_{st}$). We denote it as $g(v) = \sum_{i \neq v \neq j} \frac{\sigma_{st}(v)}{\sigma_{st}}$.

**Graph partitioning and community detection** The Fiedler vector corresponding to the second smallest eigenfunction of the graph Laplace operator is used to partition the graph [Chu97] representing the ontology. We use Louvain's [BGLL08] heuristic metric for modularity computation and community detection. This algorithm tries

© The Eurographics Association 2015.

to keep highly densed subgraphs separate from others, thus it produces certain clusters which could be separated even further and contain nodes of several topicalities.

## 5. Grontocrawler: design and specifications

Both Ontology visualization and Ontology segmentation require computational means of axiom processing to adequately capture the variability of the domain knowledge encoded. We argue that, the two can work in couple. In Grontocrawler the visualization of ontology guides the ontology segmentation process, by providing the user an overview (a summary) of an ontology. The user can then identify the relevant seed nodes, which are fed to the segmentation algorithm, yielding the induced subpart. Analagously, ontology analysis and identification of *key-concepts* [MMP*11] guides the visualization of ontologies by filtering which and how much of information should be presented to the user.

Grontocrawler is implemented as a Web application and consists of two logical components: i) ontology processing (owl to graph transformation, performed server-side), ii) interactive visual ontology exploration (performed client-side). The test version of the tool is available at this address `http://45.33.71.144/grontocrawler/`.

**Interface of the system** The interface of Grontocrawler follows a simplistic approach of an information portal similar to a welcome page of a web-search engine (e.g. Google). Inputed text - a string $a$ - is fed to $\phi$, which matches the possible concepts. We use Levenshtein distance [Lev66] (metric on strings) to decide to which concept $s$ it maps to (i.e., we perform $\arg\max_a \phi(s,a)$). $s$ is then used as a *seed* node in the graph traversal algorithm.

The user controls the level of inference (identification of *neighbors* in the graph traversal algorithm) through a checklist, in which every option is mapped to one of the RDF graph patterns ($\mathcal{L} \mapsto G$ transformation rule (cf. Table 1). The list of transformation rules are fed to the server via AJAX call for interactive response. Result of the graph traversal algorithm, a subgraph, is then presented to the user, with the size of nodes reflecting its importance measured by the centrality measures of the network (cf. Figure 1).

**OWL ontology transformation. Server-side** OWL ontology transformation into a directed graph is done on the server-side, the exact rules of transformation are summarized in Table 1. We use *RDFLib* - Python RDF processing library to perform RDF graph pattern matching and other RDF processing manipulations. We use available *RDFLib* plugins for ontology persistance mechanisms, in particular we are using *RDFAlchemy* to connect to persisted triple store in a MySQL relational database, as well as the *SPARQLUpdate* pluging to connect to arbitrary triple stores, supporting SPARQL endpoints (e.g., Stardog). The server is implemented via Python *Flask* library for wSGI server applications. Graph management and algorithms are provided by the *NetworkX* Python library for Network Analysis, we use Louvain's community detection implementation provided by the authors (available on Bitbucket).

**Interactive exploration. Client-side** The results of the query are presented to the user as a network of connected concepts. The HTTP queries to extract the semantic content of a focus entity is implemented through an AJAX calls, providing interactive means to KB exploration. The nodes are drawn interactively on the canvas of the Web browser as SVG elements, and laid out by using the force-directed layout for graph drawing [FR91]. We use the javascript D3 [BOH11] library to render 2D visualization of the graph as well as for the layout computation. Interactivity comes from the fact that it is possible to redraw and recompute the layout of the graph every time the user performs a query. Moreover, the user can drag the nodes to spread them apart for better exploration experience, the layout is recomputed everytime the nodes' configuration (position) is altered.

## 6. Initial experiments

Social Network Analysis on a network produced as a result of OWL ontology to directed labelled graph transformation gave us some insights about the structure and the semantics of the knowledge encoded in the ontology. We applied our methods on the MSH (MultiScaleHuman) ontology [FP715], which focuses on the description of multi-scale biomedical data. It connects the medical background knowledge on clinical practices (patients, acquisition sessions) to the anatomy (knee joint formalization, derived from FMA). In addition it formalizes the causal chain of cartilage degradation during Osteoarthritis. In that ontology the authors also studied User interests in data and knowledge, modeled as affinity measures of specialists (radiologists, orthopedists) to concepts in the ontology such as: specific anatomical entities or biological scale of biomedical data.

### 6.1. $\mathcal{L} \mapsto G$ (MSH ontology)

We considered two graph transformations: i) subgraph extraction where *focus* is *Femur*, all inference rules were set and ii) full graph of the MSH ontology. We denote them $G_1, G_2$ respectively and present some statistics on these networks on Table 6.1.

| Graph | # nodes | # edges | # partitions | density | # components |
|---|---|---|---|---|---|
| $G_1$ | 170 | 472 | 12 | 0.01642 | 1 |
| $G_2$ | 213 | 538 | 21 | 0.01191 | 11 |

**Table 2:** *Network statistics for two graphs*

### 6.2. Inference and hidden link discovery

We would like to discuss the purpose of inference and how it can help discover hidden links. In Grontocrawler the final representation of the ontology is represented as a directed

© The Eurographics Association 2015.

label graph, where edges are typed (their type is stored as edge attribute) and represent a structural semantic similarity link between the two concepts (represented as two nodes). Thus, a path in the graph $G$ between the two nodes $u, v$ exhibits presence of semantic similarity and its (weighted) length represents the strength of that similarity.

We provide an example where computing paths between the nodes is used to infer new relationships between the datasets in the presence of partially asserted facts. Explanation of semantic path or a path of relationships from one dataset $d_i$ to another ($d_j$) is approximated by the computation of *shortest paths* from the nodes representing $d_i$ to $d_j$. The procedure is as follows, the User enters the names of two datasets and the system presents the possible connections between the two:

```
[['Segmented femur', 'Femur', 'MRI 1'],
 ['Segmented femur', 'Radiologist', 'MRI 1'],
 ['Segmented femur', 'Gait analysis', 'MRI 1'],
 ['Segmented femur', 'Soft tissue loading', 'MRI 1'],
 ['Segmented femur', 'Medium scale', 'MRI 1']]
```

The user is thus given an overview of possible links between the two datasets, i.e. that both are related to *Femur* bone, might interest *Radiologist*, could be important in the study of *Gait analysis* and/or *Soft tissue loading* and come from *Medium scale* (spatial representation in visualization systems). These semantic paths are then used as input to the algorithm compute the induced *subgraph*, yielding as a result the semantic context in which the two occur (cf. Figure 7). Notice that the subgraph contains more information and can be used for further hidden-link discovery process.

Of course the same subgraph could have been extracted by running *ad-hoc* SPARQL [PAG09] queries to the KB, however in that case the user is not only required to know the syntax of the querying language, but also the exact *structure* of the RDF graph.

### 6.3. Community detection and modularization

Even though our networks are relatively small $\approx 200$ nodes and $\approx 500$ edges, it is not possible to present all the information at once to the user. We study the *modularity* properties of our network by applying the Louvain clustering heuristics, suitable for dense networks [BGLL08]. Some clusters were quiet surprisingly good and reflected the correct *topicality* of concepts involved and some were understandably poor. For instance, one of the clusters correctly identified entities around cartilage and histological data (cf. Figure 8), which could interest a *Molecular biologist*. The other one contained only information on *Meniscus* (cf. Figure 9), though the two belonged to the same hierarchy in the *Anatomical entity* tree, the community detection algorithm was able to separate the two, due to the presence of information of User interests in certain *anatomical entities* and our axiom processing algorithm.

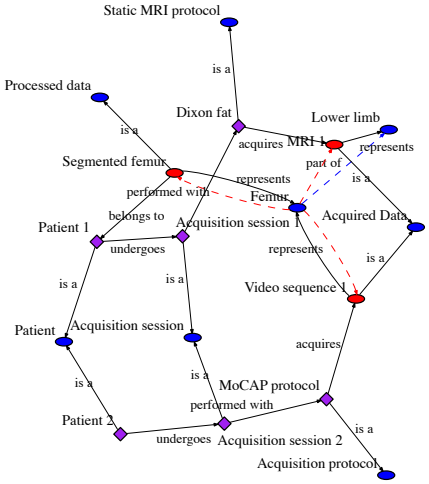


**Figure 7:** *Example of inference in G through path reachability computation, which helps discover hidden links between the datasets*

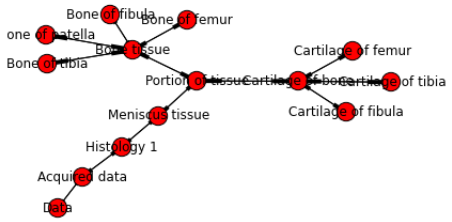


**Figure 8:** *Cluster (module) related to Cartilage*

## 7. Discussion

The OWL sublanguage which we consider in our work does not capture the whole spectrum of expressivity that OWL language can offer. Axioms expressed (ontologies) in this language do however lend themselves easier to transformation into directed labelled graphs. $\mathcal{L}$ focuses on biomedical ontologies where the considered restricted language constructors are sufficient to model (axiomatize) some of the most important biomedical relationships. Grontocrawler positions itself as a both research contribution and a technological contribution. From research viewpoint it tries to study the connection between formal methods to knowledge representation and Social Network Analysis. From a technological point of view, Grontocrawler can be considered as a means to ontology analysis with a contribution to the testing phases of ontology creation or to ontology exploration. Since it supports Web services it opens up new opportunities

© The Eurographics Association 2015.

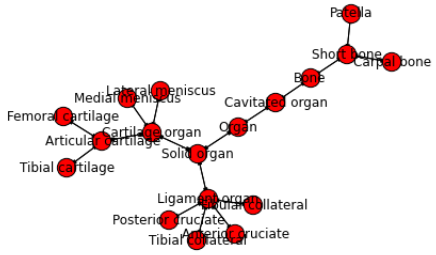


**Figure 9:** *Cluster (module) related to Meniscus*

of collaborative knowledge modelling and hypothesis testing in the biomedical domains.

**Future work and limitations** The initial experiments and the results obtained were tested on one ontology (MSH ontology), which suited the best the axioms of the $\mathcal{L}$ language introduced in the paper. More elaborate comparison/analysis of our approach on other ontologies is needed to support the research point that Grontocrawler aims to make. Well designed user studies are necessary to assert the usefulnes of the proposed methodology. The two points will be the top priority of our future work. The drawback of our visualization is that, though the edges produced by the graph traversal algorithm are typed or labelled, they are visualized indistinguishably by the same polygon line. Better glyph visualization techniques, taking into account the topological information that SNA analysis can bring, should be envisioned in the future.

**Eigenanalysis of networks** Unfortunately, eigenanalysis of the Laplacian matrices did not give much insight into the structure and the semantics encoded in the resultant graphs. This maybe partially explained by the fact that spectral analysis of graphs provides *global* properties of the network and perhaps is not the best suite for community detection. Moreover while doing spectral analysis we transformed our networks into undirected graphs, this is known to produce suboptimal partitions when the relationships are not *symmetric*. Besides, we also assumed that our networks were 1-mode networks, i.e. having the same type of edges, which is semantically not true. However, we have not found any spectral graph theory results for n-mode networks in the literature, and usually the reserachers relax this constraint by turning every network into a 1-mode network. Additional pointers may be found in [HGS05].

## 8. Conclusion

In this paper we presented Grontocrawler - a tool for interactive exploration and segmentation of biomedical ontologies. We defined a sublanguage $\mathcal{L}$ of OWL which we consider in our work. We presented a methodology of transformation of an ontology expressed $\mathcal{L}$ into a directed, labelled graph. We then showed how Social Network Analysis techniques (e.g., centrality measures, graph partitioning, community detection) can be used to i) filter the information presented to the user, and ii) provide a summary of knowledge encoded in the ontology. Finally, we showed the application of ontology exploration in the biomedical domain to help discover hidden links between the biomedical datasets.

**Acknowledgements** This work was supported from the EU Marie Curie ITN MultiScaleHuman (FP7-PEOPLE-2011-ITN, Grant agreement no.: 289897). The authors would like to thank all the partners for providing biomedical data sets.

## References

[Abb12] ABBURU S.: A Survey on Ontology Reasoners and Comparison. *International Journal of Computer Applications 57*, 17 (Nov. 2012), 33–39. Full text available. 2

[AVF*14] AGIBETOV A., VAQUERO R. M. M., FRIESE K.-I., PATANÈ G., SPAGNUOLO M., WOLTER F.-E.: Integrated Visualization and Analysis of a Multi-scale Biomedical Knowledge Space. In *EuroVis Workshop on Visual Analytics* (2014), The Eurographics Association, pp. 25–29. 3

[BBL05] BAADER F., BRAND S., LUTZ C.: Pushing the EL envelope. In *In Proc. of IJCAI 2005* (2005), Morgan-Kaufmann Publishers, pp. 364–369. 4

[BCM*03] BAADER F., CALVANESE D., MCGUINNESS D. L., NARDI D., PATEL-SCHNEIDER P. F. (Eds.): *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, New York, NY, USA, 2003. 2, 3, 4

[BG14] BRICKLEY D., GUHA R. V.: *RDF schema*. Feb. 2014. 3

[BGLL08] BLONDEL V. D., GUILLAUME J.-L., LAMBIOTTE R., LEFEBVRE E.: Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment 2008*, 10 (Oct. 2008), P10008. 6, 8

[Boe12] BOECKER M.: Teaching Good Biomedical Ontology Design. In *Proceedings ICBO* (2012). 3

[BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D3 Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Dec. 2011), 2301–2309. 7

[BvHH*04] BECHHOFER S., VAN HARMELEN F., HENDLER J., HORROCKS I., MCGUINNESS D. L., PATEL-SCHNEIDER P. F. \., STEIN L. A.: *OWL Web Ontology Language Reference*. World Wide Web Consortium, Feb. 2004. Published: W3C Recommendation Available at http://www.w3.org/TR/owl-ref/. 2, 5

[Chu97] CHUNG F. R.: *Spectral graph theory*, vol. 92. American Mathematical Soc., 1997. 6

[CSW05] CARRINGTON P. J., SCOTT J., WASSERMAN S.: *Models and Methods in Social Network Analysis*. Cambridge University Press, Feb. 2005. 2

[DVGK*11] DEL VESCOVO C., GESSLER D., KLINOV P., PARSIA B., SATTLER U., SCHNEIDER T., WINGET A.: Decomposition and Modular Structure of BioPortal Ontologies. In *The Semantic Web – ISWC 2011*, Aroyo L., Welty C., Alani H., Taylor J., Bernstein A., Kagal L., Noy N., Blomqvist E., (Eds.), vol. 7031 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011, pp. 130–145. 2

© The Eurographics Association 2015.

[DVPSS11] DEL VESCOVO C., PARSIA B., SATTLER U., SCHNEIDER T.: The modular structure of an ontology: Atomic decomposition. *IJCAI Proceedings-International Joint Conference on Artificial Intelligence 22*, 3 (2011), 2232. 2

[FP715] FP7 MULTISCALEHUMAN: *MSH Ontology: deliverable reports D8.2 (m24, m36) and OWL file*. 2015. Published: http://multiscalehuman.miralab.ch/repository/Public_download/D8.2_MSD-Ontology/. Accessed July 28, 2015. 7

[FR91] FRUCHTERMAN T. M. J., REINGOLD E. M.: Graph drawing by force-directed placement. *Software: Practice and Experience 21*, 11 (Nov. 1991), 1129–1164. 7

[GBM*08] GUPTA A., BUG W., MARENCO L., QIAN X., CONDIT C., RANGARAJAN A., MÜLLER H. M., MILLER P. L., SANDERS B., GRETHE J. S., ASTAKHOV V., SHEPHERD G., STERNBERG P. W., MARTONE M. E.: Federated access to heterogeneous information resources in the Neuroscience Information Framework (NIF). *Neuroinformatics 6*, 3 (Sept. 2008), 205–217. 3

[GHKS08] GRAU B. C., HORROCKS I., KAZAKOV Y., SATTLER U.: Modular Reuse of Ontologies: Theory and Practice. *JAIR 31* (2008), 273–318. 2

[GHM*08] GRAU B. C., HORROCKS I., MOTIK B., PARSIA B., PATEL-SCHNEIDER P., SATTLER U.: OWL 2: The Next Step for OWL. *Web Semant. 6*, 4 (Nov. 2008), 309–322. 4

[Gru93] GRUBER T. R.: A Translation Approach to Portable Ontology Specifications. *Knowl. Acquis. 5*, 2 (June 1993), 199–220. 2

[HGS05] HOSER B., GEYER-SCHULZ A.: Eigenspectral Analysis of Hermitian Adjacency Matrices for the Analysis of Group Substructures. *The Journal of Mathematical Sociology 29*, 4 (Oct. 2005), 265–294. 9

[HHJ*06] HOSER B., HOTHO A., JÄSCHKE R., SCHMITZ C., STUMME G.: Semantic Network Analysis of Ontologies. In *Proceedings of the 3rd European Conference on The Semantic Web: Research and Applications* (Berlin, Heidelberg, 2006), ESWC'06, Springer-Verlag, pp. 514–529. 2, 3

[HPSvH03] HORROCKS I., PATEL-SCHNEIDER P. F., VAN HARMELEN F.: From SHIQ and RDF to OWL: the making of a Web Ontology Language. *Web Semantics: Science, Services and Agents on the World Wide Web 1*, 1 (Dec. 2003), 7–26. 3

[KHL*07] KATIFORI A., HALATSIS C., LEPOURAS G., VASSILAKIS C., GIANNOPOULOU E.: Ontology Visualization Methods&Mdash;a Survey. *ACM Comput. Surv. 39*, 4 (Nov. 2007). 2, 3

[Lev66] LEVENSHTEIN V. I.: Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady 10*, 8 (Feb. 1966), 707–710. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965. 7

[LSS13] LEMBO D., SANTARELLI V., SAVO D. F.: Graph-Based Ontology Classification in OWL 2 QL. In *The Semantic Web: Semantics and Big Data*, Cimiano P., Corcho O., Presutti V., Hollink L., Rudolph S., (Eds.), no. 7882 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, Jan. 2013, pp. 320–334. 3, 4

[Mik11] MIKA P.: Ontologies Are Us: A unified model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web 5*, 1 (Aug. 2011). 2, 3

[MMP*11] MOTTA E., MULHOLLAND P., PERONI S., D'AQUIN M., GOMEZ-PEREZ J. M., MENDEZ V., ZABLITH F.: A Novel Approach to Visualizing and Navigating Ontologies. In *Proceedings of the 10th International Conference on The Semantic Web - Volume Part I* (Berlin, Heidelberg, 2011), ISWC'11, Springer-Verlag, pp. 470–486. 2, 3, 7

[MRW14] MARTÍN-RECUERDA F., WALTHER D.: Fast Modularisation and Atomic Decomposition of Ontologies Using Axiom Dependency Hypergraphs. In *The Semantic Web – ISWC 2014*, Mika P., Tudorache T., Bernstein A., Welty C., Knoblock C., Vrandečić D., Groth P., Noy N., Janowicz K., Goble C., (Eds.), no. 8797 in Lecture Notes in Computer Science. Springer International Publishing, Jan. 2014, pp. 49–64. 2, 3

[Mä05] MÄKELÄ E.: Survey of semantic search research. *Proceedings of the seminar on knowledge management on the semantic web* (2005). 2

[NM00] NOY N. F., MUSEN M. A.: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence* (2000), AAAI Press, pp. 450–455. 2

[NM01] NOY N. F., MCGUINNESS D. L.: *Ontology Development 101: A Guide to Creating Your First Ontology*. Tech. rep., 2001. 2

[PAG09] PÉREZ J., ARENAS M., GUTIERREZ C.: Semantics and Complexity of SPARQL. *ACM Trans. Database Syst. 34*, 3 (Sept. 2009), 16:1–16:45. 8

[PJC09] PATHAK J., JOHNSON T. M., CHUTE C. G.: Survey of modular ontology techniques and their applications in the biomedical domain. *Integrated computer-aided engineering 16*, 3 (Aug. 2009), 225–242. 2

[RM03] ROSSE C., MEJINO J. L. V.: A reference ontology for biomedical informatics: the foundational model of anatomy. *J. of Biomedical Informatics 36* (2003), 500. 4, 5

[SAR*07] SMITH B., ASHBURNER M., ROSSE C., BARD J., BUG W., CEUSTERS W., GOLDBERG L. J., EILBECK K., IRELAND A., MUNGALL C. J., LEONTIS N., ROCCA-SERRA P., RUTTENBERG A., SANSONE S.-A., SCHEUERMANN R. H., SHAH N., WHETZEL P. L., LEWIS S.: The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology 25*, 11 (Nov. 2007), 1251–1255. 2

[SK04] STUCKENSCHMIDT H., KLEIN M.: Structure-Based Partitioning of Large Concept Hierarchies. In *In: International Semantic Web Conference* (2004), pp. 289–303. 2, 3

[SR06] SEIDENBERG J., RECTOR A.: Web Ontology Segmentation: Analysis, Classification and Use. In *Proceedings of the 15th International Conference on World Wide Web* (New York, NY, USA, 2006), WWW '06, ACM, pp. 13–22. 2, 5

[Tam07] TAMASSIA R.: *Handbook of Graph Drawing and Visualization (Discrete Mathematics and Its Applications)*. Chapman & Hall/CRC, 2007. 3

© The Eurographics Association 2015.