

iCyrus: A Semantic Framework for Biomedical Image Discovery

Ahmad C. Bukhari¹, Mate Levente Nagy², Michael Krauthammer²,
Paolo Ciccarese³, *Christopher J. O. Baker¹

¹ Dept. CSAS, University of New Brunswick, Saint John, New Brunswick, E2L 4L5, Canada

² Yale Center for Medical Informatics, 300 Cedar Street, New Haven, CT 06510, USA

³ Harvard Medical School 25 Shattuck Street, Boston, MA 02115 USA

{sbukhari,bakerc}@unb.ca

Abstract. Images have an irrefutably central role in scientific discovery and discourse. However, the issues associated with knowledge management and utility operations unique to image data are only recently gaining recognition. In our previous work, we have developed Yale Image finder (YIF), which is a novel Biomedical image search engine that indexes around two million biomedical image data, along with associated metadata. While YIF is considered to be a veritable source of easily accessible biomedical images, there are still a number of usability and interoperability challenges that have yet to be addressed. To overcome these issues and to accelerate the adoption of the YIF for next generation biomedical applications, we have developed a publically accessible semantic API for biomedical images with multiple modalities. The core API called *iCyrus* is powered by a dedicated semantic architecture that exposes the YIF content as linked data, permitting integration with related information resources and consumption by linked data-aware data services. To facilitate the ad-hoc integration of image data with other online data resources, we also built semantic web services for *iCyrus*, such that it is compatible with the SADI semantic web service framework. The utility of the combined infrastructure is illustrated with a number of compelling use cases and further extended through the incorporation of Domeo, a well known tool for open annotation. Domeo facilitates enhanced search over the images using annotations provided through crowdsourcing. The *iCyrus* triplestore currently holds more than thirty-five million triples and can be accessed and operated through syntactic or semantic query interfaces. Core features of the *iCyrus* API, namely: data reusability, system interoperability, semantic image search, automatic update and dedicated semantic infrastructure make *iCyrus* a state of the art resource for image data discovery and retrieval.

Demo Page: <http://cbakerlab.unbsj.ca:8080/icyrus/>

Keywords: Linked biomedical images, Yale Image Finder, Interoperable web services

1. Background

Making biomedical image content explicit is essential with regards to making medical decisions such as diagnosis, treatment, follow-up, data management and the secondary use for biomedical research and assessment of care delivery. In the Life Sciences, spreadsheets, databases and XML files continue to be the conventional formats used to store experimental data, e.g. Biota [1], DrugBank [2] and Open Microscopy Environment (OME) [3]. Some projects go further and make their XML data accessible via knowledge-based grid services, e.g. high-throughput biological imaging data [4],

PLAZi [5]. However, the fact that data exists only in these legacy formats frequently impedes data integration and significantly impedes scientific discovery. More recently, small communities of researchers have adopted semantic technologies to support data integration and manipulation. In the *Semantic web* [6] paradigm, there exists a common framework comprising of standards and technologies that enables data to be shared and reused across applications. Specifically, semantic web uses an explicit data model to describe data in an unambiguous way such that independently generated data sets can be easily integrated under the same data model. The current proliferation of semantic web methodologies results from the use of cutting-edge standards such as the resource description framework (RDF) [7] and SPARQL [8] (a query language for RDF graphs), which are supported by several applications, due to their robustness and efficiency.

In recent years, Linked Data has emerged as the most adopted semantic web framework supporting data interoperability and reusability. Due to linked data benefits, several life science and census datasets have been transformed into linked data. Examples include LODD (linked open drug data) [9], Canadian health census to linked data [10], Bio2RDF [11], BioNLP-SADI [12] and the EMBL-EBI RDF services [13]. Moreover, several open-access non-semantic biomedical image repositories are available on the Internet such as the National Biomedical Imaging Archive (NBIA¹), NIH Images² and Yale Image Finder (YIF) [14] but none of these provide their content in a semantically accessible way. YIF is one of the most widely accessed biomedical image search engines, which retrieves biomedical images and associated papers based on queries made over the text within the image and its caption, and within the associated paper's abstract, title, and full text. YIF searches within the image using a sophisticated image segmentation method; followed by OCR with additional filtering of text for both high precision and high recall outputs [15]. Unlike the syntactic core of YIF, a recent work utilizes the semantic enrichment technique called SEBI to discover biological similar images [16]. The rest of the paper has been arranged as follows: Section "2" explains the system architecture and implementation procedure. Sections "2.4", "3" and "4" outline a series of case studies that leverage the API along with examples. Section "5", the concluding remarks section, highlights the importance of current work in comparison with available techniques and measures how our design and implementation extend the state of the art with respect to current Biomedical applications research for access to images.

2. System Architecture and Implementation

We have designed an API based on a semantic framework that exposes YIF content as linked data for integration with various other information resources and for consumption by linked data-aware data services. Figure 1 displays the iCyrus API Process Diagram. Specifically, YIF data is processed through the novel iCyrus algorithm

¹ National Biomedical Imaging Archive. <http://ncia.nci.nih.gov/>

² NIH Image Bank - National Institutes of Health. <http://imagebank.nih.gov/>

that transforms the data according to linked open data recommendations. The data is maintained in a triple store and is made accessible through a SPARQL endpoint and linked data explorer. To further integrate YIF with other data resources and frameworks, we developed Semantic Automatic Discovery and Integration (SADI) [17] web services through which the data can be accessed. This service layer also facilitates the coupling of iCyrus with Domeo, a well-known tool for open annotation, such that it provides crowd-sourced annotations as additional parameters to search algorithms that promote context-based searching. We have divided the system architecture into four phases based upon their functionalities. In the next sections, we explain the internal working of each phase.

2.1 Stage 1: Acquisitions and Manipulation Phase

In the first stage, prior to the LODification process, we acquired the image datasets from the Yale Image Finder repository to build a knowledgebase for our API [14]. Specifically, we accessed the Lucene index and transformed the data (including the metadata fields) into CSV files. After acquiring the data, it was prescreened to remove irrelevant information and to clean the CSV files. Using the Google Refine tool [18], data was migrated into a MySQL database for parallel storage alongside our triple store and for subsequent performance comparisons.

2.2 Stage 2: LODification and Storage Phase

The foremost task in semantic data publication is defining appropriate semantic vocabularies. Reusability is considered a noble practice in semantic web application development and it is generally accepted that well-known semantic vocabularies [19] should be reused to enhance data interoperability and to promote robust integration among datasets. To identify appropriate semantic mappings between available ontologies and the YIF metadata, we created a Java program that suggests possible mappings.

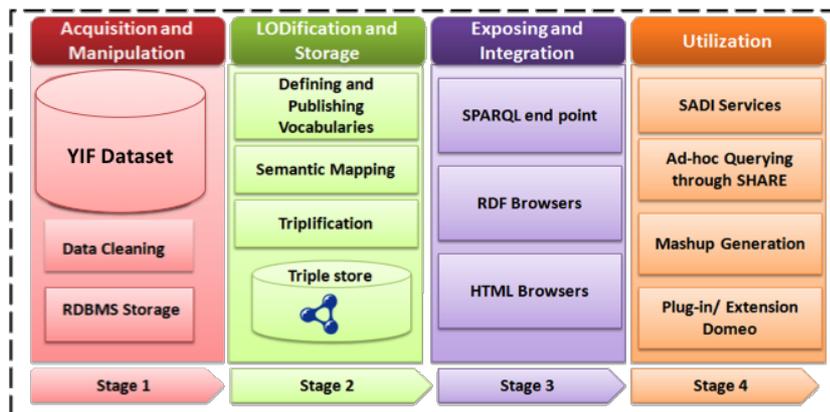


Fig.1 The iCyrus API Process Diagram.

The program extracts the tables and column names from our CSV files, storing them as variables and invokes a WordNet service [20] that lexically compares each variable with the ontology entities to find possible matches. The overall goal was to provide candidate matches for subsequent curation albeit a comprehensive benchmarking of the algorithm's performance was not made. A cursory evaluation of the derived mappings showed there were three types of results; (i) mappings that fully met our requirements, which suggested predicates such as *hasPubMedID* and *hasPMCID* in the FRBR-aligned Bibliographic Ontology (fbio); (ii) mappings that were insufficiently defined, like the *imageFeature* property that exists in DICOM Ontology³; and (iii) mappings with hosted resources that did not appear trustworthy. To manage the new vocabularies that fulfill the requirements of iCyrus and SEBI [16], we have developed the BIM (Biomedical Image) ontology [21] and set up an ontology-publishing server called UNBvps⁴ that helps to define new vocabularies and to publish the existing ontologies in an interactive way. The BIM ontology is separated in three parts; (i) image vocabularies - which hold vocabularies for the annotation of an image and/or region of interests (ROI) inside an image, as well as vocabularies to represent the pre and post processing states of an image, (ii) text entities - covers annotations from the text that are associated with an image (e.g. image captions) and provides semantic representation for NLP algorithm outputs, (iii) a provenance model - that contributes towards the maintenance of annotation versioning.

Semantic mapping, the process of mapping between structured data and semantic vocabularies, and triplification are the core internal processes of iCyrus. While there are several automatic mapping generation tools (Virtuoso RDF view [22], D2RQ mapping [23], Triplify [24]) which serve as useful mapping engines for the initial stages of a project, they fail to fully accommodate the complex domain semantics required by most real-time applications, including ours. Almost all relational database to RDF conversion frameworks follow the same principles, namely the mapping of table names to class names, the translation of column names into predicates and the contents of databases are populated as instances of class categories. We developed a customizable domain-dependent schema mapper for the curation of initial mappings based on our MySQL stored data. After refining our mappings to make them more flexible and fully compatible with our domain, a D2RQ server with a Jena Model API [25], which holds the *ModelD2RQ* class, was used to invoke the external semantic mapping file. With the help of the Jena Model API, items of interest were extracted and linked with URIs, then converted to RDF by a *dumpGenerator* method. Although available triplification frameworks permit the display of non-RDF data in triple format, their query access times are quite slow, which can affect the performance of applications built on them. A viable solution is to store the data in a triple-store to improve access times. We selected Sesame because of its free availability and the presence of an API that allowed us to access the triple store programmatically and create a customized Java Code that transferred the RDF dump into a triple store.

³ <http://bioportal.bioontology.org/ontologies/SEDI>

⁴ <http://cbakerlab.unbsj.ca/unbvps/>

2.3 Stage 3: RDF Exposing and Integration Phase

To expose the linked open image data, we configured a Sesame triple store and deployed *SNORQL*, an AJAX front-end for exploring RDF SPARQL endpoints [8]. *SNORQL* permits users to view and export data in XML, XHTML and JSON - a javascript object notation which is a popular format among web developers. When querying a virtual RDF graph, *SNORQL* [26] translates the SPARQL query back into SQL and fetches the results from the database. While data extraction through a triple store is quite fast and trouble free, data extraction from a virtual RDF graph, particularly for string comparisons, is slower and sometimes results in errors due to system specifications. To facilitate end user navigation for technical users through iCyrus linked data, we deployed *Pubby* [27], a Linked Data interface for local and remote SPARQL protocol servers. *Pubby* allows for triple store access through SPARQL queries and yields dereferenceable URIs by rewriting the URIs found in the SPARQL-exposed dataset into the *Pubby* server's namespace, which interlinks related information from external data sources.

The *SNORQL* endpoint allows external systems to access the YIF content. For example, text-based information retrieval systems may need to integrate image data for improving document ranking and for identifying related publications. To this end, such a system would use the endpoint to retrieve keywords found within images and image captions, and would combine these keywords with text from the body of the associated document. The resulting hybrid word vector can then be used to match user queries with available documents. Additionally, finding related publications can be improved by integrating relationships that exist between images of different publications. This information is currently available through the YIF web site: Given a particular image, YIF calculates the top related images by comparing the text content of that image to other images across the database [14]. This list of related images is accessible through the *SNORQL* endpoint, and may be used to identify related publications based on both text and image information.

2.4 Stage 4: iCyrus and SADI web services

To demonstrate iCyrus' usability as a semantic image API in general and a valuable resource for biomedical knowledge exploration in particular, we developed thirty web services using the SADI framework to advertise their availability. Primarily, the SADI framework is designed to achieve semantic interoperability between different web services as outlined in section 2. SADI integration allows for interesting possibilities for extending the YIF functionality using other biomedical data resources. Currently, text queries submitted to YIF via the web site are stemmed before image retrieval; however, no query expansion is performed, which may result in missed images where image captions contain synonymous terms to the user query. For example, queries including gene names retrieve images matching the query text, but not images that contain an alternative name for that gene. Combining the iCyrus SADI service, *getGeneAlterNames*, with an existing SADI service, *getImagesFromText*, elegantly adds query expansion to the workflow, thus retrieving a more complete set of images. This could support a potential query: *Display a gene, its alternate names, and chro-*

mosomal location. The *getGeneAlterNames* SADI web service takes a gene ID as input and searches through HUGO Gene Nomenclature Committee (HGNC) terms to retrieve all synonyms or alternate names of the gene along with associated general information like chromosomal location and HGNC ID. In order to execute a SADI service, the inputs and outputs have to be defined in an RDF graph before the SADI service generator method can be called [12, 17]. To build the *getGeneAlterNames* service, we defined and published the input and output classes in our iCyrus SADI service ontology⁵. To get the consolidated output view, we kept the input type in all of the iCyrus services in the *xsd:string* format. Fig. 2 shows an RDF input graph for the *getGeneAlterNames* service. Here, *semanticImageApiSadiServices:getGeneAlterNames_Input* is the input class, which takes a string parameter as input. The service algorithm accesses the HGNC database as a backend and looks for HGNC IDs; then, it links the acquired information with the service output class to improve resource interoperability and reusability. This service can work independently or in combination with other SADI services to get a consolidated RDF output. In the next section, we will show how we utilized this service to accomplish a task.

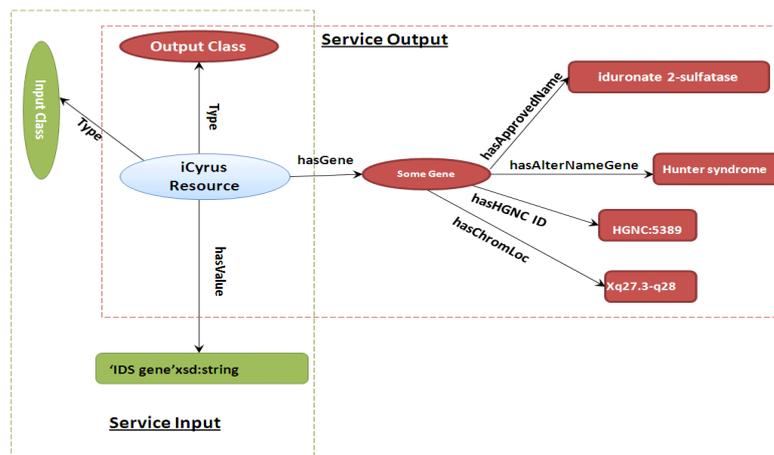


Fig 2. A graphical view of the *getGeneAlterNames* SADI service

3. iCyrus with Federated Query Client

We configured iCyrus' SADI services registry with the SHARE federated query [28] engine to illustrate SHARE's automatic information discovery feature. As an example, we ran the following query: *Display images of the 'IDS gene' from all documents along with their captions and extend the search with alternate gene names*. The SPARQL query syntax can be viewed in Fig. 3. After being invoked, the SHARE client studies the query by reading the input file from the 'FROM' clause, then fetches all the defined predicates in it, which in our query are *hasFigureID*, *hasFigureCaption*, *hasFullText*, *hasGene*, and *hasAlternateGene*. Subsequently, it invokes a method that seeks relevant web services by crawling through the service registry, looking

⁵ <http://cbakerlab.unbsj.ca:8080/ontologies/semanticImageApiSadiServices.owl>

for a match on the predicates. Once matches are found, it populates the virtual knowledge base by running all the discovered services. In our example, SHARE discovered *getGeneAlterNames* and *getImagesFromText* as related services. To process the defined query, SHARE first looked for the IDS gene through its virtual knowledge base and stored the matches in a separate results file; it then found an alternate name for IDS, iduronate 2-sulfatase (*Hunter Syndrome*), and crawled through the virtual knowledge base once again to add/append to the results file. Upon query completion, the results are displayed to the user through a web interface. Fig. 3 depicts the figure ID and caption for only two figures, a limit specified during query composition.

Query form

Enter a SPARQL query in the text box below and click the submit button.

[A list of example queries is available here.](#)

[Learn how to build your own query here.](#)

[A list of predicates is available here.](#)

SPARQL query:

```

PREFIX uysie: <http://cbakerlab.unbsj.ca:8085/unbvps/uysie#>
SELECT DISTINCT ?figid ?figcaption
FROM <http://cbakerlab.unbsj.ca:8080/cardioSHARE-yusal/input.rdf>
WHERE {
# Document -> Figure ID
?Document icyrusSADInt:hasFigureID ?figid.
# Figure -> Figure Caption
?Figure icyrusSADInt:hasFigureCaption?figcaption.
# Document -> Full text
?Document icyrusSADInt:hasFullText ?DocText.
# Gene Name -> Alternate gene name
?Gene uysie:hasGene 'IDS gene'.
    uysie#hasAlternateGene ?altGeneName.
FILTER(REGEX(?DocText,'IDS gene',?altGeneName))
}
Limit 2

```

Query results

figid	figcaption
PMC3150403/2Fpone.0022951.g001	Homology alignment of amino acid sequences among IDS, N-acetylgalactosam ...
PMC3262053/2Fnhms335437f2	MPS Non-reducing end carbohydratesThe defective enzyme for each MPS s...

Fig 3. SHARE screenshot demonstrating iCyrus SADI queries

4. iCyrus Plugin for DOME0

Domeo [29] is an extensible web application enabling users to visually and efficiently create and share ontology-based standoff annotation on HTML or XML document targets. The tool supports manual and semi-automated annotation with complete provenance records, as well as personal or community annotation with access authorization and control. Domeo can be easily extended through the development of software plugins, prompting us to facilitate the creation of annotation mashups using image metadata accessible from the iCyrus. We developed a server-side software plugin that, given a PubMed Central document, can query the iCyrus SPARQL endpoint for all the metadata related to the figures in the document. The metadata are then translated into a JSON format that is easily consumed by the Domeo client. By leveraging the iCyrus plugin, Domeo offers an enhanced PubMed Central extraction and annotation pipeline. In the DOME0 client available image metadata are displayed under each figure in the image summary panel. This includes relevant annotations

from an image caption, the title, the PMID and the publication year of each image. Hyperlinks allow users to browse an extended version of the metadata in the original webpage of Yale Image Finder and in the exploratory pages of the iCyrus project. The integration with Domeo also allows for the gathering of user feedback, including corrections and additions pertaining to YIF metadata⁶.

5. Concluding Remarks

Yale Image Finder and other portals such as NIH Images, Hardin MD, NCI Visual Online, and PHIL (Public Health Image Library), which are not semantically enabled, permit the search for images based on specific biological queries like AIDS, cancer, or other keywords. In this paper, we report on extending the functionality provided by Yale Image Finder using a combination of technologies for semantic publishing of scientific data. The core iCyrus implementation extends the YIF system and makes it interoperable with other existing semantic or syntactic frameworks by making data easily reusable, by providing feature search inside images, and by running automatic updates. From an end user perspective, iCyrus aids both technical and non-technical users from a range of biomedical domains in building mashups involving image data. Technical users can access the API programmatically and can use the resource for precise knowledge extraction or to build new mashups integrating iCyrus with other life science resources. Employing the image browser component of iCyrus, a non-technical user can navigate through the linked image data and can acquire the desired information graphically. The benefits of iCyrus include enhanced maintainability. Update to YIF in its original configuration requires modification of the Lucene index at the backend, which is considered to be error prone and labor-intensive. Moreover, index managed data is rather limited in scope, discouraging its integration with external resources where common data descriptions exist. In contrast, iCyrus stores its data in a state of the art format, RDF, which overcomes the interoperability issues associated with YIF. The underlying iCyrus technology strictly adheres principles to the semantic web a principle that makes its extension easy and automatic. Similarly, its compatibility with the LOD cloud opens up new avenues for ad-hoc integration of image data to solve outstanding challenges whereas YIF and similar systems restrict a user with read only access.

Our combined approach, deploying iCyrus with SADI, SHARE, and Domeo, makes the framework a universal solution in the bio image informatics domain where tools with diverse functionalities can be accessed as a single application, making it possible to query across multiple data sets and algorithms concurrently as well as to harvest new annotations through crowdsourcing for later reuse in image discovery. Like YIF, iCyrus is still a research prototype and the illustrations of its services presented here have confirmed some of the limitations of semantic systems identified and discussed elsewhere [30]. The key limitation is a usability challenge, namely that of query composition where the construction of SPARQL queries using the SNORQL or SHARE clients is more complex than a keyword search in YIF. While recent work has significantly eased the burden of creating SPARQL queries for

⁶ *Graphical view of iCyrus, Domeo integration. <http://tinyurl.com/icyrusdomeoplugin>

ontologies that use alphanumeric identifiers, by using more human-friendly labels during query formulation, more work is necessary to abstract this powerful albeit technical interface so that it becomes more users friendly to life scientists. Some efforts toward this goal, from natural language to SPARQL, are ongoing [31]. RDF graph browsing and faceted browsing also offer an affable substitute to SPARQL query composition and allow the user to dynamically migrate from one entity to another via incoming or outgoing properties. We have further introduced the use of SADI services and the query client SHARE into the iCyrus framework to facilitate access to data in the API and distributed services. Akin to other semantically powered federated query engines like DARQ [32] and ANAPSID [33,] introduced to facilitate transparent query to distributed linked data resources, SHARE's role in the iCyrus infrastructure is to coordinate the orchestration of SADI services designed specifically to integrate desired image related data on the fly [28]. SHARE suffers from a relatively slow query execution compared with other known clients for SADI [34]. In addition, at the time of writing, the only the available browser-based query interface compatible with the SHARE client is Sentient Knowledge explorer [35]. Use of other clients such as HYDRA and its graphical query composition interface [34, 36] may in future result in enhanced performance and further realize the existing investment we have made in building SADI services for image data.

References

1. Biota: The Biodiversity Database Manager. <http://viceroi.eeb.uconn.edu/Biota>.
2. Wishart, D.S et al. "DrugBank: a comprehensive resource for in silico drug discovery and exploration." *Nucleic acids research* 34.suppl 1 (2006): D668-D672.
3. Goldberg IG, Allan C, B, Creager D, Falconi A, Hochheiser H, Johnston J, Mellen J, Sorger PK, Swedlow JR. The Open Microscopy Environment (OME) Data Model and XML File: Open Tools for Informatics and Quantitative Analysis in Biological Imaging. *Genome Biol.* 2005; 14:R47.
4. Ahmed WM, Lenz D, Jia L, Robinson, JP, Ghafoor, A. XML-Based Data Model and Architecture for a Knowledge-Based Grid-Enabled Problem-Solving Environment for High-Throughput Biological Imaging. *Information Technology in Biomedicine, IEEE Transactions on.* 2008; 14(2): 226–240.
5. Plazi: Access to Taxonomic Literature. <http://plazi.org/>
6. Nigel, S., Hall, W., Berners-Lee, T., "The semantic web revisited." *Intelligent Systems, IEEE* 21.3 (2006): 96-101.
7. Resource Description Framework. "Model and Syntax Specification." *W3C Recommendation* 22.
8. Prud'Hommeaux, E., Seaborne, A., "SPARQL query language for RDF." *W3C recommendation* 15 (2008).
9. Samwald, M et al. "Linked open drug data for pharmaceutical research and development." *Journal of cheminformatics* 3.1 (2011): 19.
10. Bukhari, A. C., Baker, C. J. (2013). The Canadian health census as Linked Open Data: towards policy making in public health. In *9th International Conference on Data Integration in the Life Sciences; July 11-12, 2013*;
11. Belleau, François et al. "Bio2RDF: towards a mashup to build bioinformatics knowledge systems." *Journal of biomedical informatics* 41.5 (2008): 706-716.
12. Bukhari, A. C., Klein, A., Baker, C. J. (2013, January). Towards Interoperable BioNLP Semantic Web Services Using the SADI Framework. In *Data Integration in the Life Sciences* (pp. 69-80). Springer Berlin Heidelberg.

13. Faulconbridge, A., et al. "Updates to BioSamples database at European Bioinformatics Institute." *Nucleic acids research* (2013): gkt1081.
14. Songhua, X., McCusker, J., Krauthammer, M., "Yale Image Finder (YIF): a new search engine for retrieving biomedical images." *Bioinformatics* 24.17 (2008): 1968-1970.
15. Kuhn, T., Krauthammer, M., Image Mining from Gel Diagrams in Biomedical Publications. (SMBM). 2012.
16. Bukhari, A. C., Krauthammer, M., & Baker, C. J. SEBI: An Architecture for Biomedical Image Discovery, Interoperability and Reusability based on Semantic Enrichment. SWA4LS 2014
17. Wilkinson, M., Vandervalk, B., McCarthy, L.: The Semantic Automated Discovery and Integration, Web service Design-Pattern, API and Reference Implementation. *Journal of Biomedical Semantics* 2(1), 5–23 (2011)
18. Verlic, M. (2012). LODGrefine-LOD-enabled Google Refine in Action. In *I-SEMANTICS (Posters & Demos)* (pp. 31-37).
19. Sauer mann, L., Cyganiak, R., & Völkel, M. (2011). Cool URIs for the semantic web.
20. Miller, George A. "WordNet: a lexical database for English." *Communications of the ACM* 38.11 (1995): 39-41.
21. Bukhari, A. C., Nagy, M. L., Krauthammer, M., Ciccarese, P., & Baker, C. J. BIM: An open ontology for the annotation of biomedical images. ICBO 2015
22. Erling, Orri, and Ivan Mikhailov. "Virtuoso: Rdf support in a native rdbms." *Semantic Web Information Management* (2010): 501-519.
23. Bizer, C., Seaborne, A., "D2RQ-treating non-RDF databases as virtual RDF graphs." (*ISWC2004*) 7 Nov. 2004.
24. Sören, A. et al. "Triplify: light-weight linked data publication from relational databases." *Proceedings of the 18th international conference on World wide web* 20 Apr. 2009: 621-630.
25. Brian, M., "Jena: A semantic web toolkit." *Internet Computing, IEEE* 6.6 (2002): 55-59.
26. Bizer, C., Seaborne, A.: D2rq - treating non-rdf databases as virtual rdf graphs. In: *World Wide Web Internet and Web Information Systems*, p. 26 (2004)
27. Cyganiak, R., Bizer, C., "Pubby-A Linked Data Frontend for SPARQL Endpoints." *May* 28 (2008): 2011.
28. Wilkinson, M., McCarthy, L., Vandervalk, B., Withers, D., Kawas, E., Samadian, S.: SADI, SHARE, and the in silico scientific method. *BMC Bioinformatics* 11(12), S7 (2012)
29. Ciccarese, P., Ocana, M., Clark, T., "Open semantic annotation of scientific publications using DOME0." *Journal of biomedical semantics* 3.Suppl 1 (2012): S1.
30. Riazanov A, Laurila JB, Baker CJO: Deploying mutation impact text-mining software with the SADI Semantic Web Services framework. *BMC Bioinformatics, Proceedings ECCB 2010 Workshop*
31. McCarthy L, Vandervalk, B., Wilkinson, M. SPARQL Assist Language-Neutral Query Composer. [<http://arxiv.org/abs/1012.1666>] *Semantic Web Applications and Tools for Life Sciences*; December 10th, 2010; Berlin, Germany 2010.
32. Bastian, Q, Ulf Leser. "Querying distributed RDF data sources with SPARQL." *The Semantic Web: Research and Applications* (2008): 524-538.
33. Acosta, Maribel et al. "ANAPSID: an adaptive query processing engine for SPARQL endpoints." *The Semantic Web–ISWC 2011* (2011): 18-34.
34. Riazanov, A., Klein, A., Shaban-Nejad, A., Rose, G. W., Forster, A. J., Buckeridge, D. L., & Baker, C. J. "Semantic querying of relational data for clinical intelligence: a semantic web services-based approach". *J. Biomedical Semantics*. 2013. 4(9).
35. Wilkinson M. D., McCarthy, L., The SADI plug-in to IO informatics' sentient knowledge explorer. (SWAT4LS '11). 2011. ACM, New York, NY, USA, 116-118.
36. HYDRA: <http://ipsnp.co/demo-videos/>