A Visualization Service for the Semantic Web

Sean M. Falconer[1], Chris Callendar[2], and Margaret-Anne Storey[2]

[1] Stanford Center for Biomedical Informatics Research, Stanford University, USA
[2] University of Victoria, Canada
sfalc@stanford.edu, {ccallend,mstorey}@uvic.ca

Abstract. The Web presents an opportunity to openly collaborate and share visualizations of semantic web data. Many desktop tools for visually exploring ontologies exist; however, few researchers have investigated how visualizations could be used in an online environment to enhance the semantic web. In this paper, we present our experience with developing a visualization service for the semantic web. We discuss the advantages and challenges with moving to a web-based platform, as well as the features of the service through several case studies. We reflect on this experience and provide recommendations for future work and data integrations.

1 Introduction

Over the past decade, many ontology visualization tools have been developed to help users understand, create, and manipulate ontologies [8]. Visualization of ontologies is often considered to be a benefit as we can rely on powerful human cognitive and perceptual abilities not possible with pure textual representations.

Tools and repositories for semantic web resources and ontologies have been migrating from desktop environments to the interactive Web. Yet online visual tools for interacting, understanding, and browsing these information repositories and ontologies is sorely lacking. For example, Swoogle[1], contains over 10,000 ontologies in its index. However, the tools available for browsing, exploring and interacting with these ontologies are still primarily relegated to the desktop.

Much of the research focus for ontology visualization has been on how to better support navigation of the ontology graph within desktop tools. Little research has looked at how visualization could potentially play a larger role in supporting collaboration and adoption of semantic web technology. The web browser has become the universal interface, allowing researchers and application developers to reach a far larger audience than previously thought possible [3]. The Internet makes it easy to share results and collaborate over vast distances. There is no need to download special tools, and with the advent of Web 2.0, the Web has grown into an interactive environment.

In this paper, we present a web-based visualization service, called FlexViz. We discuss advantages and challenges with developing web-based visualizations, features of our service, and several example case studies.

[1] http://swoogle.umbc.edu

P. Cimiano and H.S. Pinto (Eds.): EKAW 2010, LNAI 6317, pp. 554–564, 2010.

2 Web-Based Visualization

The Web is gradually becoming a platform for visualization. Internet-based visualizations like tag clouds are now a standard navigational component of many websites. The NameVoyager [10], which lets users interactively explore historical name data, drew 500,000 visits within two weeks of its launch. Many Eyes, a project from IBM, has combined visualization with social computing[2]. Users of the site can upload new data sets and apply different visualizations to the data to facilitate analysis. The visualizations and data are shared, allowing other users to apply new views, discuss particular visualizations, and rate views.

Despite this trend, few researchers (see [7] for exception) have explored the utility of web-based visualizations of semantic web data. The Web, as a platform for exploring and interacting with structured information, is very attractive. Below we explore some of these potential advantages.

Reach a larger audience: One of the greatest features of the Web is instant usability. Downloads of special software are unnecessary, the browser has become the universal interface. The accessibility of online applications greatly enhance their adoption potential. Part of the success and popularity of the NameVoyager application was due to its public nature. Anyone can use the application, which has instant ramifications for all aspects of the site. The Web opens an application up to the "accidental user".

Social data analysis: Web-based visualization changes the focus of visualization from task-oriented problem solving to social data analysis [10]. Users can engage deeply with online visualizations and web applications, opening the topic area to in-depth social analysis [6]. Again, "accidental users", bring different perspectives and different backgrounds, which may change how the application is used. For example, the various mash-ups that have been created with Google Maps [5].

Reduce task complexity: Integration with existing web applications potentially reduces task complexity. As in the previously mentioned Swoogle browsing example, with a web-based visualization, the steps requiring a user to download and load the ontology into an editor are removed.

Tracking user behavior: Tracking in desktop environments can be difficult, as the information must be collected on the client and then forwarded to a server. However, the deployment of visualizations on the Web, particularly as hosted services, make collecting this information seamless and straightforward. An application can collect rich statistical information about their users' behaviors. This data can later be used to help improve the tool and service as well as test the adoption of new features.

Even with these advantages, few web-based ontology visualizations currently exist. Moreover, the existing visualizations, like AmiGO[3], lack interactivity and

[2] `http://manyeyes.alphaworks.ibm.com`
[3] `http://amigo.geneontology.org`

are not tightly integrated with the surrounding application. Benjamins *et al.* state that one of the six essential challenges for the semantic web is visualization support [1]. Addressing this challenge is difficult. There are several technical hurdles that must be overcome. Below we discuss several of these technical issues.

Performance: Ontologies can be very large and complex. For example, SNOMED CT, an extremely comprehensive medical terminology, contains approximately 300,000 terms and over 1 million relationships[4]. Many existing visualization tools rely on the ontology being completely stored in memory, which is not feasible for large ontologies like SNOMED CT. Web-based technologies for creating visualizations often have limited rendering performance and severe memory constraints.

Screen real estate: Even relatively small ontologies, can be difficult to visualize. Often, ontologies are visualized as a node-link diagram, which is susceptible to labeling issues and a cluttered display [8]. Managing these issues in a desktop environment is a design challenge, but on the Web, there are further constraints. Web browsers often use between 100 to 200 pixels for tabs, menus, and navigation controls. This severely limits the available canvas for a visualization.

Integration: Seamless integration with existing web applications is key to the successful adoption of a web-based visualization. This can be potentially difficult. With newer, less widely adopted technologies, users of the tool may be forced to download plugins as with Microsoft's Silverlight technology. Also, multiple technologies may need to be integrated to allow the existing web infrastructure to communicate with the visual elements.

Development environment: The choice of a development environment for supporting the visualization is difficult. A variety of potential technologies exist; SVG, Java applets, Flash, Silverlight, Firefox canvas object, etc. Each technology has advantages and disadvantages and depending on the nature of the application, the choice in technology may change.

Although there are advantages to supporting more web-based visualizations of semantic web services, there are also many challenges that must be faced. In the next section, we discuss the development and design philoposhy of FlexViz.

3 Developing FlexViz

Following a similar design philosophy as adopted by many Web 2.0 services, we wished to make FlexViz easily extensible, thus allowing the Web community to apply new data sources as they see fit. The first version of FlexViz was released in January 2008. Since then, there have been three major releases[5].

3.1 Features

The feature requirements for FlexViz were based on our years of experience developing tools for ontologies and structured data sources. Below, we provide an overview of some of the FlexViz functionality.

[4] `http://www.nlm.nih.gov/snomed`

[5] FlexViz source code: `https://sourceforge.net/projects/flexviz/`

Layouts: FlexViz supports many different graph layouts. These include vertical/horizontal tree layouts, circle layouts, spring layouts, and grid-based layouts. Layouts provide a way to automatically position nodes for legibility. New layouts can be easily integrated and applied.

Filtering: FlexViz allows a developer to specify different types of nodes and arcs. The complexity of the represented graph can be reduced by applying filters to specific node and arc types. Additionally, specific nodes can be filtered via a right-click menu. The menu allows the user to hide a particular node or the children of that node. User-driven filter helps address some of the performance concerns with web-based visualizations.

Navigation: To help support scalability, besides simple filtering, FlexViz supports step-wise node expansion. The initial view for a graph is restricted to the local neighborhood of the in-focus node. That is, only the focal node along with its immediate neighbors based on un-filtered arc types is displayed. Nodes that can be expanded further are shown with a plus icon. For simple graphs, double-clicking these nodes expands the node to display its local neighborhood based on the in-memory graph representation. However, for larger data repositories, the double-click event potentially invokes a web service call to retrieve the local neighborhood for the selected node. This call takes place asynchronously, retrieving the relevant information and eventually updating the graph. The data is cached for fast retrieval on repeated calls.

Interaction: Nodes can be dragged and re-positioned as the user desires. The graph also supports panning and zooming. The contents of the graph can be re-positioned by toggling either the "fit to screen" or "expand graph" options. The "fit to screen" button will re-position all nodes and arcs to be displayed within the viewable canvas area of the graph. Depending on the size of the graph, overlapping can occur. The user can either manually move nodes or use the "expand graph" option to make items visible. These options help to address the canvas real estate issue.

Search: The label associated with a node can be searched for within the in-memory representation of the graph. Term completion for this search is also available. The matching node is highlighted and the graph view pans to display the node. The search interface can also be extended for specific applications to support asynchronous searching (see Section 4.1 for discussion).

Exporting: FlexViz views can also be exported, either as XML or as a static graphic. For specific applications, this feature can be extended to facilitate data-specific extraction (see Section 4.1 for discussion).

The above features are some of the basic features support provided by FlexViz. The functionality can be easily extended depending on the requirements of the given domain and task. In the next section, we explore three example extensions and integrations of the FlexViz toolkit, demonstrating the flexibility available.

4 Visualizing Semantic Web Data

The FlexViz toolkit provides a generic library to create web-based graph representations. However, the usefulness of the toolkit derives from the applications that extend it. In this section, we discuss three such extensions: BioPortal integration, FOAF support, and Bio-Mixer. The extended application stack for FlexViz is shown in Fig. 1.

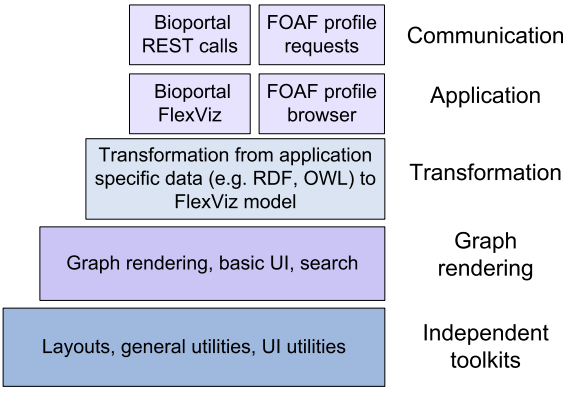

Fig. 1. FlexViz application stack. The top layer consists of application specific communication services, like REST calls to BioPortal. The second layer consists of specific application instances of FlexViz. The third tier is the transformation layer, where application specific data like RDF or OWL is transformed into the FlexViz model. The fourth and fifth tiers are the generic FlexViz toolkit libraries.

4.1 BioPortal

BioPortal has been developed as part of the National Center of Biomedical Ontology (NCBO. BioPortal provides a repository of biomedical related ontologies and resources. Ontology exploration is supported via searching or selection of a particular ontology. Ontology concepts can be explored through a class browser, similar to the class browser found in Protégé. Users of BioPortal may explore the existing ontologies to understand and learn a new ontology; it is also a critical step for ontology evaluation, sharing, mapping, and resource management.

The ontology view is essentially an indented list of the ontology classes, which forces the ontology's representation into a tree structure rather than a graph. As a result, the tree only displays inheritance relationships and not property relations. Furthermore, classes in an ontology potentially have multiple parents, which in a tree cannot be adequately represented. The class must be duplicated for each parent, which has been shown to be confusing to users [4].

To help address these deficiencies, we worked with the BioPortal development team to provide FlexViz as an ontology browsing service. The BioPortal specific extension of FlexViz is stored on our own servers, and seamlessly integrated

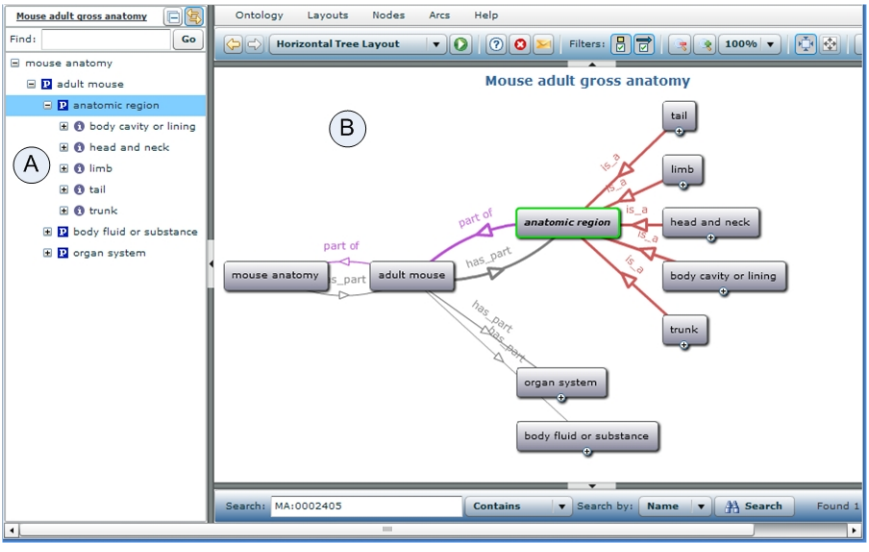

Fig. 2. Visualizing the Mouse Adult Gross Anatomy within BioPortal. (A) shows the class tree browser, while (B) shows the FlexViz graph rendering. The term *anatomic region* is the current focal term, and this is synchronized between the two views.

into BioPortal (see Fig. 2)[6]. All communication with BioPortal occurs through the REST services provided by BioPortal for accessing ontology-related data. All expansion, search, and focus operations result in a service call to BioPortal.

Several ontology specific extensions were incorporated into the BioPortal FlexViz application. We included an extension to the concept node context menu called "Link to concept". Similar to the "Link" feature in Google Maps or YouTube, this feature allows a user to share a particular visualization with another person via a specific URL or through auto-generated HTML that supports an embedded view. Both approaches allow users to share, link, or embed specific visualizations into their web pages and blogs. We incorporated this feature to facilitate collaboration and provide a light-weight mechanism to deploy instances of FlexViz wherever users deem appropriate.

Natively, a FlexViz view can be extracted as an image, which is useful for scientists using BioPortal when they wish to embed a particular ontology snapshot into a research paper or document. However, we are also working on supporting OWL and RDF extraction. This feature would support users with the task of extracting a portion of an ontology.

Figure 2 shows an example of the FlexViz integration in BioPortal. The figure shows a visualization from part of the Mouse Adult Gross Anatomy, where *anatomic region* is the focal term. The tree representation displayed in Fig. 2(A) is an extension of the base user interface implementation of FlexViz. This

[6] Source code: `https://bmir-gforge.stanford.edu/gf/project/flexviz/`

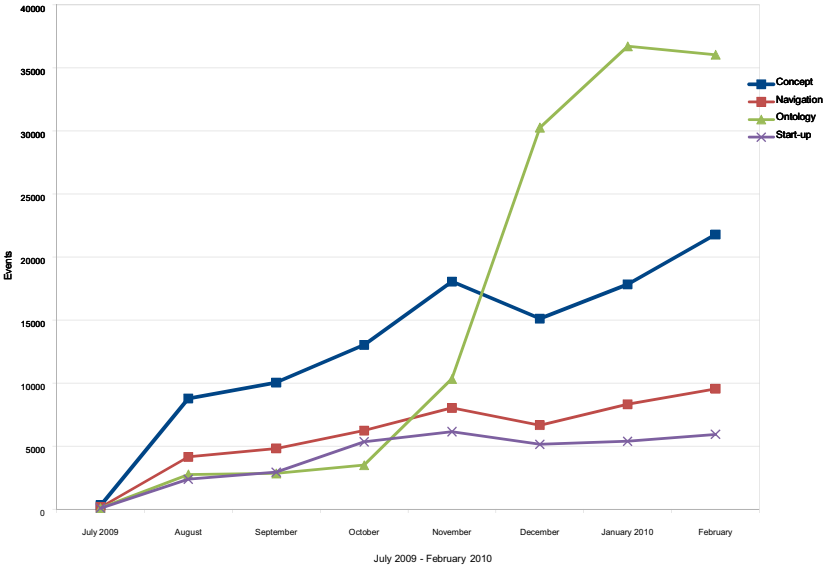

Fig. 3. Concept, navigation, ontology, and start-up events over an eight month period

extension helps to support synchronization between the tree and graph view as well as allow us to gather statistics about user browsing behavior.

Since July of 2009, we have been logging the usage of FlexViz. Figure 3, shows the number of concept, navigation, ontology, and start-up events that have been executed with FlexViz over an eight month period. Concept events are captured when a user request requires information about a particular concept, while navigation events are interaction events that do not require specific data to be loaded, for example, when a graph layout is applied. Ontology events are captured when a new ontology is selected and loaded and finally, start-up events occur when the visualization is first loaded.

4.2 FOAF

The Friend of a Friend (FOAF) project is attempting to create a web of machine-processable descriptions of people, the relationships between them, and the things they do [2]. With the growing number of social networking sites like Facebook, Twitter, Twine, and MySpace, Internet users must re-create their public profiles and the links to their friends over and over within each service. FOAF provides a standard, structured means to deal with this problem.

A FOAF profile consists of a RDF file describing a particular individual and the description of their friends. A user can make their profile available online, and potentially the profile could be shared between existing social networking services. Collections of profiles and the interconnections between people and their friends form a graph structure. Provided the friends listed in the `foaf:knows`

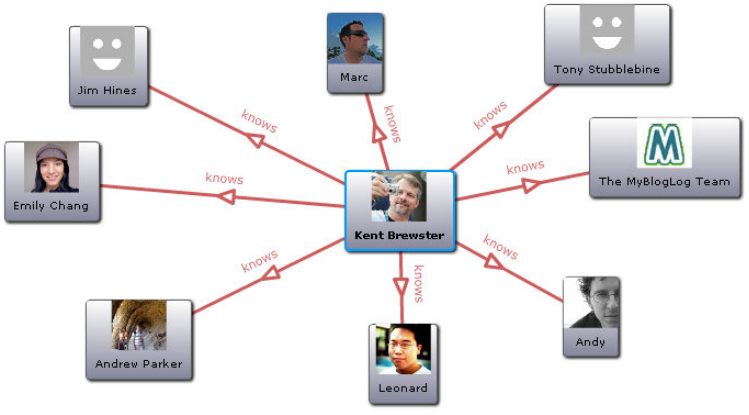

Fig. 4. Screenshot of the FOAF Browser canvas

property also have pointers to their own FOAF profiles, the graph structure can be walked by requesting the given RDF FOAF file through HTTP. In this example of FlexViz, we extend the toolkit to make use of this explicit profile graph structure. The extended version, called the FOAF Browser, helps users browse existing FOAF profiles and the interconnections between them.

We integrated with FOAF to help demonstrate the flexibility of our service, as FOAF is one of the most well known examples of the semantic web. Similar to the BioPortal integration, we created extensions to FlexViz at the *Communication*, *Application*, and *Transformation* layers (see Fig. 1). The communication extension uses HTTP to download the RDF files associated with a given FOAF profile, while the application extension consists of user interface extensions specific to FOAF, and finally the FOAF data is parsed and transformed into the FlexViz model in the transformation extension.

Figure 4 shows an example of the FOAF Browser rendering a single FOAF profile. The FOAF Browser uses the `foaf:depiction` property to render a graphical representation of the given person. If this information is not available, a default graphic is used. The nodes (people) in the graph are labeled using the `foaf:name` property, and all profile property information is available when a user moves their mouse over a given node. Provided a known person has a `foaf:seeAlso` property specified, double-clicking the node results in a new HTTP request to download the corresponding FOAF profile. As a result, the graph is expanded to display the new information.

We re-used the "Link to" idea from the BioPortal integration, thus facilitating users with integrating a visual representation of their FOAF profile into their own web pages. We also extended the filters available for pruning a graph to support specific FOAF properties. Finally, we extended the interaction features of the FlexViz graph to support a zoom effect when a node is brushed by the mouse pointer. Moving the mouse over a node results in that node growing in size in order to make the FOAF user's picture easier to see.

4.3 Bio-Mixer

Bio-Mixer[7] is a web-based mashup environment for supporting the exploration and re-mixing of biomedical ontologies. Ontology terms and mappings can be explored in interactive views such as timelines, lists and graphs. The graph view extends the BioPortal specific implementation of FlexViz (see Fig. 5)[8].

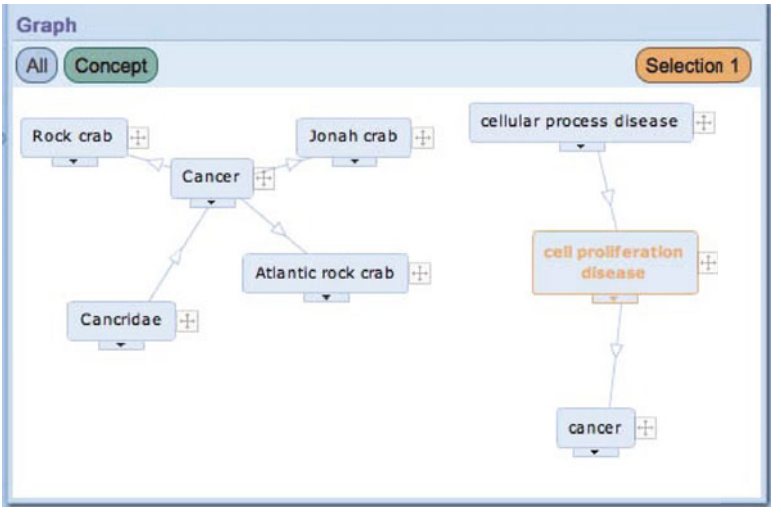

Fig. 5. Screenshot of the graph widget from Bio-Mixer. Two concept neighborhoods are displayed from two different ontologies.

Although Bio-Mixer uses the Bioportal FlexViz implementation, this use case is significantly different. Bio-Mixer demonstrates how customizable the look-and-feel and application behaviors are for FlexViz. For example, Bio-Mixer integrates multiples types of visualizations and supports synchronized interaction between them. To support this, Bio-Mixer uses the public methods in FlexViz for attaching listeners to various graph events. Bio-Mixer also uses completely different styling to seamlessly integrate FlexViz.

5 Discussion

The three case studies presented in the previous section illustrate the flexibility of the FlexViz service for semantic web applications. The design and generality of the toolkit took several iterations and many new ideas are still being explored. Below, in reference to the FlexViz service, we discuss the advantages of web-based visualizations that we previously introduced.

[7] `http://bio-mixer.appspot.com/`
[8] Source code: `http://code.google.com/p/bio-mixer/`

Reach a larger audience: Integration with BioPortal is allowing biomedical ontologies and FlexViz to reach a much larger audience. As more ontology repositories become available online, visualizations like FlexViz will need to be available to help users browse and understand the contents of the available ontologies. As was shown in Section 4.1, there has been a growing adoption of the FlexViz tool over a reasonably short period of time.

Social data analysis: This requires wide-spread adoption, which we are still evaluating. However, early usage is promising. The "Link to" feature will help facilitate this process for users as both users of FOAF and BioPortal can easily share their visualizations.

Reduce task complexity: To evaluate this, extensive user testing and iterative design is required. In our initial design, we tried to leverage other tools built for the Web that tend to use standard user interface controls that users may be familiar with. We developed FlexViz to reduce the complexity with certain types of tasks, like comparing and evaluating ontologies on the Web, and understanding FOAF profile relationships.

Tracking user behavior: We log data from the BioPortal FlexViz integration to help us understand how users are using the visualization. This will help guide us as we make further improvements to the tool.

With respect to the challenges of web-based visualizations mentioned earlier, we have attempted to address *performance* by supporting incremental navigation and search-based navigation. This helps to reduce the amount of data that must be asynchronously requested from our application-specific end-points. User tracking will also help us more easily tailor what is displayed based on previous user navigation patterns. Technology *integration* is getting easier as technologies for the Web improve. Choosing Flex as our development technology has helped to reduce integration overhead. For example, the BioPortal application integrates with FlexViz via an `iframe` with a URL pointer to our server. We have developed the BioPortal extension to have a similar look and feel as the native BioPortal so that users will not realize they are interacting with a different tool. However, FlexViz can be re-styled on demand for any particular application, as shown by the BioMixer integration. Like integration, *development environments* for web-based technologies are improving. The Google Web Toolkit along with various AJAX toolkits are making it easier for developers to create interactive web applications.

6 Conclusion

There are many advantages to developing more interactive visualization tools for the semantic web. Existing work consists primarily of static images or text representations. In our research, it took several iterations and experimentation with various technologies to create a generic toolkit that can readily be extended to a variety of semantic web resources. In the future we plan to continually enhance the performance and functionality of FlexViz. We also plan to perform

more comprehensive evaluation of the FlexViz tool and use the feedback to help improve our design. We propose that online visualizations can greatly enhance user tasks on the semantic web as well as help to promote semantic web technologies. The Web offers researchers the opportunity to provide tools that help users explore new information spaces, collaborate, and take part in social data analysis [10].

Acknowledgments

This work was supported by the National Center for Biomedical Ontology, under road map initiative grant U54 HG004028 from the National Institutes of Health.

References

1. Benjamins, R., Contreras, J., Corcho, O., Gomez-Perez, A.: Six Challenges for the Semantic Web. In: KR 2002 Workshop on Formal Ontology, Knowledge Representation and Intelligent Systems for the Web (2002)
2. Brickley, D., Miller, L.: FOAF Vocabulary Specification (2005), `http://xmlns.com/foaf/0.1`
3. Brodlie, K.: Visualization over the World Wide Web. In: Scientific Visualization Conference, pp. 23–29. IEEE Computer Society, Los Alamitos (1997)
4. Falconer, S.M., Storey, M.-A., Yamauchi, T.: CogZ: Evaluating Cognitive Support for Semi-Automatic Terminology Mapping (2009) (Under review)
5. Floyd, I.R., Jones, M.C., Rathi, D., Twidale, M.B.: Web Mash-ups and Patchwork Prototyping: User-driven technological innovation with Web 2.0 and Open Source Software. In: Proceedings of the 40th Annual Hawaii International Conference on System Sciences, pp. 86–96. IEEE Computer Society, Los Alamitos (2007)
6. Frost, J.H., Massagli, M.P.: Social Uses of Personal Health Information Within PatientsLikeMe, an Online Patient Community: What Can Happen When Patients Have Access to One Anothers Data. Journal of Medical Internet Research 10(3) (2008)
7. Hirsch, C., Hosking, J., Grundy, J.: Interactive Visualization Tools for Exploring the Semantic Graph of Large Knowledge Spaces. In: Workshop on Visual Interfaces to the Social and Semantic Web (2009)
8. Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., Giannopoulou, E.: Ontology visualization methods—a survey. ACM Computing Surveys 39(4), 10–43 (2007)
9. Mackinlay, J., Shneiderman, B.: Information visualization, Using Vision to Think. Academic Press, London (1999)
10. Wattenberg, M.: Baby Names, Visualization, and Social Data Analysis. In: INFOVIS 2005: Proceedings of the 2005 IEEE Symposium on Information Visualization, Washington, DC, USA, vol. 1. IEEE Computer Society, Los Alamitos (2005)