Semantically Enhanced Webspace for Scientific Collaboration

Daniel Harezlak[1], Piotr Nowakowski[1], and Marian Bubak[1,2]

[1] Academic Computer Center CYFRONET AGH
ul. Nawojki 11, 30-950 Krakow, Poland
[2] Institute of Computer Science AGH
al. Mickiewicza 30, 30-059 Krakow, Poland
d.harezlak@cyf-kr.edu.pl

Abstract. The paper presents an approach to constructing a collective Web-based system for knowledge management. The work refers to the concepts and ideas widely promoted by modern web communities, such as user-created and user-annotated content or reliable search mechanisms. Also, formal ways such as ontology-to-model dependencies within collective knowledge are used to build the proposed system. The main focus of this effort is directed towards scientific communities in which large amounts of experimental data need to be classified and verified. For this purpose an enhanced set of available Web tools needs to be assembled and made available as a unified system.

Key words: semantic models, web management, application plan, collaborative research

1 Introduction

The need to represent knowledge by a language that both people and computers can comprehend is obvious and has been proven almost a decade ago [1]. Since then significant effort was invested in combining the formalisms of descriptions that can be parsed by computers with free-text content published by people all over the world, creating the new notion of the Semantic Web. According to the survey [2] the Semantic Web is increasing its momentum by expanding in the areas of Internet computing such as trade, business and travel, not to mention the science domain. Currently we observe that the technologies and tools used for knowledge representation and management are becoming more stable and thus models and services are being proposed [3, 4] to realize the vision of large-scale knowledge integration.

This paper focuses on scientific aspects of the Semantic Web, especially on knowledge- and data-intensive applications, which need to better benefit from the possibilities that become available through the manifestation of the Semantic Web and its extensions. The basic challenge is to combine the collaborative and global methods of using Web resources with individual and geographically-scattered research activities. Many modern approaches try to exploit the techniques available in social Web management such as tagging, ranking or editing

Web content by all users. However, more formal mechanisms are required for scientific purposes. This goal can be supported by applying a strict semantic framework to the way in which Web research is conducted. That is why we propose a solution that incorporates a semantic layer into the available Web management routines to facilitate scientific research.

A need for such environment was observed in the ViroLab project [5] which develops a virtual laboratory [6] to facilitate medical knowledge discovery and provide decision support for HIV drug resistance [7]. Three groups of users have been identified: clinicians using decision support systems for drug ranking, experiment developers who plan complex biomedical simulations, and experiment users who apply prepared experiments (scripts) [8]. An experiment is a kind of processing which may involve acquiring input data from distributed resources, running remote operations on this data, and storing results in a dedicated space, which should not only limit its functionality to the medical disciplines but extend into other areas of science.

The following section contains current achievements in the Semantic Web area. Subsequently, a list of requirements for the proposed solution is presented. The following two sections contain the architecture and proposals of semantic enhancements, followed by current implementation status and a summary with a future workplan.

This work tries to go beyond the present state in building scientific web communities by proposing a system which covers traditional computation infrastructures with lightweight yet reliable and oriented on research web interfaces supporting knowledge management. In principle, it builds upon existing achievements of Semantic Web, however, a novel approach of managing semantic descriptions by web community members is introduced. This requires new combinations of tools for managing semantic metadata and social techniques of editing web content.

2 Related Work

Modern systems in which semantic descriptions are used to represent knowledge generally apply tested and reliable languages, such as OWL [9], which is based on an older RDF specification [10]. Another standard used by a significant group of people is WSMO [11], which provides methods to semantically describe Web services. A problem, however, arises when different groups of researchers try to create descriptions of the same phenomena or elements of reality, resulting in inconsistencies when such descriptions are merged. This requires manual alignment, which can be very time-consuming and inefficient. In order to efficiently build ontologies, a semiautomatic tool is required to provide feedback on preexisting descriptions and enable scientists to further build upon them, thus ensuring coherency.

It is easy to observe that the social Web has evolved into a global collaboration space where people from all over the world exchange experience using systems such as Facebook [12] or Flickr [13]. This way of collaboration has made

the Web an interesting tool for scientific communities, with which to exchange research results and knowledge. Several attempts were undertaken to benefit from those ideas, resulting in applications like [14] and new trends in semantic computing [15]. These attempts, however, still lack general acceptance and stability. Nevertheless, several environments are already available and are being used by minor groups. For example, myExperiment [16], currently in its beta testing phase, is a successor to well-accepted workflow management systems such as Taverna [17] or BIOSteer [18]. The project delivers a Web-based system for sharing workflows among community members; however, the infrastructure does not provide features that allow workflow execution and result management.

3 Requirements

In order to satisfy potential researchers, any new system should ease their work. Therefore, basic requirements should be identified first. Below we present a list which attempts to formalize the process in which research is conducted. In particular, it is assumed that each type of supported scientific research can be aided either by applying a computer system to conduct a virtual experiment (such as a simulation) or by presenting the results in a digital format. Following is a list of basic requirements for a knowledge Web management system.

- *application plan storage* - The notion of an application plan exists in various domains of science and can be described as a list of steps necessary to achieve a certain result. There are many ways to represent such a list. It can be accomplished either by building a workflow (e.g. with the BPEL [19] notation) or by using a script (with any available scripting language). The requirement is to provide a facility for application storage that can be accessed by authorized users. In this way published applications can be discovered, reused, assessed and improved by other scientists.
- *managing application execution* - For the application plan execution to be possible, an underlying infrastructure has to be deployed and a proper application plan execution engine needs to be set up. The whole process of application execution has to be visualized to the user and, if necessary, intermediate results should be delivered.
- *managing scientific results* - The outcome of a research activity should be represented by a result stored in a dedicated database. The results should be properly annotated and classified, available for other scientists for verification purposes.
- *collaborating with other scientists* - The system should provide collaboration tools enabling scientists to discover their work, properly restricted by security and copyright agreements. It also should be convenient to exchange experience and validate other's work within one system.

The presented list of requirements should be supported by a semantic model that facilitates all the functionalities which are to be provided by the proposed system.

Another non-functional requirement is to separate the processes of application development and conducting research. On the one hand developers do not want to be laden with the semantics of a certain research area but only restrict to e.g. data format, computation optimization, etc. On the other hand scientists want to focus only on the research without knowing the specifics of the actual implementation. This requires a certain separation layer provided by the experiment plan. the common parts between the mentioned groups are only notions of experiment plan, input data and experiment result. Developers write experiments together with underlying services, components, etc., wchich require input data and produce results (of course the format of the data is to be agreed between those two groups). The researchers execute the experiments, validate and classify the data being able to manage the semantic layer.

One last requirement that was identified is the cross-disciplinary cooperation of researchers. Creating a global and ultimate ontology seems to be an impossible challange. However, it might be possible to find intersections between them and benefit from what others work on. The approach in the proposed system is to make all the semantic metadata available to all participants. In order to do that an advanced editor is required to assist the researchers in the process of managing the metadata.

4 System Architecture

4.1 General Overview

In Fig. 1 the basic architecture is presented. The system is divided into four layers. At the bottom, the resource layer consists of services and data sources which are used to build application plans using workflow or script notations that provide some level of abstraction. In the same layer the *Metadata Store* and the *Application Repository* are deployed and used to archive semantic data and application plans respectively. The last two components are accessed by the Web application layer (shown in green) directly. The next, yellow layer is the middleware which provides an abstraction over the low-level resources and ensures unified access to the variety of technologies that implement data sources and computational services. In this way access to data and services is seamlessly woven into the notation. The *Application Execution Engine* also maintains the state of the applications during execution.

The third layer, representing Web applications, contains two modules, namely the *Metadata Engine* module and the *Execution Client* module. The first module is the one responsible for managing semantic descriptions available in the system. It also constitutes a filter and a tool that helps users manage the semantic content they provide or browse. Based on the semantic model presented in the next section users are able to:

– import their own semantic descriptions by semi-automatically aligning and mapping them against existing ones,

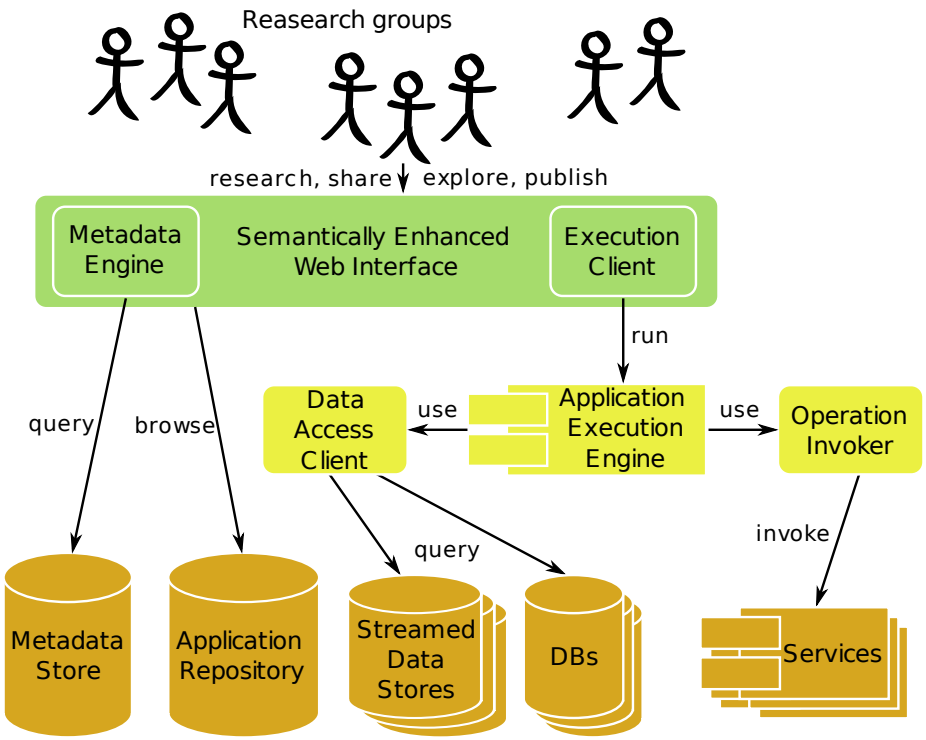

Fig. 1. Basic components of the proposed system.

- browse the existing knowledge by conveniently searching through existing ontology triples,
- quickly obtain application plans, results or publications of interest by providing key words (the whole knowledge space is tagged and annotated),
- tag and annotate the existing objects in the knowledge space.

The second module - *Execution Client* - is responsible for communication with the application execution engine and keeping the users updated with the current execution status using AJAX-oriented techniques (e.g. implemented with the GWT toolkit [20]).

4.2 Metadata Engine

The *Metadata Engine* is the main component which provides the reasoning functionality over the ontologies built within the system. It covers the low-level *Metadata Store* and exposes convenient methods to manage the knowledge structure.

In Fig. 2 a detailed architecture of the *Metadata Engine* is presented. It contains a client that enables it to access the underlying metadata store and facilitates the use of the query language used by the store. The deduction module

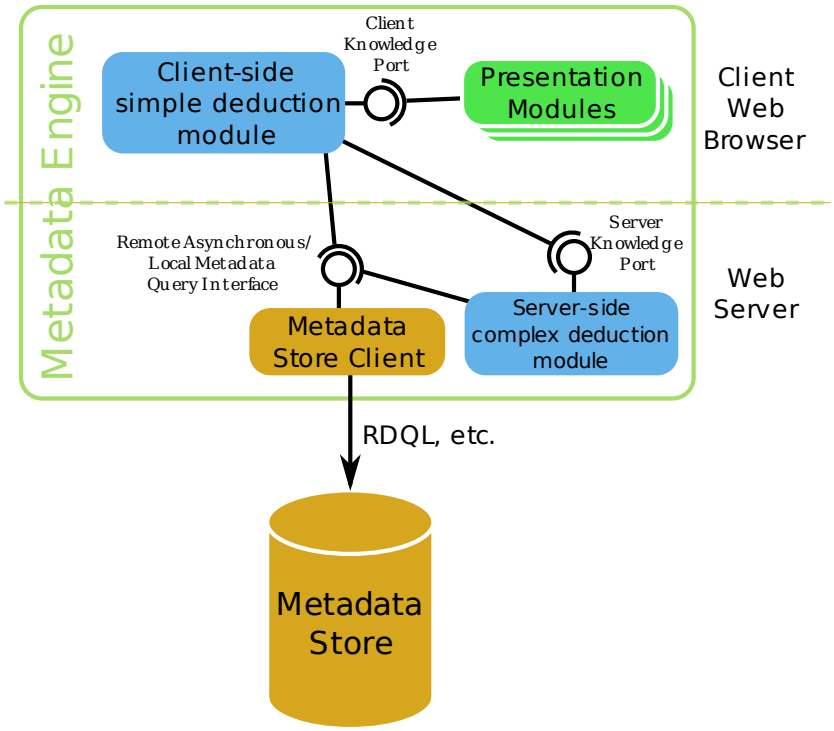

Fig. 2. Internal architecture of Metadata Engine.

is divided into two parts. For simple queries for which response times should be short the part on the client-side is used. It communicates with the client through an asynchronous channel according to the techniques used in web client-server communication models (built over standard request-response model). The calls are made directly by the visual components which concludes with their visual state update. If the queries are more complex then the deduction module on the server-side is used. To the visual components this however is transparent with only longer repsonse times.

5 Semantic Enhancements

5.1 Basic Approach

In Fig. 3 a sample of the ontology model is presented. This model is used as the basis for the *Metadata Engine* module to manage the collaboration space.

The model consists of three parts:

– *Science Domain* - (blue) - This part of the semantic description is extendable by users. This ensures that the model remains dynamic and, when required,

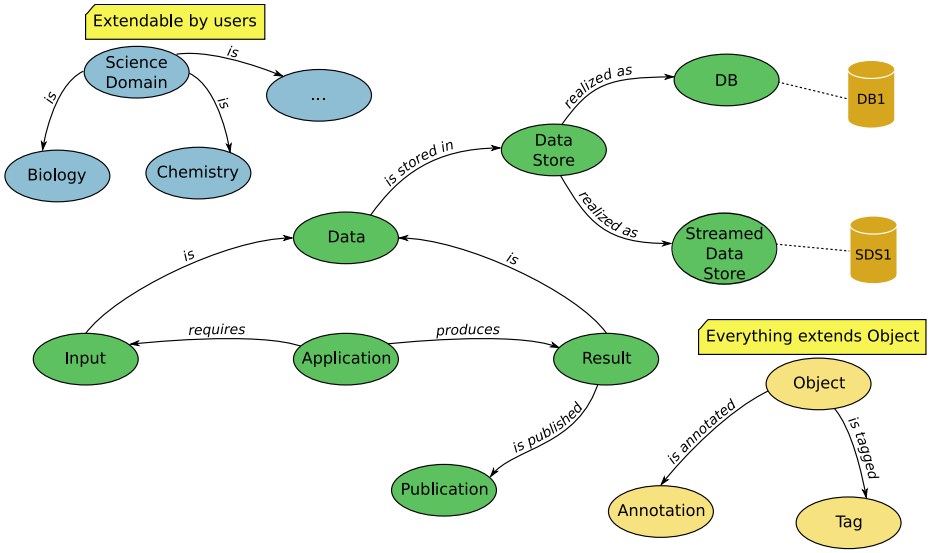

Fig. 3. Samples of semantic descriptions used in the proposed system.

users may add custom ontological descriptions to existing ones. The process is semi-supervised by the system in order to maintain coherency.

- *Basic Model* - (orange) - This model is the core of the application and its basic models. It assumes (in accordance with social Web content management) that every item within the collaboration space may be tagged or annotated. This enables the space to be enhanced by a quick search mechanism or by building a tag cloud (used for space browsing).
- *Application model* - (green) - This ontology model allows the *Metadata Engine* to keep track of the content managed by users. In particular, users are able to submit specific queries that navigate to accurate pieces of data stored in the collaboration space (e.g. list all publications that describe the outcomes of a particular application plan, etc.)

The presented model is just a proposition, showing how the final implementation could look and it remains a subject of ongoing research. It is also possible to test several different models in different research contexts.

5.2 Role and Ontology Management

In order to ensure hierarchy in the process of managing and building the ontologies proper groups need to be modelled with certain permissions. Also, a way of assessing the quality of the ontologies is required to introduce formal models of the management process.

Figure 4 depicts a sample structure of such ontology. the main *Object* node is assigned the *is editable by* relation which specifies what roles are permitted to

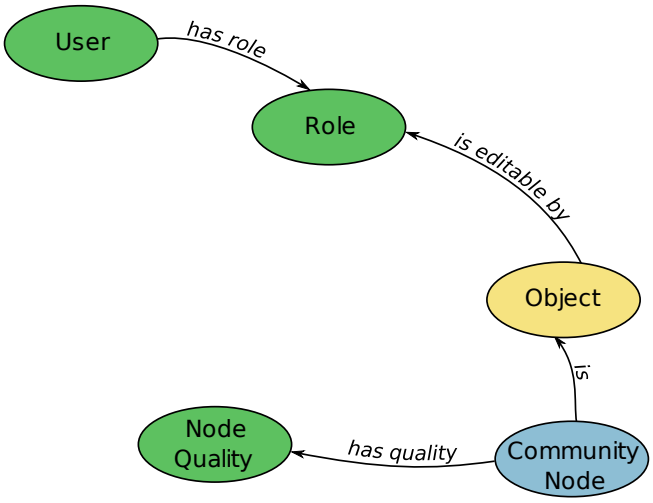

Fig. 4. Role management dependency semantics.

edit a given node. All *Role* nodes are referred by *User* nodes which creates the authorization net in the proposed model.

To enable users with the possibility of extending the current ontology graph a *Community Node* is introduced. This node is inherited by all the nodes created by community members and in the process of collaborative cooperation of scientific communities it is assessed and the quality information is stored in the individuals of the *Node Quality*. The quality will be measured by analyzing statistics of use of such knowledge node(e.g. the more users use and cite a given ontology node the higher rank it has). Further improvements of such approach will categorize the semantic descriptions into approved and validated and those still being unassessed. Hopefully, this will lay ground for building community ontologies across different science domains. The model itself may be changed while the system is working.

6 Implementation Status

Currently the presented model is being implemented within the virtual laboratory supporting the scripting approach to representing application plans [6]. The application execution engine is already [21] operational and capable of running test application plans. Simple ontology models have been built; however, they still require user assessment in order to be improved.

With respect to the web application layer a prototype of the user interface was built and a screenshot is depicted in Fig. 5.

The interface is divided into three parts:

– *application management* - In this widget the user is able to browse the collaboration space in search for application plans of interest. The search is

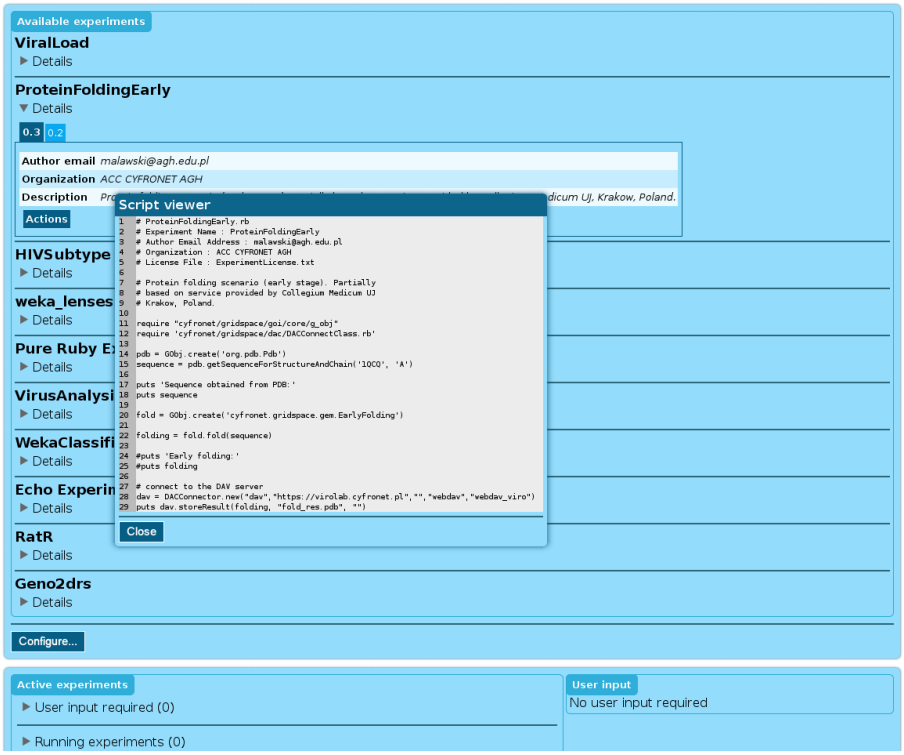

Fig. 5. Screenshot of a semantic collaboration space interface prototype.

supported by the *Metadata Engine*, so the application plans can be found according to the history of previous executions, produced results, owners or publications.

- *result management* - Results are managed by this view. Annotations and tags can be provided to assess particular results.
- *application execution status* - With this tab users may follow the execution status of their application plans and input intermediary data. The input is also supported by the *Metadata Engine* and previous results may be used as the inputs. When result type model is provided the engine suggests suitable inputs.

The overlapping window in the middle is displayed as popup and in this case is used to show the application plan script. Each application plan may be supplied with a license regarding its usage restrictions.

7 Conclusions and Future Work

This paper presents a semantic Web-based approach to constructing a scientific collaboration space. The solution combines social Web routines with the

formalisms of semantic content descriptions to facilitate the process of on-line research. Main improvements of the approach include integration of the application runtime system with result management and adoption of widely-used Web content management techniques in the area of scientific research.

At present the ViroLab virtual laboratory already integrates biomedical information related to viruses (proteins and mutations), patients (viral load) and literature (drug resistance); it enables to plan and run experiments transparently on distributed resources. Different experiments from the virology domain are executable, such as: from virus genotype to drug resistance interpretation, querying historical and provenance information about experiments, assisting a virologist with the Drug Resistance System, a simple data mining with classification. Further work will extend the list and explore re-usability in different science disciplines.

Future plans include the extension of the semantic model used for building the prototype and extending the user community to test and assess the approach. The aim is to benefit from the ideas brought by the Semantic Web trends and extend the present solutions in the area of community-driven research to make the process more reliable and efficient.

Acknowledgments. This work is partly funded by the European Commission under the ViroLab IST-027446 and the IST-2002-004265 Network of Excellence CoreGRID projects.

References

1. Chandrasekaran, B., Josephson, J.R., Benjamins, V.R.: What are ontologies, and why do we need them? IEEE Intelligent Systems **14**(1) (January/February 1999) 20–26
2. Cardoso, J.: The semantic web vision: Where are we? IEEE Intelligent Systems **22**(5) (September/October 2007) 84–88
3. Missier, P., Alper, P., Corcho, O., Dunlop, I., Goble, C.: Requirements and services for metadata management. IEEE Internet Computing **11**(5) (September/October 2007) 17–25
4. Carroll, J.J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K.: Jena: Implementing the semantic web recommendations. Technical report, HP Labs (2003)
5. Virolab Consortium: ViroLab - EU IST STREP Project 027446 (2008), `http://www.virolab.org`
6. ACC CYFRONET AGH: Virolab virtual laboratory (2008), `http://virolab.cyfronet.pl`
7. Sloot, P.M., Tirado-Ramos, A., Altintas, I., Bubak, M., Boucher, C.: From molecule to man: Decision support in individualized e-health (2006)
8. Gubala, T., Bubak, M.: Gridspace - semantic programming environment for the grid. LNCS 3911 (2006) 172–179
9. W3C: Owl web ontology language (2004), `http://www.w3.org/TR/owl-features`
10. W3C: Rdf: Resource description framework (2001), `http://www.w3.org/RDF`

11. Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., Fensel, D.: Web service modeling ontology. Applied Ontology **1**(1) (January 2005) 77–106
12. Facebook Team: A social utility that connects people with friends and others who work, study and live around them (2008), `http://www.facebook.com`
13. Yahoo! Inc: Photo sharing web space (2008), `http://www.flickr.com`
14. Fox, G.C., Guha, R., McMullen, D.F., Mustacoglu, A.F., Pierce, M.E., Topcu, A.E., Wild, D.J.: Web 2.0 for grids and e-science. In: INGRID 2007 - Instrumenting the Grid, 2nd International Workshop on Distributed Cooperative Laboratories - S.Margherita Ligure Portofino. (2007)
15. Goble, C., Roure, D.D.: Grid 3.0: Services, semantics and society. In: Proceedings of Cracow Grid Workshop 2007, ACC CYFRONET AGH (2008) 10–11
16. The University of Manchester and University of Southampton: myexperiment home page (2008), `http://www.myexperiment.org`
17. Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock6, M.R., Wipat, A., Li, P.: Taverna: a tool for the composition and enactment of bioinformatics workflows (2004)
18. Lee, S., Wang, T.D., Hashmi, N., Cummings, M.P.: Bio-steer: A semantic web workflow tool for grid computing in the life sciences (2007)
19. OASIS: Web services business process execution language (2007), `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel`
20. Google: Google web toolkit (2008) `http://code.google.com/webtoolkit`
21. Ciepiela, E., Kocot, J., Gubala, T., Malawski, M., Kasztelnik, M., Bubak, M.: Gridspace engine of the virolab virtual laboratory. In: Proceedings of Cracow Grid Workshop 2007, ACC CYFRONET AGH (2008) 53–58