UBioLab: a web-*LAB*oratory for *Ubi*quitous in-silico experiments

E. Bartocci[1,2,*], D. Cacciagrano[1], M. R. Di Berardini[1], E. Merelli[1], L. Vito[1]

[1]School of Science and Technology,
University of Camerino, Via Madonna delle Carceri 9, Camerino (MC), Italy

[2]Department of Computer Engineering, Faculty of Informatics, Vienna University of Technology, Treitlstrasse 3, 1040 Vienna, Austria

Summary

The huge and dynamic amount of bioinformatic resources (e.g., data and tools) available nowadays in Internet represents a big challenge for biologists –for what concerns their management and visualization– and for bioinformaticians –for what concerns the possibility of rapidly creating and executing in-silico experiments involving resources and activities spread over the WWW hyperspace. Any framework aiming at integrating such resources as in a physical laboratory has imperatively to tackle –and possibly to handle in a transparent and uniform way– aspects concerning physical distribution, semantic heterogeneity, co-existence of different computational paradigms and, as a consequence, of different invocation interfaces (i.e., OGSA for Grid nodes, SOAP for Web Services, Java RMI for Java objects, etc.). The framework *UBioLab* has been just designed and developed as a prototype following the above objective. Several architectural features –as those ones of being fully Web-based and of combining domain ontologies, Semantic Web and workflow techniques– give evidence of an effort in such a direction.
The integration of a semantic knowledge management system for distributed (bioinformatic) resources, a semantic-driven graphic environment for defining and monitoring ubiquitous workflows and an intelligent agent-based technology for their distributed execution allows *UBioLab* to be a semantic guide for bioinformaticians and biologists providing (i) a flexible environment for visualizing, organizing and inferring any (semantics and computational) "type" of domain knowledge (e.g., resources and activities, expressed in a declarative form), (ii) a powerful engine for defining and storing semantic-driven ubiquitous in-silico experiments on the domain hyperspace, as well as (iii) a transparent, automatic and distributed environment for correct experiment executions.

1 Introduction: state of the integration in Bioinformatics

The concept of information nowadays assumes several meanings, which are closely related to the notions of data, form, instruction, knowledge and representation. With the increasing popularity of computers, the following development of the network technologies and the appearance of the WWW, information is now available in multiple forms and it grows exponentially. Therefore, the way of helping the user to quickly, precisely and effectively get relevantly and up-to-dated useful information has quickly become in last ten years the ambitious goal for the industrial, official government and academic circles joined force.

[*]To whom correspondence should be addressed. Email: ezio.bartocci@tuwien.ac.at

A variety of information retrieval tools has been thus created by information providers, including search engines, information portals etc., which could help and assist users to filter, search for, organize, elaborate, represent related query information.

Although this scenario is quite general and characterizes several domains [1], it is particularly evident in Bioinformatics, where: (i) a huge amount of heterogeneous biological data is being generated distributedly at explosive rates, (ii) numerous computing methods and applications, often requiring expensive computational costs, are being developed daily. Further stimuli come from the Future Internet vision, a computing approach that is expected to be pervasive in our daily life (e.g., Personal Health Care System) and strongly based on complex procedures involving distributed domain knowledge and requiring timely analyses.

1.1 From data and software integration.....

Data and software integration has been and still remain in many cases the great challenge to face. We refer to [2, 3] for some literature about these issues. Some attempts to cope these problems have been proposed during the last years. An example of this trend is provided by *Gaggle* [4], a model-driven Service-oriented Architecture based on the concept of message passing where the term *touch-points* is used to mean common keys on which to join data. They integrate on a range of common or corresponding touchpoints: data values, names, identities, schema properties, keywords, loci, spatial-temporal points, etc. These touchpoints are the means by which integration is possible, and a great deal of a bioinformatician's work is the mapping of one touch-point to another. Another interesting reference is [5], an effort to think about (just-in-time) data integration at Web scale.

Service-oriented Architecture and its variants - as Open Grid Services Architecture (OGSA) [6] - are just some of several computational paradigms that have been singularly exploited and rarely combined in the attempt to realize *virtual laboratories*, e.g., frameworks for freely, easily and cooperatively integrating - as in a physical laboratory - distributed and heterogeneous bioinformatic resources. The variety of these frameworks - ranging from Grid computing to Web Services and, more recently, Cloud computing - is so consistent that any list could presumably be incomplete. Just to give an idea, a partial overview is proposed in the Discussion section. Descriptions are intentionally superficial, since a more detailed approach should mislead us from our real aim: taking note that many efforts have been done to integrate data and tool on the top of a fixed computational paradigm (e.g., architecture), but very few ones toward the interoperability of different architectures (which is one of the *UBioLab* requirements).

1.2 ....to computational paradigm interoperability: the role of the semantics

In [7], Goble and Stevens mention BioMoby [8], a tool for composing Web Services into workflows based on describing services and their input and output data types with controlled vocabularies. The founders of BioMoby seem to have drifted steadily towards the Semantic Web, but at one time they said that

[...] interoperability in the domain of Bioinformatics is, unexpectedly, largely a syntactic rather than a semantic problem.

That is to say, interoperability between Web Services can be largely achieved simply by specifying the data structures being passed between the services (syntax) even without rich specification of what those data structures mean (semantics).

Microformats get a brief mention as a means to enrich Web content with semantics –structured representation, links to supporting evidence or related data, provenance, etc. Goble and Stevens also cite the late great SIMILE Piggy Bank, a firefox extension allowing users to add their own semantic markup to Web pages– a key source of inspiration for Firegoose.

In [9] Stein assesses the state of computing in biology with the aim of "*the ability to create predictive, quantitative models of complex biological processes*". He defines *cyberinfrastructure* as consisting of data sources, computing resources, communication (not just networks, but syntactic and semantic connectivity as well), and human infrastructure (skills and cultural aspects).

The current biology cyberinfrastructure has a strong data infrastructure, a weak to non-existent computational grid, patchy syntactic and semantic connectivity, and a strengthening human infrastructure.

He seems to conclude that the Semantic Web [10] is promising, but not quite there yet. Almost echoing the quote above from [8], Stein says:

[...] in the short term, [...] we can gain many of the benefits of more sophisticated systems by leveraging the human capacity to make sense of noisy and contradictory information. Semantic integration will occur in the traditional way: many human eyes interpreting what they read and many human hands organizing and reorganizing the information.

Both of these papers suggest that the Semantic Web may finally solve these persistent integration problems, thus promoting a transition (from Web to Semantic Web), which will take place also in the Grid (e.g., Semantic Grid [11]) and more recently in the Cloud, where some projects from industry and academia –as TripCom [12], OpenKnowledge [13] and LarKC [14]– are beginning to include semantic technologies.

2 Ontologies, workflows and ubiquity: the mix of *UBioLab*

From an accurate analysis of the described scenario, two key elements are worth noting:

- On the one hand, a high degree of specialization w.r.t. a specific architecture (e.g., "type") characterizes the variety of integration frameworks proposed until now. Obviously, a "typed" approach allows any framework to be aware and to rightly handle a proper subsets of "typed" resources. However, it makes quite difficult to automatically extend the framework scope, e.g., to enable a semantically correct inclusion of new and differently "typed" resources at experiment design time, as well as correct invocations of such resources at experiment run-time.

- On the other hand, semantic methodologies and technologies have been already exploited to face the resource heterogeneity in term of "meaning", but have not been fully used to manage also their "type" heterogeneity.

In [15] the authors described briefly the vision of "knowledge in the Cloud" - which incorporates support for knowledge (semantic data), co-ordination (collaboration) and self-organization (internal optimisation) - and introduced two scenarios in which it enables the necessary collaboration of large scale and distributed knowledge.

This vision also permeates *UBioLab*, a Web-based framework which aims to be a virtual laboratory for easily managing distributed and heterogeneous (in terms of semantics and "type") domain knowledge (e.g., resources and activities, expressed in a declarative form), as well as for designing/monitoring/executing in-silico experiments on the domain hyperspace as automatic, ubiquitous and semantic-driven workflows.

UBioLab inherits its software architecture from a previous domain-independent prototype [16]: two integrated Web-based components –a knowledge manager system (KMS) and a workflow management system (WMS) – pivoting on an ontology-based knowledge model. Such a model is the core of *UBioLab*: it allows to solve, in a transparent and automatic way, factors like physical distribution, semantic heterogeneity and co-existence of different "types".

This combination of domain ontologies, Semantic Web and workflow techniques allows *UBioLab* to realize a semantic guide for bioinformaticians and biologists providing:

- a flexible knowledge organization allowing for a correct inference of the resource meaning and "type";
- a semantic-driven workflow formulation, realized by a graphical component for the assembly of (semantically) well-formed ubiquitous workflows from (semantically) heterogeneous and distributed resources;
- a transparent, automatic and distributed execution of workflows, thanks to an agent-oriented layer implementing a Migrating Workflow Model [17].

3 The *UBioLab* implementation

In the following, the *UBioLab* software architecture will be outlined describing individually the main *UBioLab* components - i.e., the KMS and the WMS - and highlighting how they globally interact.

3.1 The knowledge model

The glue of the *UBioLab* components is a well-defined multi-layered knowledge model for the semantic annotation of *resources* and *activities*, with the special purpose of contextualizing them in a given *domain* and linking each generic resource to the corresponding computational paradigm "type", so that enabling automatic and correct invocations of resource individuals at run-time.

Figure 1 shows the three layer-structure of the model, which takes inspiration from that one proposed in [18]:

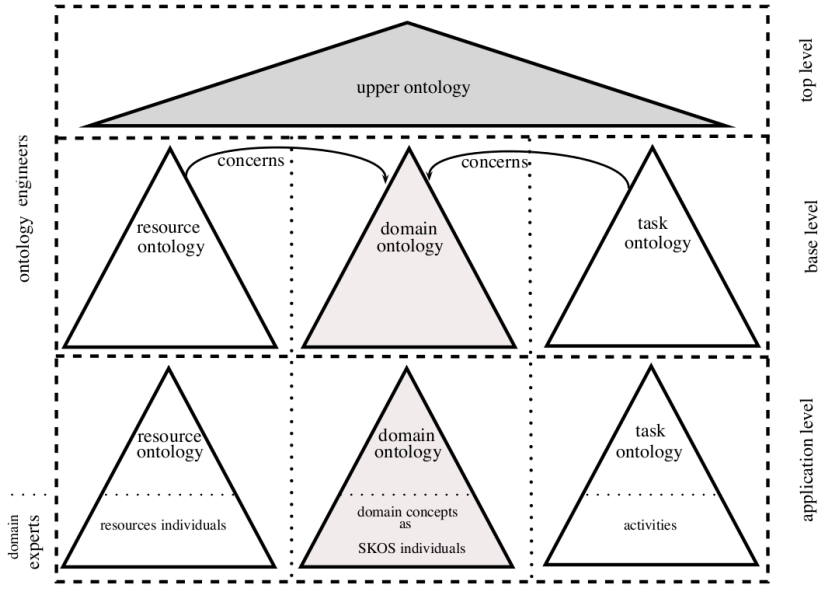

Figure 1: The knowledge model.

- *Top knowledge level:* It is formed by an Upper Ontology describing very general and domain-independent concepts shared across a large number of ontologies. The choice of the Upper Ontology concepts depends on what and how the knowledge is going to be described.

- *Base knowledge level:* It describes a specific vocabulary by specializing the terms introduced in the Upper Ontology w.r.t. a particular domain of interest.

- *Application knowledge level:* It introduces individuals (concept instances) and more specific concepts than those ones conceptualized in the Base level.

The main innovation of the underlining knowledge model is the partition of Base and Application levels in three different ontologies - *Domain*, *Resource* and *Task Ontologies* - each of which captures and models respectively domain, resource and operational aspects. In particular, Domain and Resource Ontologies follow by an "orthogonal" splitting of the Domain Ontology notion proposed in [18]. This "orthogonality" property is realized by a "*concerns*" relation, which permits to customize, in a very flexible way, the knowledge space w.r.t. a given domain and different (computational paradigm) "types", conceptualized in a linked Domain Ontology, so allowing us to rightly infer both the context (domain) and the right invocation interface ("type") for any resource individuals and, as a consequence, enabling automatic and correct resource invocations at run-time.

- *Domain Ontology*: It represents the semantic relationships between the concepts of a domain. It is implemented with a hybrid OWL-DL [19]/SKOS [20] semi-formal language, in order to provide more flexible and less formal description of concepts and metadata.

- *Resource Ontology*: It represents the kind of resources existing in the universe of a domain. It is an OWL-DL representation of a physical world, modeling the types of resources existing in the described domain.

- *Task Ontology*: It conceptualizes the operational knowledge, i.e. remote and local activities which can be invoked on and involve the resource space. The pivot of the Base level is the generic concept of *activity*.

  Abstract relations connect the Task Ontology to Domain and Resource Ontologies in order to link any activity to the context in which it works (relation "*concerns*" and its sub-relations), to the involved roles (relation "*hasRole*"), documents (relation "*hasDocument*") and objects (relation "*hasComplexInput*", "*hasComplexOutput*").

Taking into account this conceptual organization, new more specific resource concepts are typically conceptualized in the Resource Ontology Application level, connecting their more generic forms, already in the Resource Ontology, to existing domain concepts which they refer to; eventually, more specific domain concepts are inserted in the Domain Ontology Base or Application level.

Similarly, specific activities are conceptualized in the Task Ontology Application level keeping a forest structure, i.e. a tree structure for each activity where the child (hypoactivity) of a node activity (hyperactivity) is a more specific version of it. Constraints on some parameter values can be expressed in any form supported by Domain and Resource Ontologies, to determine the applicability of the activity with respect to the execution of its hyperactivity.

3.2 The knowledge management system

The KMS is a Web-based application that provides an intuitive user interface for the representation, visualization, integration, management and querying of domain and operational knowledge - conceptualized by the knowledge model already described - using Semantic Web technologies.

The Web-based approach differs *UBioLab* from others visualization tools –such as the plugin OWLViz [21] for Protégé [22]– since it enables a collaboration between different users, through the network, simply operating by mean of a Web browser.

3.2.1 Visualization of the knowledge model

Intuitive navigation is allowed by an effective Resource, Domain and Task Ontology visualization. The KMS interface also allows domain experts (in the role of Administrator, see below) to upload different (OWL-based) ontologies in a specific Web server directory, so that allowing *Ubiolab* to be parametric w.r.t. the knowledge domain.

On the Resource Ontology, the resources concerning a particular topic are connected by an arrow to that topic and it is shown only the subtree of the Domain Ontology, having concepts which are concerned by resources of the chosen types. By selecting a concept in the Resource Ontology, it is possible to visualize the resource instances (individuals) of the selected types.

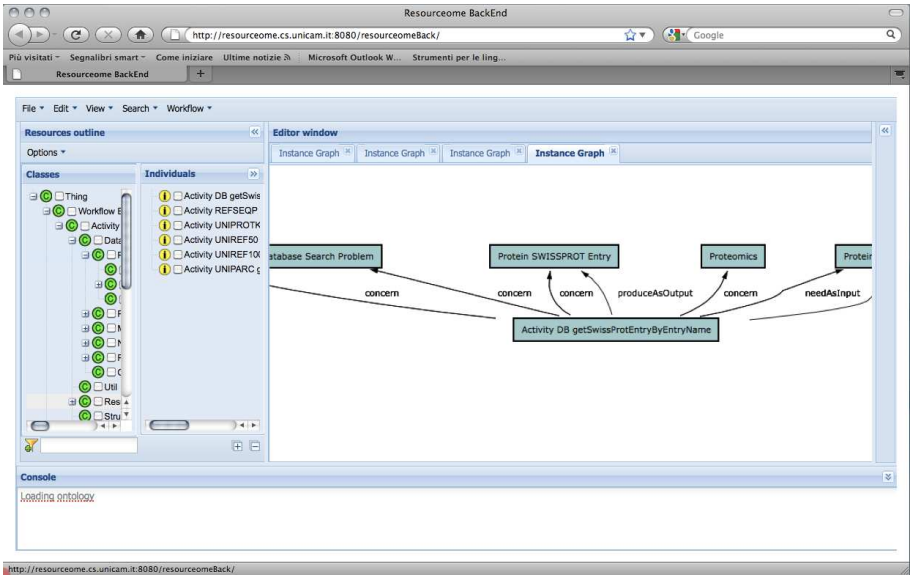

Figure 2: Browsing the knowledge model.

Once selected a particular resource instance, it is possible to see all its relationships with other resources, as well as its whole conceptualization in the Resource Ontology (see Figure 2).

Similarly, by right-clicking on a specific activity in the application level of the Task Ontology, it is possible to visualize the domain concept representing the context of the activity and any resource concepts representing the involved roles, objects and documents.

3.2.2 Management of the knowledge model

In the system, two kinds of interfaces are allowed, namely *Administrator* and *User*, being each one equipped with different privileges:

- *User*: Being a normal user not so familiar with ontologies and metadata schema, this interface permits to make modifications on Resource and Domain Ontologies only at Application level.

  More in detail, it is possible to easily add new (OWL-DL) resource individuals and relationships in the Resource Ontology; attributes and relationships required for the definition of a new resource individual can be visualized in a separate window. It is also possible to add new sub-concepts in the Domain Ontology as (blue-colored) SKOS concepts.

- *Administrator*: Assuming that an administrator has a greater experience in working with and developing ontologies than a normal user, this interface enables the access also to Top and Base levels of Domain and Resource Ontologies, as well as to any level of the Task Ontology. It is possible to deeply modify the Base level deleting, moving, adding and configuring new (red-colored) OWL-DL concepts. A SKOS parser [23] is available to translate SKOS user add-ins in term of OWL-DL concepts and relations to uniform and improve the Domain Ontology.

3.3 The workflow management system

The main actors involved in the WMS are:

- the knowledge model as an knowledge "active directory";
- a Web-based graphical interface for composing in-silico experiments, entering data, watching execution, displaying results;
- an archive to store experiment descriptions, results of executions and related traces;
- a scheduler able to invoke services included in the experiments at the appropriate time;
- a set of programming interfaces able to dialogue with remote activities;
- a set of visualization capabilities for displaying different types of results.

An in-silico experiment specification can be either translated into a workflow engine and/or stored as procedural knowledge in the knowledge model (see Figure 3). This is realized by an ontologization-compilation process involving three main components:

- The *graphical interface*: It enables the definition of in-silico experiments as (primitive and complex) activity workflows by a basic set of operators in the XML Process Definition Language (XPDL) [24][1], as well as the execution of existing or previous saved experiments, the monitoring of their execution state and the management of the produced results. The signature of workflow operator available in the WMS has been defined with the purpose to be a language-independent kernel[2]. As a consequence, any in-silico experiment specification can be also automatically conceptualized according to a corresponding XPDL-BPMN Ontology kernel and stored as procedural knowledge in the knowledge model (see Figure 3).

- *Hermes middleware*[3]: It provides the run-time environment for executing in-silico experiments as mobile and distributed code. In particular, it enables, transparently to users, the interaction with the external resources, i.e. invoked applications, and the migration of workflow executors to different sites.

- An *XPDL compiler* [28]: It is an Hermes special component which translates experiment specifications into interactive component-based specifications and generates the code to be executed on Hermes middleware. The associated workflow specification is the coordination model that describes how the generated agents cooperate to reach a particular goal.

[1] As a consequence, experiment specifications can be also edited by other applications compliant with XPDL standard.

[2] Moreover, the WMS architecture could allow different signatures to be uploaded, i.e. the WMS could support different specification languages, not only for defining workflows.

[3] Due to the lack of space, middleware architecture is not discussed here and we refer to [25–27] for further details.

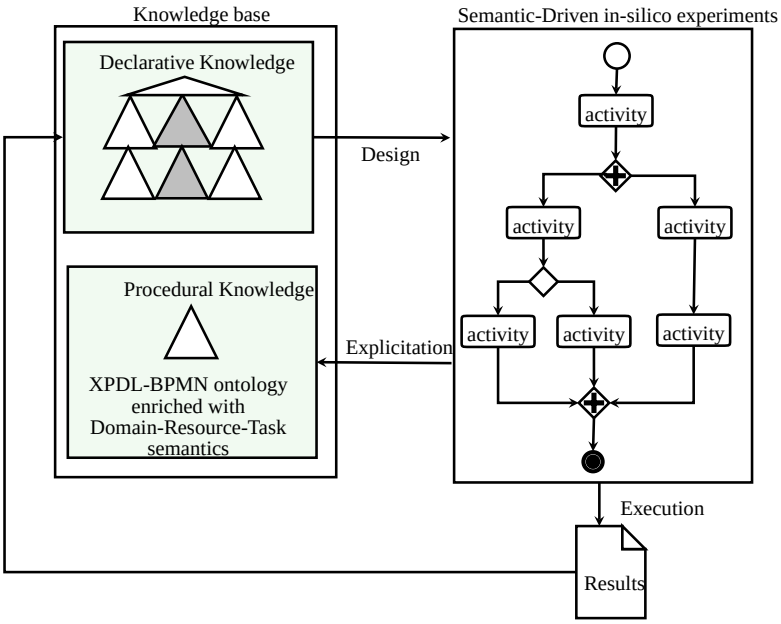

Figure 3: Knowledge life cycle.

As the KMS, also the WMS provides two interfaces –*User* and *Administrator*– enabling different privileges about definition, selection and execution of experiments related to specific goals.

In the first case, basic users can only select experiments from a finite list of goals, specify goal parameters driven by the KMS interface (i.e. navigating in a control way on Domain and Resource Ontologies, as well as adding new OWL-DL resource individuals and relationships and new SKOS concepts respectively on the Resource and Domain Ontology Application level) and visualize the obtained results.

In the latter case, the administrator can edit experiments, selecting the appropriate and involved domain, resource and activity concepts (and eventually accessing any knowledge model level), associate specific goals and store them.

Workflow exceptions are managed at two (cooperating) levels: either at the editing level –where the semantic layer naturally allows exceptions to be handled as explicit and user– defined workflow activities –or at the Hermes level– where a special agent is devoted to handle exceptions in according to different behaviors (invocation of equivalent activities, activity stop/pause/resume etc.).

4 Results

In this section, we provide an example of a process data retrieval in *UBioLab*. The goal is to obtain and to visualize all crystallographic structures related to a swissprot protein identificator. The corresponding workflow is formally described as a BPMN diagram in Figure 4.

In order to accomplish the proposed goal by the WMS, we need to infer all the activities that belong to the class *Database_Retrieval* concerning the concept *Protein*. Figure 5 shows the se-

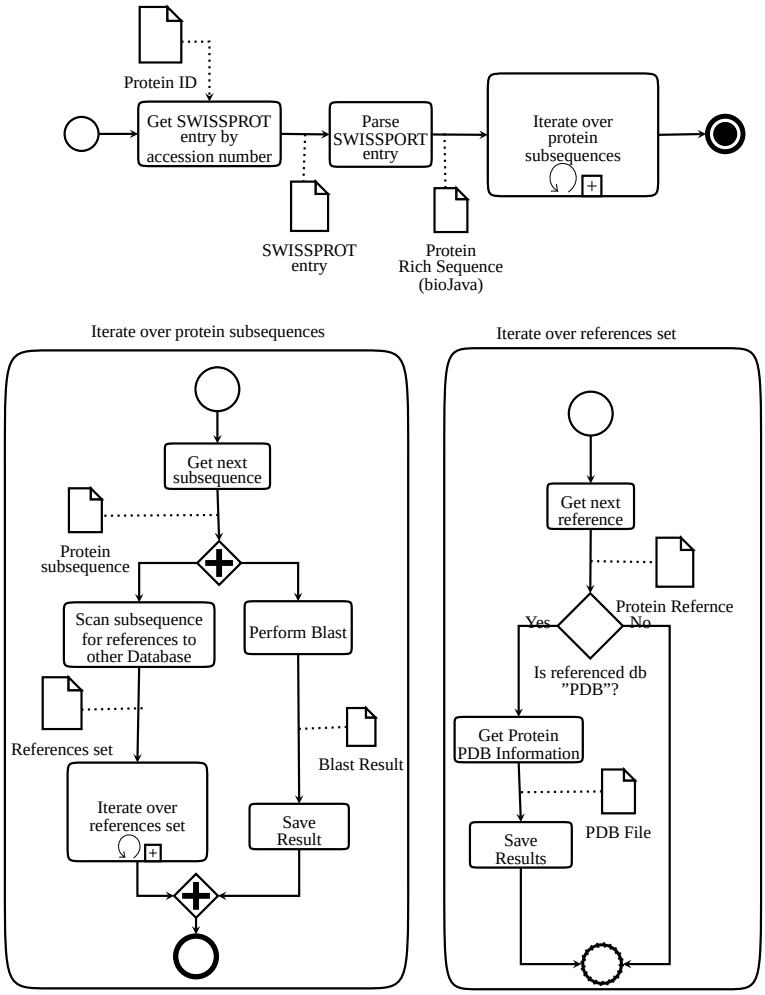

Figure 4: BPMN representation of the workflow.

mantic search: the used syntax is a triple of the form $\langle Concept, Relation, Target \rangle$, where *Concept=Database_Retrieval*, *Relation=concern* and *Target=Protein*. After choosing the proper activity, we can drag and drop it directly in an activity element of the workflow editor and to configure it with several parameters.

We can specify the order of the activities execution using special control-flow patterns:

- The *Sequence* pattern allows an activity to be executed after another: it is generally used when the output of an activity must be piped as input of the subsequent.
- The *If* pattern defines a conditional routing where the choice of the activity to be executed is case-driven: an error or exception can be caught and considered as a special case, so that the workflow becomes fault tolerant and the execution can select an alternative path when something goes wrong.
- The *Iteration* is a pattern enabling the cyclic execution of one or more activities: when a special case occurs, the control-flow leaves the cycle and the workflow execution continues.
- The *Concurrence* pattern enables the parallel execution of two or more activities.

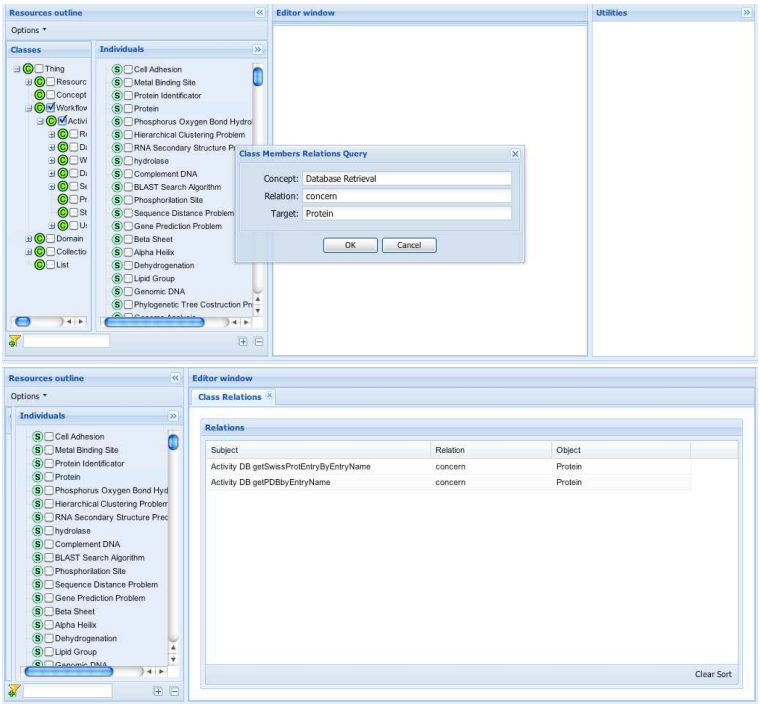

Figure 5: Semantic search of a *Database_Retrieval* activity concerning *Protein*.

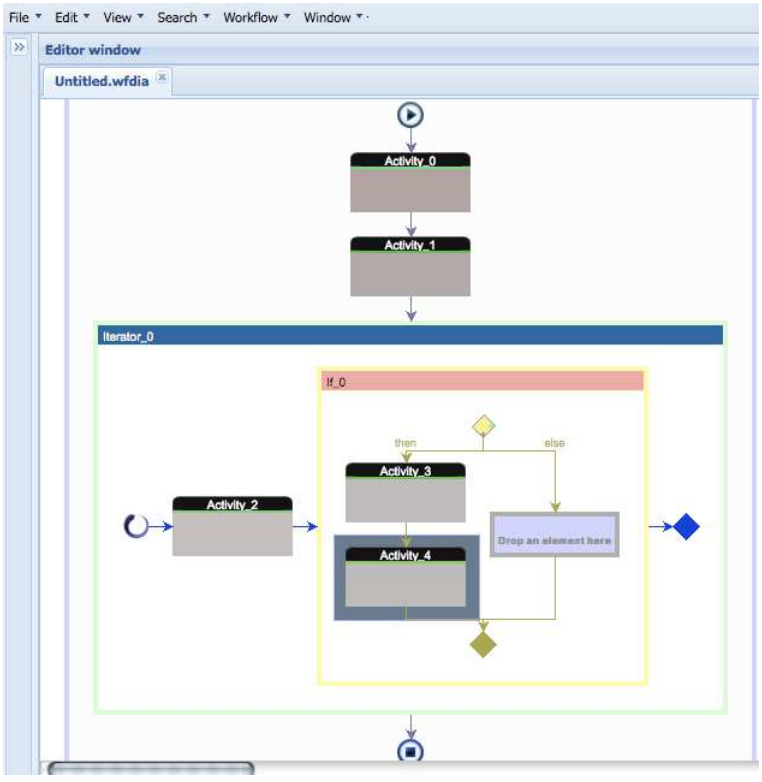

Figure 6: An in-silico experiment in *UBioLab*.

The whole workflow formalized in Figure 4 is obtained (in a very simple way) by the WMS graphic editor, as shown in Figure 6. The SWISSPROT entry file obtained by an *Activity_0*

is piped as input of an *Activity_1*, that extracts from a database the SWISSPROT entry cross-references. An iteration control-flow (*Iteration_0*) allows to evaluate the cross-reference (*Activity_2*), to choose through a conditional control-flow (*If_0*) those that refer to crystallographic structures, to fetch them (*Activity_3*) from the protein data bank and to store them (*Activity_4*) as results.

UBioLab also provides several plugins to visualize in a proper way the results: an example is the integration of the Jmol applet that, as Figure 7 illustrates, is used in this case for the 3D visualization of the fetched crystallographic structures.

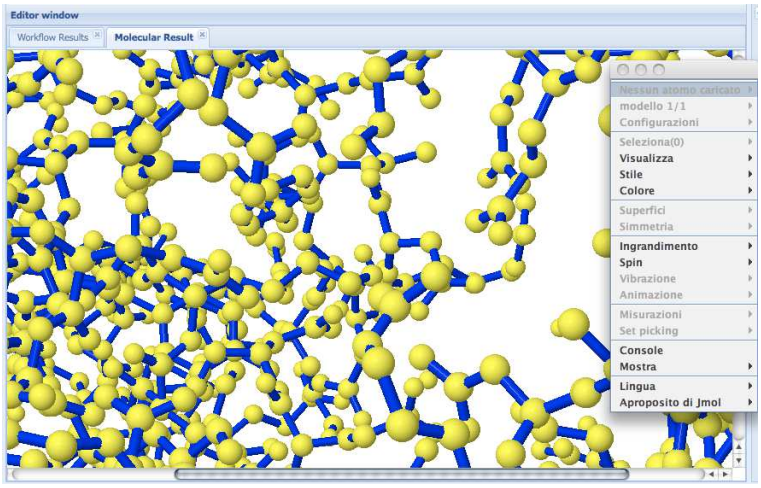

Figure 7: Visualization of a PDB file.

During the execution of the workflow, it is possible to monitor the obtained results and/or to interact with it whenever the in-silico experiment requires a conditional input from the user at run-time (Figure 8).

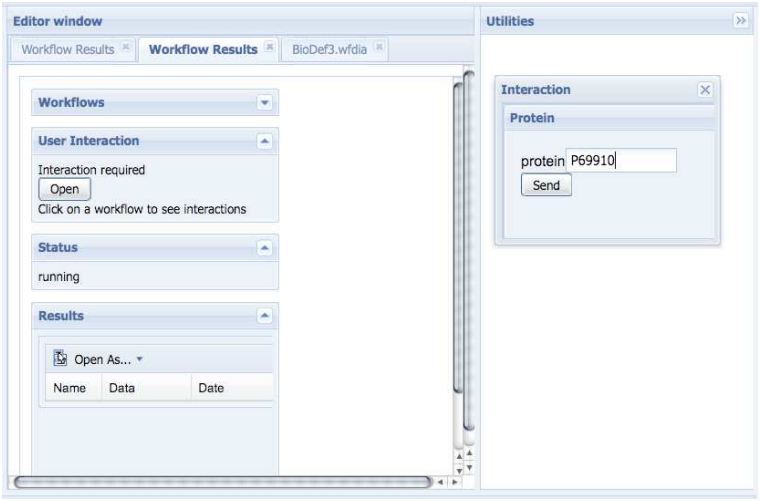

Figure 8: Input during the workflow execution.

Once the workflow is defined and tested by a bioinformatician, it is possible to publish it and to make it available to a biologist that can use it, without explicitly knowing its implementation.

5 Discussion

Existing frameworks for data and software integration are usually customized for either domain specific guided information retrieval (see Entrez [29] and SRS [30] system) or well-established mashup enhanced with specific invocation interfaces (i.e., OGSA for GRID nodes, SOAP for Web Services, Java RMI for Java objects, etc.). Special software frameworks (see BridgeDB [31]) then provide the necessary identifier mapping service for the same biological entities (gene, protein, etc.) among these heterogeneous information systems.

A popular open source software for building Grid systems and applications in the biological domain is the Globus Toolkit [32]. It provides a reference implementation of key Grid protocols. In order to obtain an "off-the-shelf" working system, it is, however, necessary to combine components from the Globus Toolkit into a higher level middleware system (e.g., gLite [33]).

The convergence of Grid with Web Services is embodied in the U.K. e-Science myGRID project [34], with its bioinformatics dedicated WMS Taverna [35].

Some Grid-based virtual laboratories have been already developed to support the user in the complexity of the Grid scenario. GeneGrid [36] aims to provide a platform for scientists, especially biologists, to access collective skills, experiences, and results in a secure, reliable, and scalable manner. The core architecture is database-managed, workflow-driven, application-oriented, and based on the OGSA model. The framework is purely service-based, providing the functionality through a number of cooperating OGSA-compliant Grid Services interacting in a predefined order.

GridFlow [37], unlike GeneGrid, is agent-based, developed for global Grid resource management using resource advertisement and discovery capabilities. An agent-based resource management system for Grid computing, ARMS [38], is also integrated with a local Grid resource scheduling system, Titan [39]. The functionalities of both ARMS and Titan are based on application performance prediction capabilities provided by the PACE system [40]. GridFlow enables users to construct Grid workflows and access Grid Services by mean of a Web portal. However, semantics is not supported and no simulation tool is available.

Recently, the paradigm of data or computing in the Cloud are becoming more and more prominent as an alternative to fully integrated approaches. Companies like Amazon with its Simple Storage Service S3 [41] and the Elastic Compute Cloud EC2 [42] or GigaSpaces Technologies Ltd. with its space-based computing platform XAP [43] explore the concept of Cloud computing for the realization of scalable and fault-tolerant applications.

GigaSpaces released a Cloud application server that enables their platform on top of the EC2 framework. The two companies argue that this combination enables an on-demand model of Cloud computing with a usage-based pricing model. Moreover, it provides Cloud portability, on-demand scalability of the Cloud, self-healing, and high-performance.

Loosely coupled solutions to storage integration were also recognized as important by the database community. [44] propose an integration framework that does not rely on a priori semantic integration, but rather on the co-existence of data sources. Such a system –called *dataspaces*– delivers a layer of core functionalities on top of the actual data providers that only virtually exposes a data integration platform to applications and users. This allows applications to focus on their own functionality rather than on the challenges of data integrity and efficiency

of integration. Furthermore, by keeping the individual data sources on distinctive machines conserves the advantages of distributed systems: no centralized server (thus avoiding bottlenecks with respect to performance and data access), robustness, and scalability of data volumes and the number of users.

For an exhaustive evaluation of the pros and cons of the frameworks mentioned in this paper and a detailed description of their main features we refer the reader to the following surveys [45–48].

Ontologies have a broad range of applicability in Bioinformatics - such as classification of medical concepts and data, database integration, collaboration between different groups, etc. - and have been already exploited to face the complexity of biological resources by several Knowledge Management Systems (KMSs) –like iTools [49], BioNavigation [50] and Bio-STEER [51]– as well as by different Workflow Management Systems (WMSs) - like Taverna, Remora [52], Kepler [53] and MS-Analyzer [54].

iTools aims at the classification and integration of the resources developed at the seven US National Center for Biomedical Computing. It is characterized by a taxonomy-like user friendly interface for browsing the managed resources (tools, in this case). The metadata representation model is a simple resource taxonomy: no semantic relationship (Object Property) between resources is managed, but only the properties (Data Property) of the visualized individuals. Interfacing (XML, SOAP and WSDL) external softwares are delegated for the updating of resource metadata.

BioNavigation is worth noting for its metadata representation philosophy, which relies on a physical graph representing the available resources and a conceptual graph of the domain. However, the mapping between the two graphs is not formalized by no relation and the knowledge representation does not allow one to manage individuals.

Bio-STEER is a Semantic Web-enabled computing environment where bioinformatics grid services are Semantic Web Services described in OWL-S. A graphical user interface guides the user in the design of a scientific workflow where the services are semantically sound; that is, the output of a service is semantically compatible with the input of the connecting service.

Taverna - a part of MyGrid project - has mainly the aim to integrate Web Services by workflows specified in a choreography language: XML Simple conceptual unified flow language (XScufl [35]). Embedded with its engine in a Java stand-alone application, it has been recently equipped with plugins that allow the user to access BioMoby.

In a similar way, in Remora a workflow is constructed visually from Moby Web Services. Using the semantic description, Remora implements a type-safe mechanism in order to guarantee data type compatibility among the output and the input of the connected services.

Kepler is a workflow tool based on an extension of the MoML language [55]: it is obtained by introducing the concept of a Director to define execution models and monitor workflows, where Web and Grid Services, Globus Grid jobs and GridFTP can be used as components.

Finally, MS-Analyzer is a software platform for realizing semantic-driven bioinformatic experiments on a very specific domain (Proteomics): it allows the integrated preprocessing, management and data mining analysis of proteomic data and it provides various services implementing spectra management and preprocessing. In particular, the composition and execution of such services is carried out through an ontology-based workflow editor and scheduler that

uses specific domain ontologies, namely WekaOntology and ProtOntology[4], which are strongly customized and oriented to conceptualize the specific domain and its main resources.

What it is worth highlighting in *UBioLab* is not the use of ontologies, but how ontologies are exploited in term of integration:

- *(Integration of domain knowledge)* The KMS can support any (OWL) domain conceptualization, overcoming most existing semantic and "typed" KMSs.

  Such a flexibility is a consequence of keeping, at the same time, a physical separation and a logical interoperation among *domain*, *resource* and *activity* concepts: the ontologized knowledge space is partitioned in three ontologies (Domain, Resource and Task Ontologies) and appropriate relationships among their respective (most general) concepts have been defined in order to keep information for each activity about its execution context, the roles (e.g., the actor types) that perform it, any involved resource equipped with its (inferred) "type", a possible implementation code, its preconditions and effects.

- *(Integration of declarative and procedural knowledge)* Similarly to BioWMS [57], the WMS permits to define any in-silico experiment specification as an activity workflow and to translate it into mobile code. However, it turns to overcome BioWMS and other traditional WMSs thanks to a strong integration of domain and operational aspects, which (i) enables a fully semantic-driven mechanism for realizing semantically correct in-silico experiments, (ii) permits to naturally capture in (and connect with) any in-silico experiment not only the associated experimental method but also its relative constraints and goals, (iii) allows any in-silico experiment to be not only executed but also stored as procedural knowledge in the knowledge model.

6 Conclusion

In this paper we have described and exploited *UBioLab* - a Web-based framework for easily managing distributed and heterogeneous (in terms of semantics and "type") domain and operational knowledge, as well as for designing/monitoring/executing in-silico experiments on the resource hyperspace as automatic, ubiquitous and semantic-driven workflows.

The possibility to store experiments as procedural knowledge, already present in the current prototype version of *UBioLab*, allows us to think about further capabilities for biological process data analysis. In fact, workflow technology is much more suitable for process data analysis than computational system biology simulation. Workflow instances, once conceptualized, do not only contain simple types, but identify semantic objects as well.

As a consequence, they could potentially allow information systems to exchange run-time information. Moreover, providing means to integrate workflow instances in a unified model, with a formal semantics and unambiguously identified objects, would enable process mining across the execution traces of multiple information systems. Our future efforts to improve *UBioLab*

[4]WekaOntology is an ontology of the data mining domain that is used to describe the tools of the Weka suite [56] and has been enriched by the description of relevant datasets and preprocessing algorithms. ProtOntology models concepts, methods, algorithms, tools and databases relevant to the proteomic domain, and provides a biological background to the data mining analysis.

are so oriented in this direction, possibly exploiting the same agent-based technology [58] used for workflow enactment.

Availability and requirements

Project name: UBioLab

Project home page: `http://cosy.cs.unicam.it/ubiolab`

UBioLab User's Guide:
`http://resourceome.cs.unicam.it/resourceomewordpress/downloads/ubiolab.pdf`

A prototype of UBioLab is available at:
`http://resourceome.cs.unicam.it/eyeOS/ Username:demo Password:demo`

Source code availability:
`http://resourceome.cs.unicam.it/resourceomewordpress/downloads/resourceome_sources+libraries.zip`,
`http://clarkparsia.com/pellet/download`, `http://www.graphviz.org/Download.php`.

Tutorial: `http://resourceome.cs.unicam.it/tutorial/`

Operating system(s): Platform independent

Language(s) and Framework(s): Java, GWT, Dot, OWL-DL, OWLAPI, SKOS

Other requirements: Web browser: Firefox 3.5 or higher, Safari 3, Google Chrome

License: Open source

Any restrictions to use by non-academics: none

List of abbreviations used

KMS: Knowledge Management System.

WMS: Workflow Management System.

OWL-DL: Web Ontology Language - Description Logic.

SKOS: Simple Knowledge Organization System.

XPDL: XML Process Definition Language.

BPMN: Business Process Modeling Notation.

Acknowledgements and Funding

This work is supported by the Investment Funds for Basic Research (FIRB) project Laboratory of Interdisciplinary Technologies in Bioinformatics (LITBIO). A special thank to the students Victor Karmansky, Nicola Paoletti and Fabio Alessandrelli that gave an important contribution in the development of *UBioLab*.

References

[1] E. Bartocci, E. Merelli and L. Mariani. An XML view of the "world". In *Proceedings of the 5th International Conference on Enterprise Information Systems, ICEIS 2003*, pages 19–27. Angers, France, 2003.

[2] L. Stein. Creating a bioinformatics nation. *Nature*, 417(6885):119–120, 2002.

[3] L. Stein. Integrating biological databases. *Nature Reviews Genetics*, 4(5):337–345, 2003.

[4] P. Shannon, D. Reiss, R. Bonneau and N. Balinga. The gaggle: An open-source software system for integrating bioinformatics and data sources. *BMC Bioinformatics*, 7:176, 2006.

[5] A. Halevy, M. Franklin and D. Maier. Principles of dataspace systems. In *PODS '06: Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–9. ACM, New York, NY, USA, 2006.

[6] I. Foster, C. Kesselman, J. Nick and S. Tuecke. The physiology of the Grid: An Open Grid Services Architecture for distributed systems integration, 2002.

[7] C. Goble and R. Stevens. State of the nation in data integration for bioinformatics. *J. of Biomedical Informatics*, 41:687–693, 2008.

[8] T. B. Consortium. Interoperability with Moby 1.0–it's better than sharing your toothbrush! *Brief. Bioinform.*, 9(3):220–231, 2008.

[9] L. Stein. Towards a cyberinfrastructure for the biological sciences: progress, visions and challenges. *Nature Reviews Genetics*, 9(9):678–688, 2008.

[10] T. Berners-Lee, J. Hendler and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[11] D. De Roure, N. R. Jennings and N. R. Shadbolt. *The Semantic Grid: A Future e-Science Infrastructure*, pages 437–470. John Wiley & Sons, Ltd, 2003.

[12] The TripCom project, `http://www.tripcom.org`.

[13] The OpenKnowledge project, `http://www.openk.org`.

[14] The LarKC project, `http://www.lark.eu`.

[15] D. Cerri, E. Della Valle, D. de Francisco Marcos et al. Towards knowledge in the cloud. volume 5333 of *Lecture Notes in Computer Science*, pages 986–995. Springer, 2008.

[16] D. Cacciagrano, F. Corradini, E. Merelli, L. Vito and G. Romiti. Resourceome: a multi-level model and a Semantic Web tool for managing domain and operational knowledge. In P. Dini, J. Hendler and J. Noll (editors), *The Third International Conference on Advances in Semantic Processing (SEMAPRO 2009)*, pages 38 – 43. IEEE Computer Society, 2009.

[17] A. Cichocki. *Migrating workflows and their transactional properties*. Ph.D. thesis, University of Houston, 1999.

[18] N. Guarino. *Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy*. IOS Press, Amsterdam, The Netherlands, The Netherlands, 1998.

[19] OWL Web Ontology Language, `http://www.w3.org/TR/owl-guide`.

[20] SKOS Simple Knowledge Organization System, `http://www.w3.org/TR/2009/REC-skos-reference-20090818`.

[21] OWLViz, `http://www.co-ode.org/downloads/owlviz`.

[22] The Protégé Ontology Editor and Knowledge Acquisition, `http://protege.stanford.edu`.

[23] SKOS parser, `http://oaei.ontologymatching.org/`.

[24] XML Process Definition Language, `http://xml.coverpages.org/XPDL20010522.pdf`.

[25] F. Corradini and E. Merelli. Hermes: Agent-Based Middleware for Mobile Computing. In *SFM*, pages 234–270. 2005.

[26] E. Bartocci, F. Corradini and E. Merelli. Enacting proactive workflows engine in e-science. In *Computational Science - ICCS 2006*, volume 3993 of *Lecture Notes in Computer Science*, pages 1012–1015. Springer Berlin / Heidelberg, 2006.

[27] E. Bartocci, F. Corradini, E. Merelli and L. Vito. Model driven design and implementation of activity-based applications in hermes. In *WOA*, volume 204 of *CEUR Workshop Proceedings*, pages 25–31. CEUR-WS.org, 2006.

[28] E. Bartocci, F. Corradini and E. Merelli. Building a multiagent system from a user workflow specification. In *WOA*, volume 204 of *CEUR Workshop Proceedings*, pages 96–103. CEUR-WS.org, 2006.

[29] G. Gibney and A. D. Baxevanis. Searching NCBI Databases Using Entrez. *Curr. Protoc. Hum. Genet.*, 71:6.10.1–6.10.24, 2011.

[30] T. Etzold, A. Ulyanov and P. Argos. SRS: information retrieval system for molecular biology data banks. *Meth. Enzymol.*, 266:114–128, 1996.

[31] M. P. van Iersel, A. R. Pico, T. Kelder, J. Gao, I. Ho, K. Hanspers, B. R. Conklin and C. T. Evelo. The BridgeDb framework: standardized access to gene, protein and metabolite identifier mapping services. *BMC Bioinformatics*, 11:5, 2010.

[32] Globus Toolkit, `http://www.globus.org/toolkit`.

[33] gLite - lightweight middleware for Grid computing, http://glite.web.cern.ch/glite/.

[34] R. D. Stevens, A. J. Robinson and C. A. Goble. myGrid: personalised bioinformatics on the information Grid. *Bioinformatics*, 19(suppl_1):302–304, 2003.

[35] T. Oinn, M. Addis, J. Ferris et al. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, 2004.

[36] P. V. Jithesh, N. Kelly, P. Donachy, T. J. Harmer, H. Perrott, M. McCurley, M. Townsley, J. Johnston and S. McKee. GeneGrid: Grid based solution for bioinformatics application integration and experiment execution. In *CBMS*, pages 523–528. 2005.

[37] J. Cao, S. A. Jarvis, S. Saini and G. R. Nudd. Gridflow: Workflow management for Grid computing. In *CCGRID*, pages 198–205. 2003.

[38] J. Cao, S. A. Jarvis, S. Saini, D. J. Kerbyson and G. R. Nudd. Arms: An agent-based resource management system for Grid computing. *Scientific Programming*, 10(2):135 – 148, 2002.

[39] D. P. Spooner, J. Cao, J. D. Turner, H. N. Lim Choi Keung, S. A. Jarvis and G. R. Nudd. Localised workload management using performance prediction and QoS contracts. In *Eighteenth Annual UK Performance Engineering Workshop (UKPEW' 2002), University of Glasgow, UK*. 2002.

[40] G. R. Nudd, D. J. Kerbyson, E. Papaefstathiou, S. C. Perry, J. S. Harper and D. V. Wilcox. PACE — A toolset for the performance prediction of parallel and distributed systems. *The International Journal of High Performance Computing Applications*, 14(3):228–251, 2000.

[41] Simple Storage Service S34, http://aws.amazon.com/s3.

[42] Elastic Compute Cloud EC25, http://aws.amazon.com/ec2.

[43] N. Shalom. The scalability revolution: From dead end to open road - an SBA concept paper, http://www.gigaspaces.com, 2007.

[44] M. Franklin, A. Halevy and D. Maier. From databases to dataspaces: a new abstraction for information management. *SIGMOD Rec.*, 34:27–33, 2005.

[45] A. Manconi and P. Rodriguez-Tomè. A survey on integrating data in bioinformatics. In M. Biba and F. Xhafa (editors), *Learning Structure and Schemas from Documents*, volume 375 of *Studies in Computational Intelligence*, pages 413–432. Springer Berlin / Heidelberg, 2011.

[46] Z. Zhang, V. B. Balic, J. Yu, K. Cheung and P. Townsend. Data integration in bioinformatics: Current efforts and challenges. In *Bioinformatics - Trends and Methodologies*, pages 41–56. Springer Berlin/Heidelberg, 2011.

[47] W. Jiang, M. Baumgarten, Q. Dai and Y. Zhou. The deployment and evaluation of a bioinformatics grid platform –the hust_bio_grid. *Computers & Electrical Engineering*, 38(1):19–34, 2012.

[48] P. Romano. Automation of in-silico data analysis processes through workflow management systems. *Briefings in Bioinformatics*, 9(1):57–68, 2008.

[49] I. Dinov, D. Rubin, W. Lorensen et al. iTools: a framework for classification, categorization and integration of computational biology resources. *PLoS ONE*, 3(5):2265, 2008.

[50] S. Cohen-Boulakia, S. Davidson, C. Froidevaux, Z. Lacroix and M. Vidal. Path-based systems to guide scientists in the maze of biological data sources. *J. Bioinformatics and Computational Biology*, 4(5):1069–1096, 2006.

[51] S. Lee, T. D. Wang, N. Hashmi and M. P. Cummings. Bio-steer: A semantic web workflow tool for grid computing in the life sciences. *Future Gener. Comput. Syst.*, 23:497–509, 2007.

[52] S. Carrere and J. Gouzy. REMORA: a pilot in the ocean of BioMoby web-services. *Bioinformatics*, 22(7):900–901, 2006.

[53] I. Altintas, C. Berkley and E. Jaeger. Kepler: an extensible system for design and execution of scientific workflows. In *Proceedings 16th International Conference on Scientific and Statistical Database Management: 21-23 June 2004; Santorini Island; Greece*. 2004.

[54] M. Cannataro, P. Guzzi, T. Mazza and P. Veltri. MS-Analyzer: Intelligent preprocessing, management, and data mining analysis of mass spectrometry data on the grid. *International Conference on Semantics, Knowledge and Grid*, 0, 2005.

[55] A. Lee and S. Neuendorffer. MoML - a modeling markup language in xml - version 0.4. Technical report, University of California at Berkeley, 2000.

[56] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2005.

[57] E. Bartocci, F. Corradini, E. Merelli and L. Scortichini. BioWMS: a web-based Workflow Management System for bioinformatics. *BMC Bioinformatics*, 8(Suppl 1):S2, 2007.

[58] E. Bartocci, D. Cacciagrano, N. Cannata, F. Corradini, E. Merelli, L. Milanesi and P. Romano. An agent-based multilayer architecture for bioinformatics grids. *NanoBioscience, IEEE Transactions on*, 6(2):142 –148, 2007.