31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

A semantically aware platform for the authoring and secure enactment of bioinformatics workflows

M. Tsiknakis[1], S. Sfakianakis[1], G. Zacharioudakis[1], L. Koumakis[1], A. Kanterakis[1], G. Potamias[1], D. Kafetzopoulos[2]

Abstract—**Recent advances in the field of bioinformatics present a number of challenges in the secure and efficient management and analysis of biological data resources. Workflow technologies aim to assist scientists and domain experts in the design of complex, long running, data and computing intensive experiments that involve many data processing and analysis tasks with the objective of generating new knowledge or formulate new hypothesis.**

In this paper we present a bioinformatics workflow authoring and execution environment that intends to greatly facilitate the whole lifecycle of such experiments. Emphasis is given on the security and ethical requirements of these scenarios and the corresponding technological response. In addition we present our semantic framework used for supporting specific user-requirements related to the reasoning and inference capabilities of the environment.

I. Introduction

During the last decades the scientific community in general and the biomedical community in specific experiences an increasing need for efficient data management and analysis tools. The problems introduced by the need for the reliable and secure management, processing, and understanding these large sets of biological data generally require an assortment of information technology techniques and methods.

Current state of the art technologies like the Service Oriented Architecture (Web Services), the Grid [1], and the Semantic Web [2], enable the biomedical informatics community in the integration of these resources for performing complex scientific experiments. Scientific workflows have been proposed as a mechanism for coordinating processes, tools, and people for scientific problem solving purposes and aim to support coarse-granularity, long-lived, complex, heterogeneous, scientific computations [3]–[4].

In this paper we present our work for supporting the design and enactment of discovery driven bioinformatics workflows in the context of the ACGT European integrated project [5]. The objective of the ACGT (Advancing Clinico-Genomic Trials on Cancer: Open Grid Services for improving Medical Knowledge Discovery, www.eu-acgt.org) is the development of a semantically rich infrastructure facilitating seamless and secure access and analysis, of multi-level clinical and genomic data enriched with high-performing knowledge discovery operations and services in support of multi-centric, post-genomic clinical trials. The project builds on open software framework based on Web services, (WS)-Resource Framework (WSRF) [6] and Open Grid Services Architecture (OGSA), the de facto standards in grid computing.

The objective of this paper is not to present all aspects of the work in ACGT, as this would require substantially more space. It rather focuses on the presentation of the technical issues addressed in designing a web-based workflow environment for the design, publication and trusted enactment of discovery-driven bioinformatics workflows. In addition the requirements for embedding intelligence in the workflow environment are discussed and design decisions are presented.

II. Requirements For The ACGT Workflow Environment

One of the most important constraints for the management of personal clinical and genomic data is the compliance to the ethical and legal data protection requirements. In ACGT a generic data protection framework has been defined which is based on a technical security infrastructure as well as on organizational measures and contractual obligations [7]. Most of these security requirements are dealt with the Grid infrastructure layer. In particular the Grid Security Infrastructure (GSI) [8] supports user authentication through digital signatures and also the delegation of user privileges to a service so that it can retrieve data or perform an action on the user's behalf and without the user's intervention.

In agreement with the authors in [9] the decision of using a standard workflow definition language in ACGT has been taken. BPEL [10] was chosen from the business process management world as the most prominent and well supported technology. Nevertheless the choice of BPEL gave rise to two more requirements. The first one is the provision of a user friendly workflow authoring environment. The second relates to the need for an infrastructure that would make possible the invocation of the ACGT secure grid services from inside the BPEL-based workflows, since BPEL and the Web Services standard security specifications do not deal with such requirements.

In addition to these requirements we also have specific user-requirements related to the reasoning and inference capabilities of the environment. Reasoning is required for matching the description of services requests coming from the end user applications to the contents of the service knowledge base. The way this matching query is performed

[1]Institute of Computer Science, Foundation for Research and Technology-Hellas (ICS-FORTH), Heraklion, Crete, Greece
{tsiknaki, ssfak, gzaxar, koumaki, kantale, potamias}@ics.forth.gr
[2]Institute of Molecular Biology and Biotechnology, Foundation for Research and Technology-Hellas (ICS-FORTH), Heraklion, Crete, Greece
{kafetzo@imbb.forth.gr}

to the semantic descriptions of the available services is guided by the foundational and domain specific ontologies. Inference is needed because these description logic based ontologies have specific "recipes" for extracting new knowledge. A simple example to make it clearer is the following: "Find me services that perform sequence alignment". The system should be able to deduce that there are different kinds of sequence alignment (pairwise/multiple, global/local, etc.) and therefore return matches from all of them.

III. SEMANTIC FRAMEWORK OF THE ACGT SERVICES AND TOOLS

Semantics provide "meaning" for "understanding" the entities and the processes in a domain of discourse. Therefore we need to clearly define the role of semantics descriptions of services and to prioritize the different use cases of them in order to provide some useful and practical solution. For these reasons we have selected the service discovery, selection, and "matchmaking" (i.e. composition) as the primary use cases where semantics descriptions for services fit in. All of these are advanced features of a modern problem solving environment such as the Workflow Editor and Enactment environment that the ACGT aims to deliver.

Based on a variety of discovery-driven use-cases studied, we have concluded that different kinds of semantic descriptions are required, with functional, informational and behavioural semantic descriptions been of particular importance. The functional descriptions provide semantics descriptions about the service capabilities and therefore are important for the discovery of services based on what they can do for the user. Also at the semantic level the informational descriptions should support the discovery and integration scenarios for web services since they provide information about the input and output messages of the services. Finally behavioral descriptions are an interesting class where especially the externally visible behaviour of the service can be used for automatically constructing parts of a workflow or "workflow templates".

For constructing the service semantics in ACGT we have chosen OWL-S [21] as the upper ontology. Nevertheless, in addition to OWL-S, there should be some domain specific ontology (or ontologies) that fill in the missing semantics. BioMOBY and myGrid ontologies **Error! Reference source not found.** provide such domain specific ontologies. In particular the BioMOBY Object ontology supplies a large set of bioinformatics data types and formats that can be used for annotating the service parameters. Also the myGrid Services ontology accommodates a hierarchy of service capabilities that is again bioinformatics specific.

The general architectural view of the ACGT semantic framework is shown in the figure below. It basically consists of the following components:

- The service registries and repositories. In ACGT this is the Metadata Repository but additional third party registries exist, such as the Biomoby ones. These are the primary sources of service descriptions and need not be implemented with the same technologies or contacted and searched with the same protocols.
- The "RDFizers" are components for exporting the service registries information in the schema defined by the foundational ontology

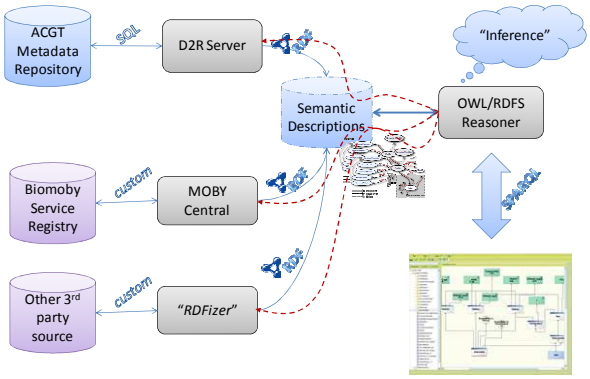

Figure 1: The Semantic Integration Framework

- The Reasoner is the component that performs the actual tasks of service discovery or matching by employing certain inference rules on the RDF data exported by the "RDFizers".

End user tools, like the workflow editor, or other services interact with the Reasoner in order to make the proper entailments and inferences and answer their queries.

IV. IMPLEMENTATION

In order to address the architectural and security requirements previously stated a number of tools, services, and components have been collectively designed, in addition to the tools and services of the semantic framework. We use the term "ACGT Workflow Environment" to refer to this infrastructure as a whole. This environment supports the whole workflow life cycle: from designing new workflows, to their enactment, provenance management, reusing and repurposing. It comprises the following entities:

- Workflow Editor: the graphical web-based tool that enables the design and editing of workflows in an easy and intuitive way
- Workflow Enactor: the BPEL compliant, third party orchestrator for the enactment of the workflows
- Proxy Services: the gateways and representatives of the real ACGT services that present a BPEL and Web Services conformant interface to the Enactor

These components are further described in more detail in the following sections.

A. *The Workflow Editing Environment*

The workflow editing environment, called Zeno, has been designed as a modern web application. It consists of the client side, which runs as a "rich internet application" (RIA) inside the user's web browser, and the server side, which is responsible for the core "business-logic". These two components communicate with each other in a unidirectional, "firewall friendly" way, employing the "AJAX" technique so that Zeno provides a desktop application's look and feel and interactivity.

Zeno follows a dataflow, instead of control flow, paradigm: its workflows are directed acyclic graphs (DAGs) where the links between the nodes of the graph represent data that are transmitted between the services and each service will start execution when it has data in all its input parameters. The actual workflow construction is done visually, by "dragging-and-dropping" and connecting, by drawing lines, the output and the inputs of analytical and data management services. For each processing step inside the workflow there is always metadata information about the input and output parameters and in the cases of a GridR data analysis script [11] or a mediator query the script or query code is shown as well.

During the construction of a workflow Zeno performs automatic validation of the data connection links. This is based on the metadata description of the input and outputs of the services. For the syntactic composition of services the metadata needed is mostly about the parameter data types. The Zeno also takes advantage of the semantic framework presented in the previous section, which is responsible for actually validating the connections and providing service discovery based on semantics

When a workflow is ready the user is allowed to enact it by specifying any required input parameters. In addition, a workflow that a user feels confident about its status and usefulness can be published so that also other people can use it. The publication process includes the supply of the necessary metadata (e.g. descriptions of inputs and outputs, functional classification, etc.) and the registration of the executable workflow in the ACGT service registry.

B. BPEL compliant interfaces to secure Bioinformatics services

The incompatibilities between the BPEL processes and the GSI secured ACGT services can be overcome by supplying the necessary "layer of indirection": the Proxy Services. These proxies or wrapper services provide BPEL friendly "facades" of the original, real ACGT services, effectively working as calls transformation bridges between the two worlds (Figure 2).

The proxy services mechanism can be also used to provide higher level abstractions. An example of this is to remodel the interface of the service in order to simplify it or provide through one gateway adapter service access to many services by dynamic creation of custom interfaces. In the case of the "GridR Proxy Service", by using the wrapping technique we were able to encapsulate the underlying "GridR Service" [11] and hide its technical details, but also to support the notion of "Scripts as Services", in which a different web service interface is exposed from the same single proxy service, depending on the actual R-script that is being "proxied".

In a similar manner the wrapping technique has been applied to provide secure access to WSRF OGSA-DAI resources [12] and simplify the connection to them by enclosing and hiding the technical hindrances of OGSA-DAI, like perform documents and stateful resources.

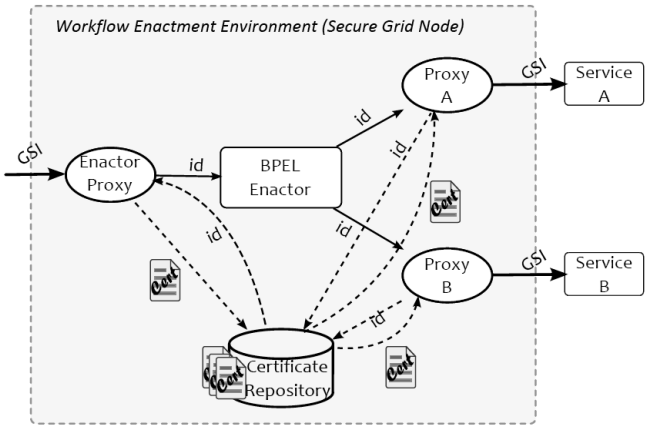

Figure 2: The architecture of the Workflow Enactment Environment

C. BPEL transformation and workflow execution

In order for the data flows created by the workflow editor to be "enacted" (i.e. executed) a transformation to an executable language should be performed. In our case the BPEL workflow enactor is actually used as a "virtual machine" for the execution of the high level scientific processes designed in the workflow editor but a transformation step to BPEL is required.

The mapping to BPEL and the creation of the BPEL descriptions of our workflow editors data flows are performed by the BPEL transformation tool which is a subcomponent of the workflow editors back end. The transformation tool leverages the flow construct of the BPEL to map the graph representation of the editor's workflows to an identical graph in the BPEL description. The use of the flow element has the additional benefit that at runtime, when the final BPEL process is deployed and enacted, independent activities are executed in parallel.

The data flows between the execution steps in the input description represent (data) dependencies and therefore the corresponding source and target links are introduced in the BPEL output. Since BPEL is more control-flow oriented, variables are introduced to represent the exchanged messages and the data connections between the activities in the input description are modelled as the assignments of "output" to "input" BPEL variables.

V. RELATED WORK

As an important scientific tool workflows have attracted a lot of attention for e-Science. A good survey and classification of scientific workflow engines and editors is presented in [13]. Especially in the bioinformatics area the Taverna [14] workbench has gained a lot of popularity as a free software tool for designing and executing workflows. Other tools like Triana [15] are also widely used with similar functionality.

In comparison to these existing tools our environment differs in the following main directions: the use of the Web infrastructure for supporting the whole workflow life cycle, from editing to enactment and reuse, the need for addressing real world requirements for authorization and privacy, and

the compliance to standard based solutions such as the use of BPEL as the workflow execution language.

To our knowledge fully web based workflow editors with desktop look and feel and similar features and behaviour, like "drag-and-drop" and so on, have not been developed so far. On the other hand the use of BPEL as an execution language for scientific workflows has already been explored in a number of publications [16]–[17]. The invocation from BPEL engines of stateful, WSRF-based Grid services was discussed in a number of occasions, e.g. [18]. The Proxy Services infrastructure we adopted has been also described in [19] for the invocation of Grid services. Proxy Services were once again "discovered" in [20] and proposed as a generic methodology for bridging BPEL engines with secure Grid services. In our work we provide additional rationale for such Proxy Services infrastructure in support of Grid credential delegation and we further refine the proposed architecture.

VI. DISCUSSION AND FUTURE WORK

The ACGT Workflow Environment is modelled as a "Software as a Service" type of application. It is centrally deployed and accessible though the Web. There are a couple of shortcomings to this approach, for example the network latencies that could be high for some users and severely affect interactivity and responsiveness. On the other hand there are important advantages following this approach. The Zeno as a web application is almost universally accessible, requires no installation, and can be upgraded at any time. Support for long running workflows in detached mode, i.e. without requiring the users' presence, is provided by default. Also the publication, sharing, and re-purposing of the workflows and building user communities in a "social web" ("web2.0") manner is much easier to support.

The "Proxy services" methodology makes scientific workflows with stringent security requirements compatible with business process management technologies and the experiences so far show that it is flexible enough to overcome many technical obstacles.

Finally, with respect to the Zeno editor, there's ongoing work for supporting control flow in a way that preserves most of its current declarative character, and additional ways to handle complex data types.

REFERENCES

[1] I. Foster, "The Grid: Computing without bounds," Scientific American, vol. 288, no. 4, pp. 60–67, 2003.

[2] N. Shadbolt, T. Berners-Lee, and W. Hall, "The Semantic Web Revisited," IEEE INTELLIGENT SYSTEMS, pp. 96–101, 2006.

[3] A. Belloum, E. Deelman, and Z. Zhao, "Scientific workflows," Scientific Programming, vol. 14, no. 3-4, p. 171, 2006.

[4] G. Fox and D. Gannon, "Special Issue: Workflow in Grid Systems: Editorials," Concurrency and Computation: Practice & Experience, vol. 18, no. 10, pp. 1009–1019, 2006.

[5] M. Tsiknakis, M. Brochhausen, J. Nabrzyski, J. Pucacki, S. Sfakianakis, G. Potamias, C. Desmedt, and D. Kafetzopoulos, "A Semantic Grid Infrastructure Enabling Integrated Access and Analysis of Multilevel Biomedical Data in Support of Postgenomic Clinical Trials on Cancer," Information Technology in Biomedicine, IEEE Transactions on, vol. 12, no. 2, pp. 205–217, 2008.

[6] K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, I. Sedukhin, D. Snelling, S. Tuecke, and W. Vambenepe, "The WS-Resource Framework Version 1.0," in Global Grid Forum, available at http://www.globus.org/wsrf, 2004. [Online]. Available: https://www-unix.globus.org/wsrf/specs/ws-wsrf.pdf

[7] B. Claerhout, N. Forgó, T. Krügel, M. Arning, and G. De Moor, "A Data Protection Framework for Transeuropean genetic research projects." Studies in health technology and informatics, vol. 141, p. 67, 2008.

[8] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke, "Security for Grid services," in proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing, 2003, pp. 48–57.

[9] A. Barker and J. van Hemert, "Scientific Workflow: A Survey and Research Directions," LECTURE NOTES IN COMPUTER SCIENCE, vol. 4967, p. 746, 2008.

[10] A. Arkin, S. Askary, B. Bloch, I. Francisco Curbera, B. Yaron Goland, N. Kartha, S. Commerce, and O. Alex Yiu, "Web Services Business Process Execution Language Version 2.0." [Online]. Available: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html

[11] D. Wegener, T. Sengstag, S. Sfakianakis, S. Ruping, and A. Assi, "GridR: An R-Based Grid-Enabled Tool for Data Analysis in ACGT Clinico-Genomics Trials," in e-Science and Grid Computing, IEEE International Conference on, 2007, pp. 228–235.

[12] M. Antonioletti, M. Atkinson, R. Baxter, A. Borley, N. Hong, B. Collins, N. Hardman, A. Hume, A. Knox, M. Jackson et al., "The design and implementation of Grid database services in OGSA-DAI," Concurrency and Computation: Practice & Experience, vol. 17, no. 2, pp. 357–376, 2005.

[13] J. Yu and R. Buyya, "A Taxonomy of Workflow Management Systems for Grid Computing," Journal of Grid Computing, vol. 3, no. 3, pp. 171–200, 2005.

[14] T. Oinn, M. Greenwood, M. Addis, M. Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, D. Marvin et al., "Taverna: lessons in creating a workflow environment for the life sciences," CONCURRENCY AND COMPUTATION, vol. 18, no. 10, p. 1067, 2006.

[15] I. Taylor, I. Wang, M. Shields, and S. Majithia, "Distributed computing with Triana on the Grid," Concurrency and Computation: Practice & Experience, vol. 17, no. 9, pp. 1197–1214, 2005.

[16] W. Emmerich, B. Butchart, L. Chen, B. Wassermann, and S. Price, "Grid Service Orchestration Using the Business Process Execution Language (BPEL)," Journal of Grid Computing, vol. 3, no. 3, pp. 283–304, 2005.

[17] A. Akram, D. Meredith, and R. Allan, "Evaluation of BPEL to Scientific Workflows," in Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on, vol. 1, 2006.

[18] F. Leymann, "Choreography for the Grid: towards fitting BPEL to the resource framework: Research Articles," Concurrency and Computation: Practice & Experience, vol. 18, no. 10, pp. 1201–1217, 2006.

[19] K. Chao, M. Younas, N. Griffiths, I. Awan, R. Anane, and C. Tsai, "Analysis of Grid Service Composition with BPEL4WS," in AINA' 04: Proceedings of the 18th International Conference on Advanced Information Networking and Applications Volume, vol. 2, 2004, p. 284.

[20] P. Amnuaykanjanasin and N. Nupairoj, "The BPEL orchestrating framework for secured grid services," in Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on, vol. 1, 2005.

[21] D. Martin, et al "Bringing semantics to web services: The OWL-S approach," Lecture Notes in Computer Science, Springer, vol. 3387, pp. 26-42, 2005

[22] K. Wolstencroft, P. Alper, D. Hull, C. Wroe, P.W. Lord, R.D. Stevens, and C.A. Goble "The myGrid ontology: bioinformatics service discovery," International Journal of Bioinformatics Research and Applications (IJBRA), 3(3), 303-325, 2007