Enacting proactive workflows engine in e-Science

Ezio Bartocci, Flavio Corradini and Emanuela Merelli

Dipartimento di Matematica e Informatica, Università di Camerino
Via Madonna delle Carceri 9 - 62032 Camerino, Italy
{ezio.bartocci, flavio.corradini, emanuela.merelli}@unicam.it

Abstract. The dynamic nature and the geographic distribution of scientific resources, require flexible, adaptive and fault tolerant computational environment where an in-silico experiment can be executed as a workflow of activities. In this paper we propose a software environment to dynamically generate domain-dependent workflow engines consisting of a proactive multiagent system -a distributed, concurrent system- generated from the workflow specifications. The proposed architecture has been implemented on Hermes, agent-based mobile computing middleware, and validate within "Oncology over Internet" project.

1 Introduction

Over the past few years, new high-throughput methods for data collection in life science, e.g. microarray processing, have greatly increased data generation. So as, the wide use of the Web has fostered the scientists' work –solving complex scientific problems and making new discoveries– to take more and more place within a project team that shares data sources and computational methods in a collaborative way. As consequence, the traditional scientific process has become computationally intensive and *in-silico* experiments -described as processes of several activities to test hypotheses, derive a summary and search for patterns [16]- are laboriously executed in a large, distributed and dynamic environment.

Moreover, the dynamic nature of scientific resources enhances further on the computational environment complexity. In fact, the execution of a *in-silico* experiment may simultaneously demand data integration from several application domains (e.g. biology, pharmacology, chemistry), tool integration -analysis techniques (e.g. data mining and text mining) computational methods- typically offered as services [15] and dynamically updated, added or removed.

Nowadays, e-Science – the use of advanced computing technologies to support scientist – seems to be the only way to face the complexity of the scientific computational environment.

We believe that the workflow technology together with an effective and efficient resource management system [2] could be a good start to face the complexity that surrounds scientist's work and then to help him in taking advantage of the huge amount of available resources.

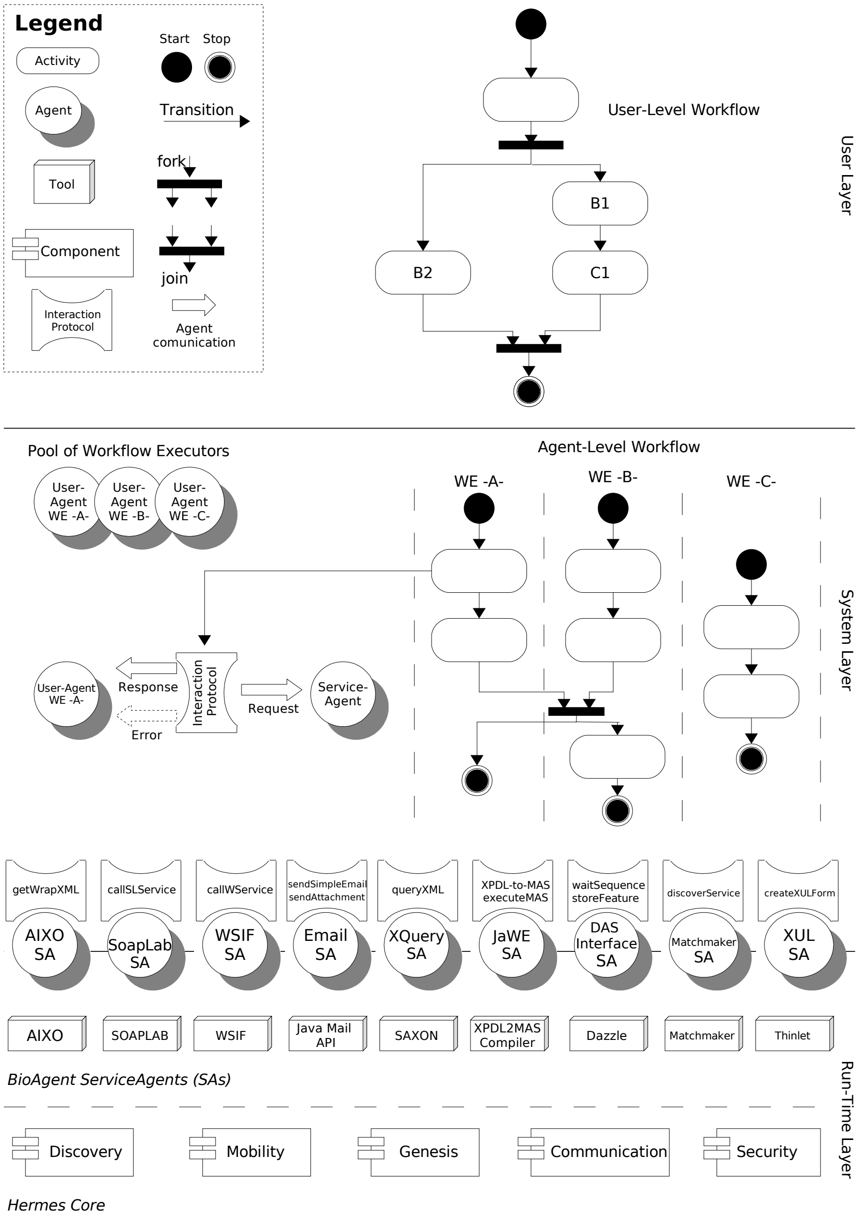


Fig. 1. BioAgent/Hermes architecture

In-silico experiments can be naturally specified as workflows of activities, that implement data analysis processes in standardized environments. Furthermore, the workflow owns the advantage to be reproducible, traceable and to reuse intermediate results; fundamental features to validate a scientific experiment. The software component that “defines, manages and executes workflows through the execution of software whose order of execution is driven by a computer representation of the workflow logic”, according to Workflow Management Coalition (WfMC) reference model [10], is named Workflow Management System (WMS).

In e-Science domain, several WMSs-like [12, 11, 14, 9, 18] have been already developed and adopted to support the daily work of a bioscientist. Taverna [12] -a part of MyGrid project [17]- has mainly the aim to integrate Web Services by workflows specified in a legacy choreography language called XML Simple conceptual unified language flow (XSculf). Biopipe [11] framework instead, provides a set of wrappers to directly interface resources like executable programs and data adaptors. It doesn't support the use of synchronization operators, like fork and join, because a bioinformatics experiment is just a sequential pipeline. Wildfire [18] provides an integrated environment for the construction and execution of workflows based only on EMBOSS/Jemboss [3] applications. Pegasys [14] system enables bioscientists to create and manage sequence analysis workflows. It includes numerous analytical tools and provides database capacities to maximize information captured during the execution of a workflow. Besides the fact that, the above mentioned systems are not Workflow Managent Systems w.r.t the Workflow Reference model, none of them have been designed to face unforeseen circumstances -if a resource is missed or something goes wrong- and to take the most convenient decision in the dynamic environment in which they act. And, their workflows are generally static, so as their workflow engine centralizes the execution and the coordination of the computation.

In this work, we intend to overcome the above limitations by proposing the dynamic generation of a workflow engine, associated to a single workflow specification. Our approach exploits the proactiveness of agent-based technology to embed the application domain features inside the agents behavior. The resulting workflow engine is a multiagent system -a distributed, concurrent system-typically open, flexible, and adaptive.

The paper is organized as follows. In Section 2 we described our workflow engine architecture based on agent technology. Then, in Section 3 we explain our implementation on Hermes agent-based mobile middleware. In Section 4, we discuss the “Oncology over Internet” project, as a case study of a proactive workflow engine in e-Science. We conclude in Section 5.

2 Agent-based Workflow Engine

A workflow is a distributed application, that involves the coordinated execution of human and system activities, usually, in an heterogeneous environment. Based on our previous work [5], we consider a workflow as coordination model for a pool of agents -Workflow Executors (WEs)- that implements the workflow engine for a specific workflow instance.

Agents are autonomous active entities that can perceive, act and react in their environment, and communicate with other agents. The agents can be mobile. A collection of agents able to cooperate, in their autonomy, for a common goal forms a multiagent system.

In our approach, the generation of the workflow engine is performed by a compiler in a two phase agent-generation procedure. In the first step a User Level Workflow (ULW), specified by a workflow specification language, is mapped to an Agent Level Workflow (ALW). This mapping is performed by recursively substituting activities of the user-level specification with a workflow of primitive agent-level activities. A User-Level Activity Database (ULAD) maintains the correspondence between user-level activities and ALW. The ALW specifies all entities involved in the execution of a workflow; thus the constraint of spatial and temporal coupling communication can be respected since the compiler knows exactly when communication takes place and which are the receivers and which the senders. In the second step, the compiler concretely generates agents from the ALW specification. To achieve this result, the compiler uses the User-Level Activity Implementation Database (ULAID) and the Database of Skeleton (DoS). The ULAID stores the implementation of agent-level activities, and DoS stores "empty" implementation of agents (the skeletons). A WE is obtained by plugging the specific behavior into the skeleton. The resulting set of WEs gives rise to a agent-based workflow engine whose role will be compliant to the WfMS architecture as later described in Figure 4

The above approach has been implemented on Hermes architecture [6] whose detailed description in given in the next section.

3 BioAgent: Hermes Deployment in Bio-domain

Hermes is an agent-based mobile middleware, for the design and the execution of activity-based applications in distributed environments. It is structured as a component-based, agent-oriented system with a 3-layer software architecture: user layer, system layer and run-time layer (Figure 1). User layer allows designers to specify their application as a workflow of activities using the graphical notation provided by DroFlo [7] and JaWE editor [8]. System layer provides a context-aware compiler to generate a pool of user mobile agents from the workflow specification. Run-time layer supports the activation of a set of specialized service agents, and it provides all necessary components to allow agent discovery, mobility, creation, communication and security. Service-Agents (SAs) in the run-time layer are localized to one platform to interface with the local execution environment. User-Agents (UAs) in the system layer are Workflow Executors (WEs), created for a specific goal that, in theory, can be reached in a finite time by interacting with other agents; afterward the agent will die by killing itself. XML Process Definition Language (XPDL) [19], a WfMC standard, has been chosen as workflow specification language at user layer. At the end of a two steps process a compiler translates XPDL to a Multi Agent System (MAS) as Java bytecode ready to be executed in Hermes middleware. Hermes can be configured for specific application domains by adding domain-specific component libraries (ULW, ULAD, ULAID) and thus customizing in a proper way through service agents. The deployment of Hermes in the bio-domain is named, in the sequel of this paper, as BioAgent. BioAgents is a workflow management system for bio-scientists. It represents a flexible environment suitably designed to support the bioscientist's activities during an in-silico experiment. The main BioAgent functionalities supported by a set of specialized cooperative bio-service agents (SA) are described as follows and shown in Figure 1:

Data and Tools Integration - AIXO SA [1] provides a set of wrappers able to access and to present any data source as a collection of XML documents. AIXO (Any Input XML Output) is flexible and modular, it allows to manage many input data sources from HTML to XML, database, flat file, CGI and command line programs;

Web Services Discovery and Invocation - Matchmaker SA [4] localizes services and biomedical resources in general, that best fit the requests of a User Agent. While, the WSIF SA Service Agent allows other agents to dynamically invoke a Web Service. Moreover, SoapLab SA [13] can control a set of Web Services providing programmatic access to many bioinformatics applications on remote computers.

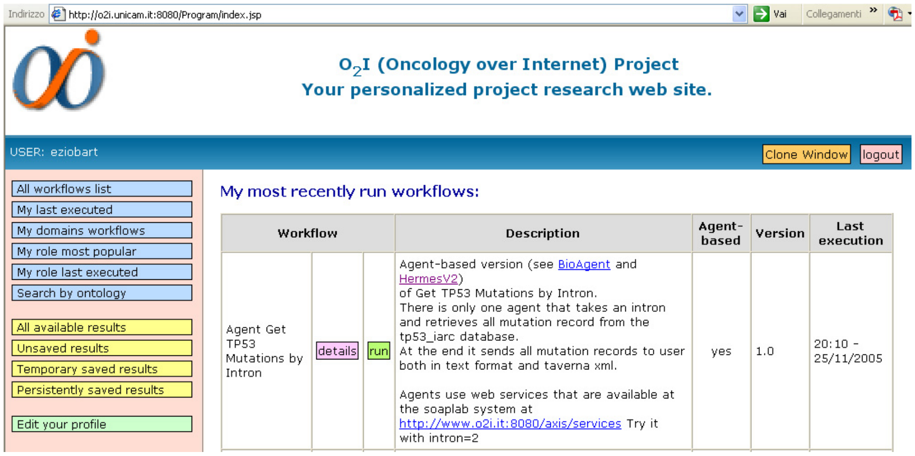
XML Manipulation - XQuery SAprovides a tool to manipulate and query an XML document creating a suitable view of a Web Service invocation or a wrapper output;

Input and Output Management - XUL SAallows the automatic frame-based form generation to support the iteraction among the bioscientist user and WEs to monitor and dynamically change the workflow execution. Furthermore, Email SA allows user to receive the final and intermediate results by email.

4 Case Study: Oncology over Internet Project

BioAgent/Hermes Workflow Manamegement System has been used and validate within the Oncology over Internet (O2I) [1] project. The main goal of the project has been the design of a framework to support searching, retrieving and filtering information from Internet for oncology research and clinics. To that purpose, we have design and developed a Web portal [2] (see Figure 2) to manage, organize and execute workflows of biomedical interest, whose system architecture is shown in Figure 3. It includes three main components: the workflow manager (WCA), the user interface (UI) and the workflow executor (WE). Workflows are created

[1] http://www.o2i.it
[2] http://www.o2i.it:8080/Program



O_2I (Oncology over Internet) Project
Your personalized project research web site.

My most recently run workflows:

Workflow			Description	Agent-based	Version	Last execution
Agent Get TP53 Mutations by Intron	details	run	Agent-based version (see BioAgent and HermesV2) of Get TP53 Mutations by Intron. There is only one agent that takes an intron and retrieves all mutation record from the tp53_iarc database. At the end it sends all mutation records to user both in text format and taverna xml. Agents use web services that are available at the soaplab system at http://www.o2i.it:8080/axis/services Try it with intron=2	yes	1.0	20:10 - 25/11/2005

Fig. 2. Screenshot of O2I Portal

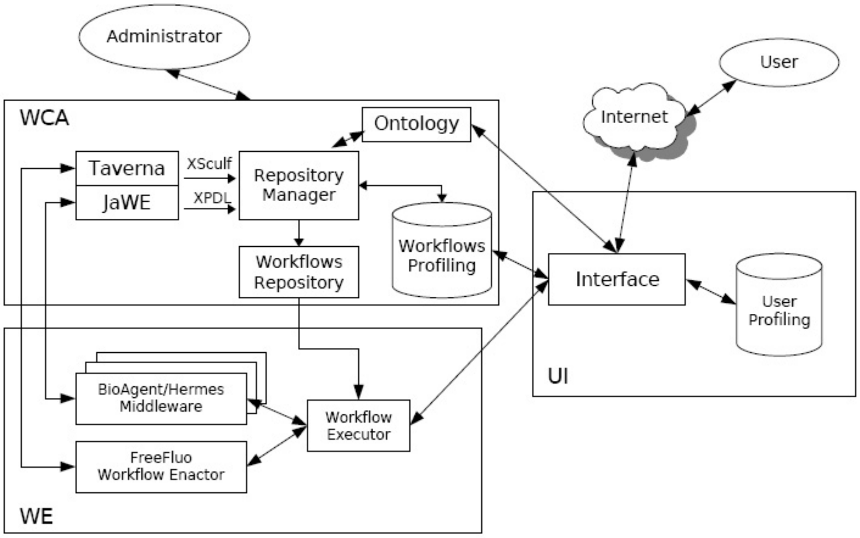


Fig. 3. The general O2I system architecture includes three main blocks: workflow creation and annotation (WCA), user interface (UI) and workflows execution (WE).

and tested by administrator both in XSculf using Taverna Workbench and in XPDL using JaWE. They are then stored in a workflow repository. In both cases, workflows can be annotated by using a specially designed ontology. This ontology describes bioinformatics tasks on the basis of their input and output data, processing type and application domain. The user interface supports end users authentication and profiling and allows for the selection and launch of workflows. User can choose the workflow to be executed selecting a proper user and workflow profile or the keywords provided by the ontology. The selection can be assisted by the user profile and by ontology. Workflows are executed by the third component that is based on FreeFluo Workflow Enactor and BioAgent/Hermes middleware. While the first is used to carry out the Taverna workflow execution, the second compile XPDL specification generating opportunely a pool of mobile Workflow Executors. In this latter case the workflow execution is carried out by proactive, cooperative User Agents that interact with bio-service Agents through messages exchange, and when necessary, decentralize the workflow execution exploiting mobility.

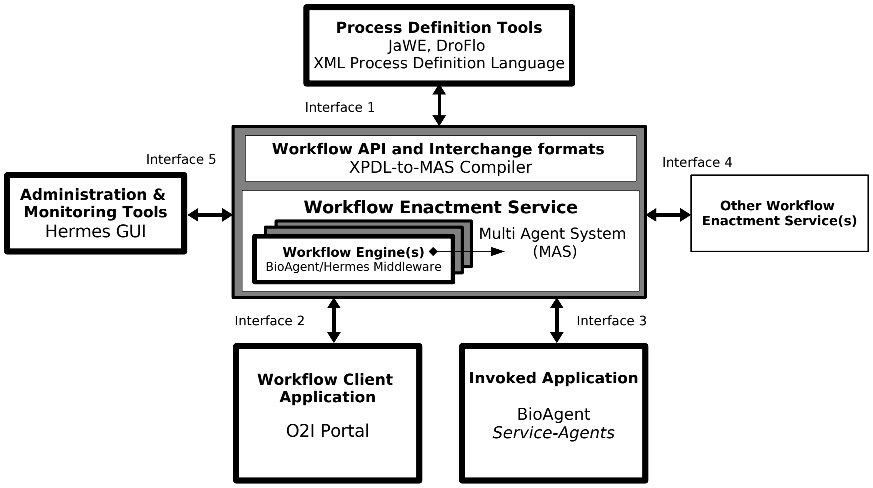

Process Definition Tools
JaWE, DroFlo
XML Process Definition Language
Interface 1
Workflow API and Interchange formats
XPDL-to-MAS Compiler
Interface 5
Interface 4
Administration & Monitoring Tools
Hermes GUI
Workflow Enactment Service
Workflow Engine(s)
BioAgent/Hermes Middleware
Multi Agent System (MAS)
Other Workflow Enactment Service(s)
Interface 2
Interface 3
Workflow Client Application
O2I Portal
Invoked Application
BioAgent
Service-Agents

Fig. 4. Bioagent/Hermes WfMS according to WfMC Reference Model

5 Conclusion

In this work, we proposed an agent-based Workflow Engine that exploits the proactiveness of agents to adapt to a dynamic execution environment. BioAgent/Hermes is an example of proactive workflow engine that together with O2I portal, JaWe, DroFlo and XPDL, as shown in Figure 4, costitues according to the Workflow Management Coalition Reference Model, a Workflow Management System.

Acknowledgements

This work was partially supported by the Italian Ministry of Education, University and Research (MIUR), projects "Oncology over Internet (O2I)" and "Laboratory of Interdisciplinary Technologies in Bioinformatics (LITBIO)".

References

1. E. Bartocci, L. Mariani, and E. Merelli. An XML view of the "World". In *Int. Conference on Enterprise Information Systems, ICEIS*, pages 19–27, 2003.
2. N. Cannata, E. Merelli, and R. B. Altman. Time to organize the bioinformatics resourceome. *PLoS Comput Biol.*, 1(7):e76, 2005.
3. T. Carver and A. Bleasby. The design of jemboss: a graphical user interface to emboss. *Bioinformatics*, 19(14):1837–1843, 2003.
4. F. Corradini, C. Ercoli, E. Merelli, and B. Re. An agent-based matchmaker. In *proceedings of WOA 2004 dagli Oggetti agli Agenti - Sistemi Complessi e Agenti Razionali*, 2004.
5. F. Corradini, L. Mariani, and E. Merelli. An agent-based approach to tool integration. *Journal of Software Tools Technology Transfer*, 6(3):231–244, 2004.
6. F. Corradini and E. Merelli. Hermes: agent-base middleware for mobile computing. In *Mobile Computing*, volume 3465, pages 234–270. LNCS, 2005.
7. DroFlo. Openwfe. http://web.openwfe.org/, 2005.
8. Enhydra. Jawe. http://jawe.enhydra.org/, 2003.
9. A. Garcia Castro, S. Thoraval, L. Garcia, and R. MA. Workflows in bioinformatics: meta-analysis and prototype implementation of a workflow generator. *BMC Bioinformatics*, 6(1):87, 2005.
10. D. Hollingsworth. The Workflow Reference Model, January 1995.
11. S. Hoon et al. Biopipe: a flexible framework for protocol-based bioinformatics analysis. *Genome Research*, 8(13):1904–15, 2003.
12. T. Oinn et al. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–54, 2004.
13. M. Senger, P. Rice, and T. Oinn. Soaplab - a unified sesame door to analysis tools pages. In *Proceedings of UK e-Science All Hands Meeting*, 2003.
14. S. Shah et al. Pegasys: software for executing and integrating analyses of biological sequences. *Bioinformatics*, 1(5):40, 2004.
15. L. Stein. Creating a bioinformatics nation. *Nature*, 417:119–120, 2002.
16. R. Stevens and Others. Performing in silico experiments on the grid: a users perspective. *Proceedings of UK e-Science All Hands Meeting*, page 4350, 2003.
17. R. Stevens, A. Robinson, and C. Goble. mygrid: personalised bioinformatics on the information grid bioinformatics. *Bioinformatics*, (19):302 – 304, July.
18. F. Tang et al. Wildfire: distributed, grid-enabled workflow construction and execution. *BMC Bioinformatics*, 6(1):69, 2005.
19. WfMC. Xml process definition language (xpdl). WfMC standard, October 2005.