

# Enacting Proactive Workflows Engine in e-Science

Ezio Bartocci, Flavio Corradini, and Emanuela Merelli

Università di Camerino, Dipartimento di Matematica e Informatica  
Camerino, 62032, Italy

{ezio.bartocci, flavio.corradini, emanuela.merelli}@unicam.it

**Abstract.** The dynamic nature and the geographic distribution of scientific resources, require flexible and adaptive 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-specific, agent-based workflow engines from workflow specifications. The workflow engine is a proactive multiagent system -a distributed, concurrent system- whose autonomous components interact in performing the workflow activities in a specific domain. The proposed approach has been implemented on Hermes, agent-based mobile computing middleware, and tested 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, and the wide use of the Web has fostered scientists to share 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 concurrent activities- are laboriously executed in a large, distributed and dynamic environment.

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 workflow and agent-based technologies [2] together with an effective and efficient resource management system [1] could be a good start to face the complexity that surrounds scientist’s work.

An in-silico experiment can be naturally specified as a workflow of activities implementing the data analysis process in a standardized environments. The workflow owns the advantage to be reproducible, traceable and compositional by reusing other workflows; features that are fundamental 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 [5], is named Workflow Management System (WMS). In e-Science domain, although several systems that support the daily work of a bioscientist have been proposed in literature [7, 8, 4], they are not compliant to the WfMC Reference Model. And, none of them have been designed to face unforeseen circumstances and to take on the fly decisions. Moreover, workflows are generally static and workflow engines centralize 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 and mobility 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.

In the remainder of this paper we describe our workflow engine architecture based on agent technology, then, we explain our implementation on an agent-based mobile middleware. The approach is illustrated with a case study on the "Oncology over Internet" project.

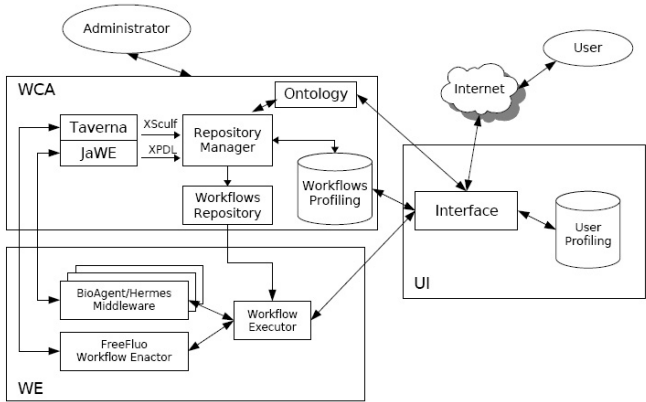
## 2 Agent-Based Workflow Engine

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 [2], we consider a workflow as coordination model for a pool of agents -workflow executors- that implements the workflow engine for a specific workflow instance. In this context agents are autonomous active entities, encapsulate the execution of independent activities, execute their tasks, concurrently to the work of the other agents, cooperate, in their autonomy, for a common goal forms a multiagent system (MAS) [6].

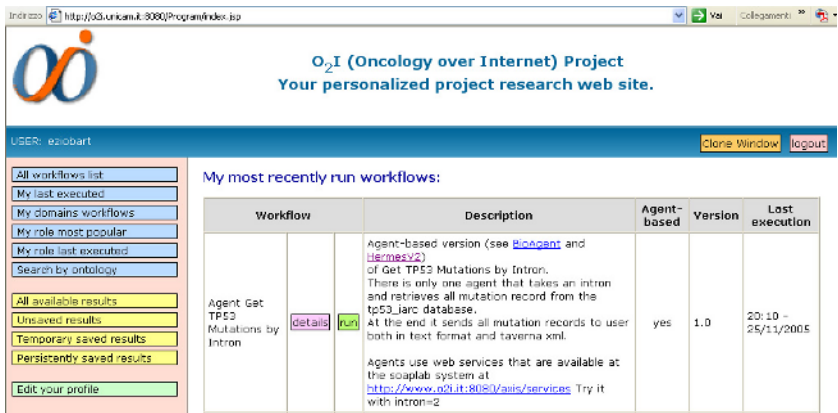
In our framework, a user, after defining a workflow with a graphical notation, compiles the workflow specification in a set of primitives performing the behaviours of a pool of agents. This approach has been implemented on Hermes [3] architecture. Hermes is an agent-based mobile middleware. The choice to use Hermes has been conditioned by the 3-layers -user, system, runtime-, component-based architecture that facilitates the management of domain-specific components [2] toward the development of a workflow-to-MAS compiler. Moreover, Hermes supports the agent mobility through code mobility. User layer allows designers to specify their application as a workflow of activities using the graphical notation. System layer provides a domain-specific 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 domain-specific service agents and it supports agent discovery, mobility, creation, communication and security. Service agents in the run-time layer are localized in one platform to interface the local execution environment. User agents in the system layer are workflow executors. They are dynamically created for a specific goal that, in theory, is reached in a finite time by interacting with other agents.

### 3 Case Study: Oncology over Internet Project

The proposed approach has been tested within the Oncology over Internet (O2I) project (<http://www.o2i.it>). The main goal of the project has been the design of a framework to support searching, retrieving and filtering information from Internet in the oncology research and clinics domain. The O2I system architecture is shown in Figure 1. It includes three main components: the workflow manager (WCA), the user interface (UI) and the workflow executor (WE). Workflows are created and tested by administrator both in XSculf using Taverna Workbench and in XPDL -a WfMC standard- using JaWE -an open source workflow editor that supports XPDL. The user interface supports end users authentication and profiling and allows for the selection and launch of workflows (see Figure 2).



**Fig. 1.** The general O2I system architecture includes three main blocks: workflow creation and annotation (WCA), user interface (UI) and workflows execution (WE)



**Fig. 2.** Screenshot of O2I Portal

Workflows are executed either on FreeFluo Workflow Enactor by a centralized execution or on Hermes by a proactive, decentralize execution of BioAgents. While the first is used to carry out the Taverna workflow execution, the second compiles XPD L specification into a pool of mobile workflow executors. In the latter case the workflow execution is carried out by proactive, cooperative bio-user agents that interact with bio-service agents through messages exchange, and when necessary, decentralize the workflow execution exploiting mobility.

## 4 Conclusion

In this work, we proposed an agent-based Workflow Engine that exploits the proactiveness of agents to adapt to a dynamic, domain-specific, execution environment. We developed a prototype for the biomedical domain; the resulting Workflow Management System, according to the Workflow Management Coalition Reference Model, supports the dynamic generation of proactive workflow engines.

## 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. N. Cannata, E. Merelli, and R. B. Altman. Time to organize the bioinformatics resourceome. *PLoS Comput Biol.*, 1(7):e76, 2005.
2. 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.
3. F. Corradini and E. Merelli. Hermes: agent-base middleware for mobile computing. In *Mobile Computing*, volume 3465, pages 234–270. LNCS, 2005.
4. 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.
5. D. Hollingsworth. The Workflow Reference Model, January 1995.
6. N. R. Jennings. On agent based software engineering. *Artificial Intelligence*, 117(2):277–296, 2000.
7. T. Oinn et al. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–54, 2004.
8. S. Shah et al. Pegasys: software for executing and integrating analyses of biological sequences. *Bioinformatics*, 1(5):40, 2004.