Agent Technology and Generic Workflow Management in an e-Science Environment

Zhiming Zhao Adam Belloum Peter Sloot Bob Hertzberger

Informatics Institute, University of Amsterdam
Kruislaan 403, 1098SJ, Amsterdam, the Netherlands
{zhiming|adam|sloot|bob}@science.uva.nl
To appear in GCC2005, Beijing, China http://kg.ict.ac.cn/gcc2005/.

Abstract. In e-Science environments, the support for scientific workflows emerges as a key service for managing experiment data and activities, for prototyping computing systems and for orchestrating the runtime system behaviour. Supporting domain specific applications via a common e-Science infrastructure enables knowledge sharing among different applications, and thus can broaden the range of the application and multiply the impact of scientific research. However, most of the existing workflow management systems are driven by the domain specific applications; the applicability to different domains is limited. In this paper, we discuss possible solutions to this problem and present our research in an ongoing project: Virtual Laboratory for e-Science (VL-e). Agent technologies are used to encapsulate the intelligence for problem solving strategies and for workflow orchestration.

1 Introduction

In e-Science environments, Scientific Workflow Management Systems (SWMS) emerge as the fibre to glue different levels of issues: experiment planning, resources deployments and the runtime execution control of the experiment. By automating the management of experiment routines, a SWMS hides the underlying details of the Grid infrastructure and allows a scientist to focus on the high level domain specific aspects of the experiments [1,2]. In the past decade, SWMSs have been realised in different application domains, e.g., in bio informatics [3, 4], in high energy physics [5], and in astronomical observations [6].

Reusing the successful and stable results of SWMSs can not only improve the efficiency for developing advanced high-level application specific functionality, but also reduce cost and risks for utilising an e-Science infrastructure in a new problem domain. More importantly, supporting domain specific applications via a common infrastructure enables knowledge sharing among different applications and thus can broaden the range of the application and multiply the impact of scientific research [7]. This issue has been highlighted in a number of ongoing e-Science projects; three research efforts can be enumerated. The first one is from the resource perspective. A standard interface for coupling SWMS resources is essential to improve the reusability of a SWMS. In Taverna, an

open world assumption is adopted [8]; Grid service is used as the basic architecture to interconnect resources, therefore the services developed by the other SWMSs can also be deployed. Using a knowledge backbone, e.g., an Ontology based mechanism, to enhance a SWMS enables the semantic level sharing and querying of various SWMS resources as in [9, 10]. The second effort is to distinguish the reusable services from a SWMS and to encapsulate them as generic components in an e-Science environment, such as generic scheduling strategies in Pegasus [11]. The last one is to reuse different SWMS by providing different levels of interoperability mechanisms among them [12]. Most of the recent discussion is at the resource level, which aims to invoke software resources of another SWMS by wrapping the components. In this paper, we present our research along the third effort in the context of a Dutch e-Science project: Virtual Laboratory for e-Science (VL-e) [13].

The VL-e project aims to realise a generic e-Science framework where scientists from different domains can share their knowledge and resources, and perform domain specific research. VLAM-G (Virtual Laboratory Amsterdam for Grid) environment [14], a Grid enabled e-Science framework developed in a previous project[1], is currently used as the first prototype. The VLAM-G environment provides a user friendly interface for managing software components and for composing reusable experiment templates, but only limited support for scientific workflows. In this paper, we discuss the feasibility and challenges in including scientific workflow support as part of generic e-Science services, and propose an agent based solution.

This paper is organised as follows. First, we analyse the basic issues in realising a SWMS and briefly describe the research context of the Dutch Vl-e project. After that, we discuss the shortcomings of current implementation of the generic VL-e framework, and propose an agent based solution to improve it. The differences between our solution and the other related work are also discussed.

2 Generic scientific workflow management in an e-Science framework

In [15, 16], we distinguished three main functional components from a SWMS: a workflow model, an engine and user support. Due to the diversity of the science disciplines, workflow models are often domain specific, e.g., data streams between experiment instruments and the analysis tools are modelled as a workflow in high energy physics applications [17], while human involved adaptation in predefined imaging processing are highlighted in medical imaging applications [18].

In principle, a generic e-Science SWMS can be derived from domain specific SWMSs using two basic approaches.

An *abstraction* approach abstracts the common characteristics from different SWMS implementations, including the workflow model, the engine, and the user support. Generic solutions to these abstracted issues are then encapsulated as reusable workflow services in the e-Science framework. This approach is

[1] Virtual Laboratory II.

pretty much like deriving classes from objects in an Object Orient methodology. The domain specific features will stay at application level, and all the generic workflow support will be provided by the e-Science framework.

Ideally, this approach will contribute a generic SWMS which can serve different domain specific applications. It is based on the condition that the existing SWMSs have fully captured the dynamics of the domain specific scientific experiments, and a generic workflow model can be possibly abstracted from these domain specific workflows models. However, in practice, this condition is far away from the reality. Having mature domain specific workflow models for different domains is not going to be truth in a short period of time. Domain scientists may have continuously changing requirements on the workflow model when they have new ambitions on exploring his domain problems which makes a workflow model take very long time to evolve as a *mature* one. A more practical approach, called an *aggregation* approach, can be used.

An *aggregation* approach starts from a success model of domain specific workflows and extends it to support other domains by including workflow engines for that domain into the system. Compared to the abstraction approach, this one reuses the intelligence of existing SWMSs as a whole, and avoids the reimplementation of the same workflow model.

Different levels of interoperability between a slave SWMS and the host SWMS is essential to implement this approach. It requires detailed knowledge of both SWMS implementations, in particular when the underlying middleware of SWMSs differ.

Theoretically, both approaches are applicable in realising an e-Science environment. However, from the state of the art of the domain specific SWMSs, the aggregation approach is more practically feasible. We will discuss how we apply this approach in the VL-e project in the next Section.

In the next section, an agent based solution is proposed.

3 VL-e Workflow Conductor (VLWF-Conductor)

In this section, we propose an architecture called VL-e workflow Conductor (VLWF-Conductor), which provides solutions to the missions mentioned above from three aspects: providing an enhanced workflow model, an agent based flow execution engine, and necessary user support.

3.1 A Petri net based model

In our early work, we have implemented a place transition (PT) graph based mechanism called *scenario net* [19] to model the interaction constraints between components in an interactive simulation system. In a scenario net, transitions are used to model the activities that an actor (also called a role) in a flow will perform, and places and the relation links between places and transitions are used to model dependencies among activities. In a scenario net, places and the relation links from places to transitions can also associate with guard expressions. In [20],

we have demonstrated the application of scenario nets in modelling scenarios in interactive simulation systems.

In VLWF-Conductor, scenario net is used as the basic mechanism to model the workflow.

3.2 An agent based workflow engine

To execute a workflow, an engine necessarily deals with different levels of issues: locating computing resources, scheduling tasks, orchestrating the activities, passing data, adapting the execution, and managing runtime information. Decentralised these control and realising them as autonomous components can encapsulate the intelligence and hide the complexity from different levels.

Agent technologies provide a suitable approach to include control intelligence in the behaviour of a set of operations; therefore, we use them to encapsulate the control intelligence and to carry out the flow control. We group the support for these phases into two parts: one is for the pre-processing and post-processing of a workflow, and one is for managing computing tasks of the workflow. The control intelligences in these parts are encapsulated as two agents: a *Study Manager* and a *Scenario Conductor*.

A *Study manager* (SM) is an agent for managing the lifecycle of an experiment. A SM is instantiated for each workflow instance; it manages different types of experiment data and schedules the execution of a workflow by applying domain specific strategies. When a SM receives a workflow description, it first does necessary pre-processing of the workflow, e.g., checking if involved resources for the workflow can be located, if similar experiments have already been executed, and if the mata data for different experiment processes available. After that, it makes plan for scheduling computing parts of the workflow.

A *Scenario Conductor* is instantiated by a SM for executing a sub-workflow with computing tasks. A SC realises the functionality for discovering resources, mapping workflow onto the resources, interpreting workflow and orchestrating the runtime activities of the resources. A SC also acts as a wrapper to a foreign workflow engine when it is employed in the workflow execution. A SC realises the engine level interoperability among different sub-workflows. When a foreign engine is included a specific sub-workflow, the execution intelligence of that engine is interfaced to the SC via top-level workflow as a whole. The SM and SC handle the coordination issues.

At runtime, agents collaboratively manage the information of an experiment and orchestrate the computing tasks.

4 Discussion and conclusions

As we have distinguished in the introduction section that there are three efforts for deriving generic services from domain specific SWMSs. The discussion in this paper belongs to the third one: reusing existing workflow systems in an e-Science environment by enabling the interoperability among them via a generic framework.

We have not fully implemented VLWF-Conductor, yet we did test the feasibility for integrating the VLAM-G framework with the other workflow systems, e.g., Nimrod [21]. From the discussion, we can at least conclude follows:

1. Generic workflow management services are essential to realise a common e-Science framework for transferring and sharing knowledge among domains.
2. Aggregating the state of art SWMSs in an e-Science environment is a feasible approach to realise a reusable framework for domain specific applications.
3. Agent technologies are a suitable approach to implement the control intelligence for flow control.

5 Future work

We are currently surveying a list of SWMSs. The implementation details of these SWMSs are analysed from the perspective of application characteristics. Using the survey results, we will then implement the first prototype of VLWF-Conductor and develop the interface for other workflow engines.

Acknowledgement. This work was carried out in the context of the Virtual Laboratory for e-Science project (www.vl-e.nl). Part of this project is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC&W) and is part of the ICT innovation program of the Ministry of Economic Affairs (EZ). The authors of this paper would like to thank all the members in the VL-e SP2.5 sub program.

References

1. Jr. George Chin, L. Ruby Leung, Karen Schuchardt, and Debbie Gracio. New paradigms in problem solving environments for scientific computing. In *Proceedings of the international conference of Intelligent User Interface*, San Francisco, 2002.
2. R. McClatchey and G. Vossen. Workshop on workflow management in scientific and engineering applications report. *SIGMOD Rec.*, 26(4):49–53, 1997.
3. Mark Ellisman, Michael Brady, David Hart, Fang-Pang Lin, Matthias Muller, and Larry Smarr. The emerging role of biogrids. *Commun. ACM*, 47(11):52–57, 2004.
4. Robert D. Stevens, Alan J. Robinson, and Carole A. Goble. mygrid: personalised bioinformatics on the information grid. In *ISMB (Supplement of Bioinformatics)*, pages 302–304, 2003.
5. I. Augustin, F. Carminati, J. Closier, E. van Herwijnen, J. J. Blaising, D. Boutigny, and et al. Hep applications evaluation of the edg testbed and middleware. *CoRR*, cs.DC/0306027, 2003.
6. The astrogrid project homepage. In *http://www.astrogrid.org/*, 2005.
7. Jim Blythe Carl Kesselman Hongsuda Tangmunarunkit Yolanda Gil, Ewa Deelman. Artificial intelligence and grids: Workflow planning and beyond. *IEEE Intelligent Systems*, pages 26–33, January 2004.
8. Tom Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R. Pocock, Anil Wipat, and Peter Li. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics Journal.*, online, June 16, 2004.

9. Carole Goble and David De Roure. The grid: an application of the semantic web. *ACM SIGMOD Record*, 31(4):65–70, 2002.
10. Srini Narayanan and Sheila A. McIlraith. Simulation, verification and automated composition of web services. In *Proceedings of the eleventh international conference on World Wide Web*, pages 77–88. ACM Press, 2002.
11. Ewa Deelman, James Blythe, Yolanda Gil, Carl Kesselman, Gaurang Mehta, Sonal Patil, Mei-Hui Su, Karan Vahi, and Miron Livny. Pegasus: Mapping scientific workflows onto the grid. In *European Across Grids Conference*, pages 11–20, 2004.
12. MyGrid. Link-up project - e-science sisters programme. In *http://www.mygrid.org.uk/linkup/*, 2005.
13. VL-e. Virtual laboratory for e-science. In *http://www.vl-e.nl/*, 2005.
14. H. Afsarmanesh, R.G. Belleman, A.S.Z. Belloum, A. Benabdelkader, J.F.J. van den Brand, and et al. VLAM-G: A Grid-based Virtual Laboratory. *Scientific Programming: Special Issue on Grid Computing*, 10(2):173–181, 2002.
15. Zhiming Zhao, Adam Belloum, Hakan Yakali, Peter Sloot, and Bob Hertzberger. Dynamic workflow in a grid enabled problem solving environment. In *Proceedings of the 5th International Conference on Computer and Information Technology (CIT2005)*, page accepted, Shanghai, China, September 2005. IEEE Computer Society Press.
16. Zhiming Zhao, Adam Belloum, Adianto Wibisono, Frank Terpstra, Piter T. de Boer, Peter Sloot, and Bob Hertzberger. Scientific workflow management: between generality and applicability. In *Proceedings of the International Workshop on Grid and Peer-to-Peer based Workflows in conjunction with the 5th International Conference on Quality Software*, page accepted, Melbourne, Australia, September 19th-21st 2005. IEEE Computer Society Press.
17. Henri Casanova. Distributed computing research issues in grid computing. *ACM SIGACT News*, 33(3):50–70, 2002.
18. Lewis Hassell and John Holmes. Modeling the workflow of prescription writing. In *SAC '03: Proceedings of the 2003 ACM symposium on Applied computing*, pages 235–239, New York, NY, USA, 2003. ACM Press.
19. Zhiming Zhao, G. D. van Albada, and P. M. A. Sloot. Agent-based flow control for hla components. *International journal of simulation transaction, special issue Agent Directed Simulation*, 81(7):in press, 2005.
20. Z. Zhao. *An agent based architecture for constructing interactive simulation systems.* PhD thesis, University van Amsterdam, Amsterdam, The Netherlands, (Promoter: Prof. Dr. P. M. A. Sloot), 2004.
21. Tom Peachey, David Abramson, Andrew Lewis, Donny Kurniawan, and Rhys Jones. Optimization using nimrod/o and its application to robust mechanical design. In *PPAM*, pages 730–737, 2003.