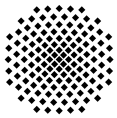

Universität Stuttgart

SimTech
Cluster of Excellence

M. Sonntag D. Karastoyanova F. Leymann

The Missing Features of Workflow Systems for Scientific Computations

Stuttgart, August 2010

Institute of Architecture of Application Systems (IAAS)
University of Stuttgart,
Universitaetsstrasse 38
70569 Stuttgart, Germany
{sonntag, karastoyanova, leymann}@iaas.uni-stuttgart.de
www.iaas.uni-stuttgart.de

Abstract This paper discusses technical aspects of how business workflow management systems can be improved in order to apply them in the field of scientific workflows and reap all their benefits. We give recommendations how to address the discovered gaps in support for scientific applications. The approach we follow addresses the requirements of scientists and scientific applications, which we also identify in this work.

Keywords Scienific workflows, business workflows.

Reference Sonntag, M., Karastoyanova, D., and Leymann, F. (2010) "The Missing Features of Workflow Systems for Scientific Computations", 3rd Grid Workflow Workshop (GWW), Software Engineering Conference, GI-Edition Lecture Notes in Informatics (LNI), P-160.

Stuttgart Research Centre for Simulation Technology (SRC SimTech)

SimTech – Cluster of Excellence
Pfaffenwaldring 7a
70569 Stuttgart
publications@simtech.uni-stuttgart.de
www.simtech.uni-stuttgart.de

The Missing Features of Workflow Systems for Scientific Computations

Mirko Sonntag, Dimka Karastoyanova, Frank Leymann

Institute of Architecture of Application Systems
University of Stuttgart
Universitaetsstrasse 38
70569 Stuttgart, Germany
{sonntag, karastoyanova, leymann}@iaas.uni-stuttgart.de

Abstract: This paper discusses technical aspects of how business workflow management systems can be improved in order to apply them in the field of scientific workflows and reap all their benefits. We give recommendations how to address the discovered gaps in support for scientific applications. The approach we follow addresses the requirements of scientists and scientific applications, which we also identify in this work.

1 Introduction

There are two main application areas driving and utilizing workflows – businesses and scientific computations and experimenting. Since the early 90s, workflow management systems (WfMSs) are applied by enterprises to support their business. Over the years, the workflow technology matured and is nowadays established and well-proven in the business area. Business WfMSs are universally designed independent of the concrete business area of employing enterprises. Because of the generic approach the workflow technology follows, typically lots of configuration options for such systems exist which contributes to their complexity. Therefore it is commonplace for IT experts to implement business processes of enterprises and to set up the software infrastructure. Business workflows often represent the products of enterprises (e.g. a loan approval workflow of a bank stands for the product “loan”) [LR00].

In recent years, workflows gained more and more attention in science to support scientists in their work. Scientific WfMSs are not built using the existing workflow technology but are designed and developed from scratch. The reason can be found in the very different requirements of scientific workflows compared to their business counterpart [Gi07]. Scientific WfMSs are often tailored to a particular application domain. In this context, workflows implement scientific simulations, experiments, and computations typically dealing with huge amounts of data. Scientists model, execute, monitor, and analyze workflows. Since they are no IT experts, special attention is paid on the usability aspects of the systems.

Since recently there are endeavors to join the efforts of these two communities. To be more precise, efforts were made to harness the traditional workflow technology for scientific workflows [BG07, Wa07, Ba08, So10]. This approach is promising since it brings the strengths of the technology to scientific applications. Despite their generic properties, business WfMSs are not yet capable of covering some of the challenges arising because of particular needs of scientists and specific existing software and hardware infrastructures. A full-fledged Grid support, for example, is crucial since many scientific applications are conducted in a Grid environment as they depend on large storage resources and computational power a single scientific institute can hardly afford.

To the best of our knowledge, there is no systematic discussion and comparison of the scientific workflow systems and workflow systems used widely in business applications. Moreover, there are only a few research works that report on how the conventional workflow technology needs to be improved in order to accommodate all the requirements of scientist.

This paper provides a more technical view on problems coming up when employing traditional workflow technology in the scientific area. The considered technical aspects are thereby explained and compared to existing scientific WfMSs by exposing similarities and differences. Another main contribution is a set of recommendations about how the discussed barriers can be overcome. The reader is given a means at hand to understand the challenges of building a scientific WfMS on top of the traditional workflow technology.

The rest of the paper is structured as follows: Section 2 briefly discusses related work in the field of investigating scientific workflows and existing scientific workflow systems. Section 3 presents the technical consideration of business WfMS concepts and their application to scientific workflow management (WfM). Finally, Section 4 concludes the paper and gives directions for future work.

2 Related work

In [Gi07] it is investigated which specific requirements scientific workflows impose on a supporting infrastructure. The discussion is held on a system level considering various functional (e.g. reproducibility, automatic adaptation) and non-functional properties (e.g. stability, usability). As opposed to this, our approach is discussing technical aspects of business WfMSs and their applicability in the area of scientific WfM.

In [BG07] typical characteristics of business workflows are explained (e.g. transactions, fault tolerance, use of standards) and their benefits for scientific workflows are presented. Moreover, features unique to scientific workflows are identified (e.g. evolving nature, non-computer experts as workflow designers, data centrality). In [Lu09] an inverse approach is followed: common features of scientific workflows (e.g. data flow orientation) are collected and compared to business workflows. However, both papers argue on the workflow model and language level. In contrast to that, we compare business and scientific WfMSs and their underlying technical concepts.

The considerations in this paper are based on requirements of scientists on scientific WfMSs. These requirements are both gathered out of the mentioned works and derived from several scientific WfMSs: Kepler [Al04], Taverna [Oi06], and Triana [Ch05] are scientific workflow systems that are able to deal with Web services (WS); Pegasus [De04] and SEGL [Cu08] are workflow systems specialized for the use in a Grid environment; Sedna [Wa07] is a workflow modeler based on the conventional workflow technology; Trident [Ba08] and e-BioFlow [Wa09] are systems based on business WfMSs.

3 Fitting conventional workflow systems for scientific computations

Currently, there are efforts to establish the conventional workflow technology in the scientific domain [Ba08, So10, Wa09, Wa07]. This promises to bring along a number of advantages: simplified collaboration of different scientific groups through the use of technology standards, already existing tools that can be used as development basis, or a higher robustness compared to existing scientific workflow systems, to name just a few. On the other hand, a lot of problems come up when trying to use traditional WfMSs in science because of very indifferent requirements by the users and the employed infrastructures. Business WfMSs are rather general systems since they are intended to serve enterprises independent of their business model and hardware infrastructure. Furthermore, several types of workflows can be handled by one and the same WfMS (production, administrative, collaborative, and ad hoc) [LR00]. This yields a very high complexity of the systems (e.g. number of tools, involved user roles, configuration options) and heavy IT support is needed to translate business processes into a machine-readable form and to customize the software.

Instead of making use of existing and established solutions scientific WfMSs are often built from scratch. This is because the systems are tailored to the needs of scientists [Gi07] and to the respective operation area. Scientists as non-computer experts require, for example, high usability of the software. Scientific workflows are often operated in a specific scientific domain (e.g. e-BioFlow is preferential for life science workflows). This is mainly manifested in the tools built-in in the workflow editor to solve a certain scientific problem in a domain specific language. There are also scientific WfMSs particularly designed for the operation in Grid environments (e.g. Pegasus, SEGL).

In the following, we dig deeper into the issue of investigating the applicability of business WfMSs in the scientific area. To do so, we account for a number of technical aspects of business WfMSs, analyze them, and give recommendations about how to extend or adapt them to meet the requirements of scientists.

Life cycle and tool integration. In business process management (BPM), a very well-known subject is the life cycle of business processes containing several separated and repeatable management phases as shown in Figure 1. Naturally, each phase is conducted by a particular user role with specific tasks, knowledge, capabilities, and working methods. To account for this situation the BPM life cycle is typically supported by many different tools with a complex interplay, e.g. an editor for workflow modeling, or an

engine for workflow execution. As opposed to this, scientific WfMSs often consist of a single tool (e.g. Kepler, Triana, Taverna). This is due to the fact that scientists are typically the only user group of a system. When employing a business WfMS for the use in the scientific domain, the tool portfolio needs to be thinned out and/or integrated so as to be experienced as a single tool for scientists. Such an integrated tool should hide the complexity of the underlying software infrastructure for modeling, configuration, execution, monitoring, and analysis of workflows.

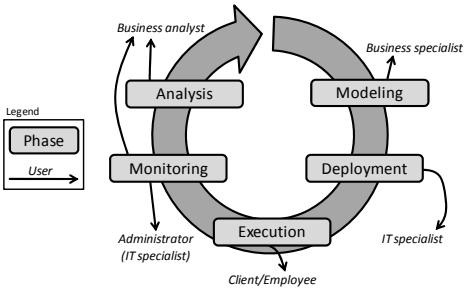

Figure 1: BPM life cycle

Modeling and deployment. An important aspect in business WfM is the deployment of process models on a workflow engine. Deployment is a technical step to put process models into production. It requires deep knowledge of the used infrastructure (e.g. workflow engine, web server) and technology (e.g. WSs, partner links in BPEL [Al07]). The purpose of deployment is manifold: translation of process models into a format optimized for execution (typically held in a database); external configuration of processes increases their reusability in different contexts; installation of processes so that they can be frequently executed even late after modeling; providing processes to customers (e.g. as WS). In contrast, most scientific workflows are simply executed without deployment. The execution code of a workflow is already known on workflow editor level making a translation step unnecessary. Instance state is held in main memory only. Configuration is contained directly in the workflow definition. That means to change its configuration (e.g. bind another resource for a task) the workflow itself has to be changed (adapted in the terminology of conventional workflows). Scientific workflows are typically executed by scientists themselves immediately after modeling or even during modeling in a trial-and-error manner [Wa09]. Workflows are mainly not provided to be invoked by third parties (e.g. WSs are only called synchronously so that no callback is needed). An exception is Triana where a workflow can be explicitly deployed and provided as WS. Nevertheless, if a business WfMS needs to be used in the scientific domain, the deployment step should be kept due to the many advantages it yields, such as reusability of workflow parts, efficient execution with the help of an optimized format, robustness when storing instance state in a database. Since on the other hand scientists are no computer experts and are neither able to nor want to cope with the complexity of a deployment step, we propose to hide the deployment. An option is to conduct it transparently for the scientist as part of a “run workflow” operation. Information needed for deployment could be derived from workflow model properties (e.g. in case of an installation as WS the workflow model’s name combined with an activity name could act as service name and could be part of the port).

Execution. By joining the Grid technology and Service-Oriented Architectures (SOA) with the help of the Open Grid Services Architecture (OGSA) and the Web Services Resource Framework (WSRF) the first steps towards Grid-awareness in business

WfMSs are taken. WfMSs that make use of the WS technology can now access Grid resources offered as stateful WSs [Sl06]. Since Grid support goes beyond conventional WS invocations, an Enterprise Service Bus (ESB) can be used to keep track of all resources on the Grid, match functional properties, negotiate policies, and eventually find and bind appropriate services/resources on behalf of a workflow [We05]. However, Grid-enabled workflow systems in science (e.g. Pegasus, SEGL) implement additional concepts: e.g. searching for idle resources, code and data shipping/staging, job scheduling, authentication, authorization, or credential delegation, which are not issues solved by conventional workflow systems. For a full-fledged Grid support an ESB must be extended to support such concepts. The incorporation of existing Grid middleware services could simplify this exercise.

Monitoring and analysis. Monitoring of business processes is crucial to observe the system state, and to visualize business or performance data on the workflow level with the help of tables and diagrams, for instance. Monitoring is highly customizable and hence a very complex part of WfMSs. Typically lots of instances of a process model are monitored simultaneously because business analysts are mainly interested in aggregated results. In scientific WfM, monitoring of single instances is more important in order to allow scientists to follow the progress of their computations. In existing scientific systems, customizing monitoring is very restricted if possible at all. Adopting business monitoring features in the scientific domain is beneficial due to powerful methods for observing workflow runs, user notifications, and customizability. Nevertheless, the concepts need extensions to fit scientific needs. Monitoring should be focused on single instances and geared towards the workflow model graph structure. Although Oracle's process server [Ora], for example, allows the user to inspect running instances in a graph-based manner, this is done by a particular Web interface instead of an integrated view in the workflow modeling tool. The experience of having the functionality of the workflow system available at one place is not given. Furthermore, the level of monitoring needs to be widened: in a Grid environment, information about used services, underlying operating systems and hardware are required for a coherent view on the overall system [FK04, Mi09]. In business WfM, it is important *that* a service keeps the promised functional and non-functional properties. It is not important *how* a service achieved its aims in terms of used resources, software, and tools. Quite the contrary, this information can even be the business model of enterprises and hence their secret. Therefore, a business WfMS and the utilized infrastructure need to be extended for the use in the scientific domain. The used resources need to provide an interface to reveal current resource properties, such as size of the job queue, installed operating system, number and load of processors. This can be achieved with the help of WSRF. Since WSRF is only a general framework that enables querying the state of a resource, it is up to the resource which properties it is willing to reveal. That means, the incorporated resources need an interface to allow a monitor querying at least an agreed upon number of properties.

Flexibility. In conventional workflow technology, flexibility mechanisms, such as run time adaptation concepts, are investigated to a great extend [RD98, Ka06]. Well-known approaches are, amongst others, to insert or delete an activity, to reiterate parts of a workflow, or to inquire a second opinion to finish an activity [LR00]. Although such

features promise to support scientists in conducting experiments in a trial-and-error manner, they are currently almost unaddressed in the scientific community. An exception is the e-BioFlow system [Wa09] where an ad hoc editor enables scientists to execute and re-execute incomplete workflows (so-called workflow fragments [Eb09]). However, when adopting the concepts in the scientific area, carefully designed extensions are needed on the functional and non-functional level. For instance, modifications of workflows need to be thoroughly tracked for the sake of reproducibility. This is beyond the current scope of auditing information that mainly track events raised during workflow execution. Moreover, the application of adaptation and flexibility concepts in a Grid environment with stateful resources needs to be examined. Consider, for example, the workflow in Figure 2(a) that invokes a program A on resource A. The program creates data that resides on the resource. Later on, the workflow invokes program B that is also located on resource A and that relies on the data produced by program A. Figure 2(b) visualizes the same scenario but the workflow was adapted after the execution of activity 1. Activity 2 now invokes program B on resource B. That means the appropriate data has to be shipped from resource A to B or else program B would raise an error. One could imagine several approaches to solve these kinds of problems, e.g. a validation component of the modeling tool could prohibit such modifications, or the ESB could do the data shipping transparently for the scientist. The situation gets even more complex if the resources were bound lately. A workflow language that allows specifying data dependencies between one or more activities (or even tools) could simplify the solution of this specific problem. Unfortunately, most business WfMSs are control flow-oriented and thus need an appropriate extension.

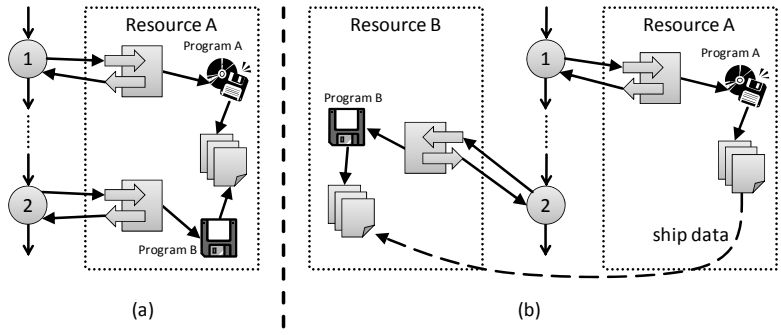

Figure 2: Workflow adaptation in a situation with stateful services

Provenance. For legal or analysis reasons business WfMSs track information about workflow execution on the workflow instance level in an audit trail. Scientists require provenance information to reproduce their results (i.e., obtaining similar results with the same initial data). Auditing data can contribute to a provenance record but are insufficient for an accurate reproducibility. Information beyond the workflow instance level is needed, such as concrete services found, bound, and invoked by an ESB. Provenance tracking is challenging especially in highly dynamic environments as in scientific computing (e.g. resources may come and go, workflows are modified at run time). That means, the audit trail must be at least extended by ESB events that are expressive enough to follow the search, selection, and binding of resources.

4 Conclusion and outlook

This work discussed the technical characteristics of business WfMSs and evaluated them for the purpose of applying them for modeling and execution of scientific applications for simulations, experiments, and computations. Based on this analysis we identified a number of missing features of conventional WfMSs:

- That fact that business WfMSs consist of several, not integrated tools impedes usability for scientists.
- An explicit deployment step requires deep knowledge of the underlying technology (e.g. BPEL's partner link concept, WSs) and the employed infrastructure (e.g. workflow engine, web application server) and is hence hard to accomplish by a non-computer expert.
- Typically, workflows are started by incoming messages or with the help of additional workflow clients. Scientists need to start workflows from within the modeling tool.
- Monitoring is focused on aggregated statistical information on a workflow level over several workflow instances. Scientists require a graph-oriented monitoring of single workflow instances as well as monitoring on additional levels, such as employed tools, operating system and hardware.
- Grid-awareness is uncommon in current ESB implementations [IBM] [ASM] (e.g. searching for idle resources, code and data shipping, job scheduling, authentication, authorization, or credential delegation).
- A seamless trial-and-error workflow design as often required by scientists is usually not supported by conventional WfMSs. Flexibility mechanisms in stateful Grid environments impose additional yet unaddressed requirements.
- Audit trails store information on a workflow instance level and thus do not provide a provenance tracking mechanism that ensures confidence in and reproducibility of scientific results. Especially, provenance tracking in flexible scientific environments is challenging.

We provided recommendations as of how the identified drawbacks can be addressed. Based on the results of this work we aim at implementing a scientific WfMS built on top of an existing open-source workflow system, a modeling tool, and a service bus. Special attention will be paid to the usability, Grid-awareness, monitoring, flexibility, and data-centrality of the system.

Acknowledgements. The work presented in this paper has been funded by the DFG Cluster of Excellence Simulation Technology[1] (EXC310).

References

[Al04] Altintas, I. et al.: "Kepler: An Extensible System for Design and Execution of Scientific Workflows". In: *Int'l Conf. on Scientific and Statistical Database Management*, 2004.

[1] `http://www.simtech.uni-stuttgart.de`

[Al07] Alves, A. et al.: Web Services Business Process Execution Language (BPEL) Version 2.0, OASIS standard, April 11th, 2007.

[ASM] Apache ServiceMix. [Online] http://servicemix.apache.org/home.html [February 17th, 2010]

[Ba08] Barga, R. et al.: "The Trident Scientific Workflow Workbench". In: *IEEE International Conference on eScience*, 2008.

[BG07] Barga, R.; Gannon, D.: "Scientific versus Business Workflows". In: Taylor et al. (Eds.), *Workflows for e-Science: Scientific Workflows for Grids*, Springer, 2007.

[Ch05] Churches, D. et al.: "Programming Scientific and Distributed Workflow with Triana Services". In: *Concurrency and Computation: Practice and Experience*. Special Issue on Scientific Workflows, 2005.

[Cu08] Currle-Linde, N. et al.: "SEGL: A problem solving environment for the design and execution of complex scientific Grid applications". In: *3rd Russian-German Advanced Research Workshop*, Springer, 2008.

[De04] Deelman, E. et al.: "Pegasus: Mapping Scientific Workflows onto the Grid". In: *2nd European AcrossGrids Conference*, Springer, 2004, pp. 11-20.

[Eb09] Eberle, H. et al.: "Workflow Fragments". In: OTM Part I, 2009.

[FK04] Foster, I.; Kesselmann, C.: "The Grid 2: Blueprint for a New Computing Infrastructure". Morgan Kaufmann, 2004.

[Gi07] Gil, Y. et al.: "Examining the Challenges of Scientific Workflows". IEEE Computer, 40(12), 2007.

[IBM] IBM: WebSphere Enterprise Service Bus. [Online] http://servicemix.apache.org/home.html [February 17th, 2010]

[Ka06] Karastoyanova, D.: "Enhancing Flexibility and Reusability of Web Service Flows through Parameterization". PhD thesis, TU Darmstadt and University of Stuttgart, 2006.

[LR00] Leymann, F.; Roller, D.: "Production Workflow: Concepts and Techniques". Prentice Hall, 2000.

[Lu09] Ludaescher, B. et al.: "Scientific Workflows: Business as Usual?" In: *7th International Conference on Business Process Management (BPM)*, 2009.

[Mi09] Mietzner, R. et al.: "Business Grid: Combining Web Services and the Grid". In: *Transactions on Petri Nets and Other Models of Concurrency (ToPNoC)*, Special Issue on Concurrency in Process-aware Information Systems, Springer Verlag, 2009.

[Oi06] Oinn, T. et al.: "Taverna: Lessons in Creating a Workflow Environment for the Life Sciences". In: *Concurrency and Computation: Practice and Experience*, 2006.

[Ora] Oracle: Oracle BPEL process manager. [Online] http://www.oracle.com/appserver/bpel_home.html [February 17th, 2010]

[RD98] Reichert, M.; Dadam, P.: "ADEPT$_{flex}$ – Supporting Dynamic Changes of Workflows Without Losing Control". In: *Journal of Intelligent Information Systems*, 10(2), 1998.

[Sl06] Slomiski, A.: "On using BPEL extensibility to implement OGSI and WSRF Grid workflows". In: *Concurrency and Computation: Practice and Experience*, vol. 18 (10), pages 1229-1241, 2006.

[So10] Sonntag, M. et al.: "Process Space-Based Scientific Workflow Enactment". In: *Special Issue on Scientific Workflows in the International Journal of Business Process Integration and Management (IJBPIM)*, Inderscience Publishers, 2010. (to appear)

[Wa07] Wassermann, B. et al.: "Sedna: A BPEL-based Environment for Visual Scientific Workflow Modeling". In: Taylor et al. (Eds.): *Workflows for e-Science: Scientific Workflows for Grids*, Springer, 2007.

[Wa09] Wassink, I. et al.: "Designing workflows on the fly using e-BioFlow". In: *7th Conference on Service Computing (ICSOC)*, 2009.

[We05] Weerawarana, S. et al.: "Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More". Prentice Hall, 2005.