

# Scientific workflow management: between generality and applicability

Zhiming Zhao Adam Belloum Adianto Wibisono Frank Terpstra Piter T. de Boer  
Peter Sloot Bob Hertzberger  
Informatics Institute, University of Amsterdam  
Kruislaan 403, 1098SJ, Amsterdam, the Netherlands  
{zhiming|adam|wibisono|ftrpstra|ptdeboer|sloot|bob}@science.uva.nl

## Abstract

*In a Problem Solving Environment (PSE), a Scientific Workflow Management System (SWMS) provides a meta environment for managing activities and data in scientific experiments, for prototyping experimental computing systems and for orchestrating the runtime system behaviour. The realisation of a SWMS is often driven by domain specific applications and thus is at application level. Investigating the common characteristics in domain specific SWMSs and encapsulating them in a generic framework improve the reusability of the SWMS components and reduce the costs for introducing an e-Science framework in a new science domain. In this position paper, we present our research in an ongoing project: Virtual Laboratory for e-Science (VL-e). In the VL-e project, we are building a generic e-Science framework which will support scientists from different domains to share their knowledge and to perform specific experiments. We summarise the lessons we have learned from a previous VL-e implementation, and discuss the plan for improving the quality of the SWMS support in the VL-e framework.*<sup>1</sup>

**Key words:** Grid, Problem Solving Environments, Scientific Workflow, Software Engineering.

## 1 Introduction

Grid environments break down barriers to communication and interaction, and make valuable resources accessible among groups of trusted users (called Virtual Organisations (VO)). Such environments enable global distributed collaborations involving large numbers of people and large scale resources, and make data and computing intensive sci-

<sup>1</sup>This work was carried out in the context of the Virtual Laboratory for e-Science project ([www.vl-e.nl](http://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).

entific experiments feasible. Benefiting the Grid infrastructure, a new paradigm for scientific research: e-Science [21] emerges.

One of the important themes in e-Science is to develop effective Grid enabled Problem Solving Environments (PSE) for different science domains. Organising software utilities, e.g., simulators, visualisation and data analysis tools, as a meta experimental environment, a PSE allows a scientist to plan and conduct his experiments at high level of abstraction [16]. An important PSE guise is a Scientific Workflow Management System (SWMS) [25]. A SWMS explicitly models the dependencies between experiment processes, and orchestrates the runtime behaviour of involved resources according to a flow description. Data flow, control flow or Petri net based mechanisms have been used to model scientific workflows [22, 28].

In the past decade, scientific workflows and their support environments have attracted a great interest in the research context of e-Science and Grid enabled PSEs [4, 11, 22]. Generic services contributed by early Grid projects for encapsulating and binding computing and software resources form a reusable framework for realising components in a SWMS. However, most of the SWMS are historically driven by applications in specific domains, e.g., bio informatics [17, 31], high energy physics [6], and astronomical observations [1]. Investigating the common characteristics in these domain specific systems and implement them as part of a generic framework emerges as an important need for e-Science applications. 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 costs and risks for utilising an e-Science infrastructure in a new problem domain. Virtual Laboratory for e-Science (VL-e) [34] is an e-Science research project in the Netherlands; it aims to realise a Grid enabled generic 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

[3], a Grid enabled PSE framework developed in a previous project<sup>2</sup>, 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. In this position paper, we will discuss the feasibility and challenges in including scientific workflow support as part of generic e-Science services in the context of VL-e.

This paper is organised as follows. First, we analyse the basic issues in realising a SWMS and distinguish the main quality issues and challenges. And then, we briefly describe the current implementation of the generic VL-e framework and discuss its shortcomings. After that, we give our agenda on improving the quality of the current implementation, and present some ongoing work.

## 2 Scientific workflow management systems

The concept of *workflow* was originally invented in the business domain as *the automation of a business process, in whole or parts, where documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules* [2]. Promoted by the progresses in business workflow systems and in Problem Solving Environments, and in particular by the rapid development in Grid environments, a scientific workflow management system provides a meta environment for managing activities and data in scientific experiments, for prototyping experimental computing systems and for orchestrating the runtime system behaviour.

### 2.1 Scientific workflow

Due to the diversity between the scientific disciplines, scientists from different domains view a scientific workflow differently, e.g., data streams between experiment instruments and the analysis tools are modelled as a workflow in high energy physics applications [9], while human involved adaptation in predefined imaging processing are highlighted in medical imaging applications [19]. Scientific workflows are thus often embodied as different guises in the domain specific SWMS implementations, e.g., data pipelines based integration among components [22], runtime activity orchestration of interactive simulation [39] and composition of service based resources [26]. To investigate the common characteristics among these systems, we need an abstract view on different levels of aspects.

In our view, a scientific workflow orchestrates activities in the scale of an entire scientific experiment, not only the tasks being automated by the system, but also the routines

which do not occur in computers but are performed by human users. In this scale, a SWMS provides solutions to issues at three different levels:

1. *Experiment modelling.* Scenarios in an experiment need to be modelled using a formal mechanism.
2. *Workflow building.* A scientist can compose a workflow and validate the description using the semantics of the flow model.
3. *Runtime execution.* An execution engine is needed to instantiate a workflow and to schedule its execution.

The development of a SWMS is highly interdisciplinary, not only the modelling of the interaction dependencies in a flow is difficult, but also the realisation of the runtime control between components is time consuming. To extract the common characteristics from SWMSs of different domains, additional requirements on the implementation are demanded. We will discuss them in the next section.

### 2.2 Quality issues and challenges

In principle, a SWMS can be utilised in another suitable domain at different level: the flow modelling mechanism, the composition environment, and the resources orchestration mechanism. However, in practice, the solutions to these issues in one domain rarely consider the requirements for another, in particular the user support for composing and controlling flow is specialised for domain specific experiments, which hamper the reusability of a SWMS. For instance, a data flow based experiment planning environment in Higher Energy Physics does not directly support human centred flow execution paradigm demanded by some bio informatics experiments. To be effective in a generic e-Science framework, a scientific workflow management system needs to fulfil general quality criteria apart from meeting the specific requirements from the application domain. We highlight some of these generic quality requirements as follows:

1. The modelling mechanism of workflow needs to be flexible and generic. The mechanism is not domain specific. It necessarily captures the dynamics in the runtime environment and can be used to describe the basic scientific experiment scenarios.
2. A synthetic and friendly environment is preferable in a SWMS. Apart from building and executing a workflow, a scientist can also customise the configuration of the environment so that the scientist can map the interface elements onto the concepts he is familiar with in the specific domain.

<sup>2</sup>Virtual Laboratory II.

3. An execution engine utilises the state of the art Grid middleware for binding resources, and provides solutions to integrate with engines on different middleware and to benefit early domain or platform specific implementations. At runtime, the engine can efficiently execute a workflow using the underlying resources and fulfil the basic performance requirement.
4. User support is desired in a SWMS to provide assistance at different levels: e.g., composition and runtime control.

From the implementation point of view, the middleware employed in a SWMS to control underlying resources is basically provided by Grid environments, and sharable among all Grid enabled PSEs. We thus argue that the essential issues distinguish a domain specific SWMS from the others are its flow modelling mechanisms and the interface for a user to build a workflow and to control its runtime execution. Understanding the characteristics of domain specific experiments is crucial to abstract common flow patterns and demanded user support from them.

### 3 Virtual Laboratory for e-Science and the VLAM-G environment

#### 3.1 Background

The Virtual Laboratory for e-Science (VL-e) project aims to realise a Grid enabled generic framework where scientists from different domains can share their knowledge and resources, and perform domain specific research. Currently, there are six domains included: food informatics, medical diagnosis and imaging, bio-diversity, bio-informatics, high energy physics, and tele-science. One of the core ideas is to identify the common characteristics of scientific experiments in different domains and abstract the support for these common issues into a shared framework. VLAM-G, a Grid enabled PSE framework developed in the previous project of VL-e, is currently used as the first prototype of the shared framework. Fig. 1 shows the basic architecture.

The VLAM-G environment provides generic services for managing data and software resources, and for supporting scientific workflow. Using the VLAM-G environment, a user is allowed to perform his experiment location independently and to utilise Grid resources in his experiment transparently. Apart from that, the VLAM-G framework also integrates the information management services with the lifecycle of a scientific experiment [23].

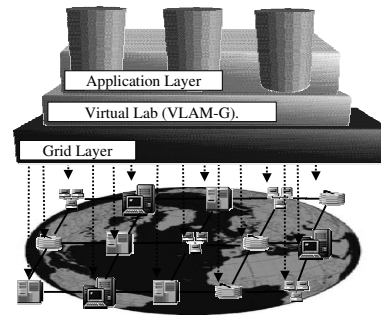


Figure 1. A conceptual vision of the VL-e project.

#### 3.2 Experiment processes and resource topology

In VLAM-G, the processes, both activities based on regular lab instruments and computing tasks<sup>3</sup>, of an experiment are explicitly modelled using three types of elements: *physical entities* which are the instruments to be used, *activities* to be performed by the scientists, and *data elements* which are the input/output of the activities. An experiment is modelled in three levels: abstract descriptions of the dependencies between experiment processes called *Process Flow Templates* (PFT)-, instantiations of a template called *Studies*-, and finally the actual flow of data analysis and computing activities in a *Study* called an *Experiment Topology*.

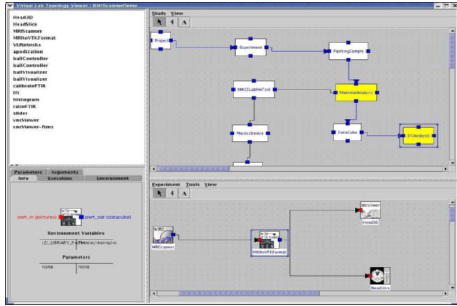
Using the GUI provided by the VLAM-G environment, a domain expert defines PFTs, and a scientist instantiates a PFT as a *Study*. The activity steps give the details of actual manual laboratory steps in the experiment or the computing tasks. A set of self-contained software entities, called *modules*, for performing computing tasks are described as an *Experiment Topology*. The VLAM-G environment provides a GUI for a scientist to describe a topology and to execute it via an engine (also called a Run Time System (RTS)). Fig. 2 shows a screen snapshot of the VLAM-G environment.

At run time, a topology can be executed on the underlying Grid infrastructure via a resource manager in VLAM-G. A scientist can have two types of interactions with the experiment instance: tuning the parameters of a specific component via the VLAM-G interface, or manipulating a specific module when it is interact-able. A database infrastructure is employed to manage both the static and runtime information.

#### 3.3 Lessons learned

The VLAM-G environment has been used by users from a number of different application domains. During the past

<sup>3</sup>Such as simulation, visualisation and data analysis.



**Figure 2. The interface of VLAM-G for composing PFT and Topology.**

year, we learned a lot from the experiences of the previous implementation. In this section, we briefly summarise the lessons learned from the perspectives of flow modelling and software engineering.

### 3.3.1 Flow modelling

In the VL-e project, both the development team of the generic framework and the scientists from different domains realise the key role that a SWMS plays in an e-Science environment. The domain scientists in the VL-e community have described their requirements on the workflow support in a detailed wish list [36], however the current VLAM-G framework only adopts the most generic model to describe the constraints between software components: data dependencies. This model is straightforward for the data stream based experiments, but provides limited support for the dynamic issues in the flow control.

In the current VL-e framework, human activity based interaction dependencies are not explicitly modelled in the current PFT model. A user can not steer the execution of an experiment at run time. Apart from it, the execution control of the flow is not explicit; when a flow instance is submitted, the relation between modules (components) is fixed. The user can not modify the content of a topology when it is running. Finally, a topology can be executed on the Grid in a transparent way, however limited strategies are provided to adapt the execution according to the availability of the resources and the characteristics of the problem.

Because of these shortcomings, a scientist has to develop control intelligence inside the components when he wants to realise a complex scenario for his experiment, which hampers the reusability of both components and the experiment template. The key issue is that the current VLAM-G has limited capability for modelling complex workflow and for supporting the related runtime issues. An enhanced flow model and a flexible engine are needed.

### 3.3.2 Engineering perspective

Apart from the conceptual modelling of scientific workflow, the actual quality of the software construction is another issue we have learned from the current implementation. The current prototype of VLAM-G environment was mostly built in the research context of the project; the software was not carefully designed and implemented as industrial packages normally do. Crafting scientific codes into a big distributed environment is bug prone. The amount of effort demanded by debugging and maintaining the basic environment critically hampers the domain scientists to utilise them in practical cases.

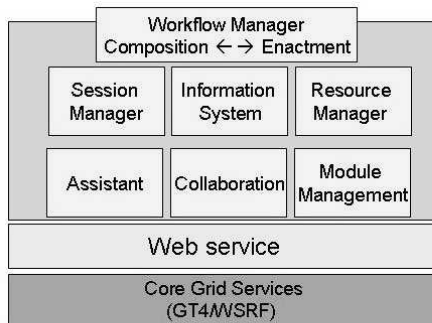
In the current implementation, functional components, e.g., the GUI management, flow composition and execution control, are tightly coupled. The user interface has limited adaptability, which does not meet the requirement, as we have discussed above, of *a reconfigurable user interface which can be customised for presenting concepts of different domain specific experiments*. Tight coupling also results in another shortcoming of the current implementation: the limited reusability of the system components. Since the development of the SWMS is bound to a data flow based mechanism, adding new features to control the runtime flow execution requires an almost complete redesign.

These lessons have been learned by the VL-e management team. A separate team has been established for controlling software quality and for bringing industrial software development experience to the VL-e framework development.

## 4 Improving the quality of the VL-e framework

To achieve this goal the VL-e framework has to provide a flexible and generic workflow management system to support defining complex scientific experiments and a well defined development and integration model, which reduce the effort to develop or port an existing application to the VL-e environment

The functional description presented in Fig. 3 shows seven main components: Workflow Manger (WM), the AS-Sistant (ASS), the Session Manager (SM), the Resource Manager (RM), and the Module Manager (MM). The workflow management is the main interface to the VL-e generic services; it offers the appropriate abstract level to the VL-e end-users. Aiming at a collaborative environment where different communities of end-users can cooperatively share and publish their results, the VL-e workflow management service extends a number of basic services and takes the support for collaborative work into account. The Session Manager (SM) is the key service for establishing the VL-e collaborative system. The SM controls all the end users'



**Figure 3. New design of the VL-e framework.**

activities within a single session, the SM keeps track of the list of end-users collaborating in a session, their respective roles, and allow cross sessions communication<sup>4</sup>. The basic collaboration mechanisms are provided by the collaboration Manager (CM) which controls and coordinates the interaction of VL-e end-users within and cross active sessions. A first prototype of this component have been already implemented as a Grid service using Globus toolkit GT3 [30]. The VL-e shared resource and data are under the control of Information Manager (IM) which stores all needed the information about the VL-e experiments. The IM service is used by all the other components to store and efficiently retrieve information about a given application, The IM service provide mechanisms for data view management and federated databases [23]. The Resource Manager (RM) maps the requested task on the available Grid resources. The RM communicates directly with the Grid middleware to collect information about the available resources, define the best schedule to optimize the execution of the application on the given resources, distribute the tasks on according to the schedule, and monitors the execution; more details on this components are given in [35]. The VL-e framework is an evolving working environment where experiments and best practices are carefully annotated and saved in order to be reused by the Assistant service which will assist the VL-e end-users at both design time and run time.

In the rest of this section, we discuss how we plan to improve the quality of Workflow Manager from both flow modelling and software engineering perspectives.

#### 4.1 From data flow to Petri net

In the new design, the data flow based mechanism will be improved. The dynamics of user interaction, e.g., runtime steering on decision points, will be described in the model. Because of the well-established theoretical framework and more importantly the suitability for representing

<sup>4</sup>In the context of the VL-e a session refers to an active scientific experiment

the common flow patterns [33], a Petri Net based approach is employed to extend the current flow description.

Place transition graphs are a sort of automata that can handle relations between conditions and the occurrence of events [13]. Basically, a place transition (PT) graph consists of three parts: a finite set of *places*, a finite set of *transitions*, and a finite set of relation links between places and transitions. In our earlier work, we implemented a PT graph based mechanism called *scenario net* [39] to model the interaction constraints between components in an interactive simulation system. A scenario net models the interactions using two types of dependencies between the roles' activities. The concurrency dependencies between them constitute the first type, which are represented as the relation links between places and transitions, and the tokens of the places. The second type of the dependencies is the specific conditions for each action, which are represented as the control expressions in the pre-set of the transition and in the links between the transition and its pre-set. The execution of a transition updates the state of the scenario net. The basic rules of the PT graph handle the concurrency dependencies between the activities by changing the marking of the net, and the execution rules for the place expressions update the control conditions for the activities.

In [38], we have demonstrated the application of scenario nets in modelling scenarios in interactive simulation systems. As the first step, we will include the *scenario net* model as a new *topology* description mechanism to enhance a PFT.

#### 4.2 Application scientists in the loop

The generic framework, we are aiming at in this project, is driven by the VL-e applications and the technology push of the Grid technology. Nowadays, Grid technology is shifting towards a service oriented paradigm, which has to be considered when designing the VL-e framework in order to take advantage of the recent developments in the Grid area. The VL-e workflow has to support both the application developers and end-users scientists. The requirements of these categories of users is different, while the application developers request a clear and well defined interface to develop grid-based applications, the end-users require a user friendly interface which will help them speedup the cycles of their research.

Including application scientists in the design loop of the VL-e framework is essential to synchronisation their visions on the e-Science framework and thus reduce the implementation risks of the framework. We use a series of synchronisation activities, e.g., questionnaires, face-to-face meetings, and web based feature request, to bring the feedback from application scientists in the system design.

### 4.3 Software engineering

We follow general software engineering disciplines to develop the new design. Component based approach will be the main methodology to encapsulate the system functional units and to craft them into a large and flexible system. The integration between software resources and the flow execution engine will be based on a standard interface between Grid middleware and domain specific components. Using this layer, domain experts can build their software components without being requested for knowledge of the underlying Grid details, and more important the realisation of the resources can be migrated to new Grid middleware by wrapping components with a new interface.

In the next section, we use a sample scenario which is from one of the domain applications to describe the desired functionality of the new SWMS.

### 4.4 A sample scenario

In this section a scenario is given of an existing case study concerning traffic prediction. We will briefly describe the case study and how it will benefit from using a scientific workflow within our future version of the VL-e. The scenario we use involves predicting (car) traffic based on data that consists of measurements with one minute granularity over the period of one month. In the scenario a workflow will be constructed and performed in which 24 hour predictions are made from this data. The scientist who constructs and performs the workflow is not a specialist in the predictive method employed. He will have to find a method to make the predictions and adapt his data to be suitable both for this method and his goal of 24 hour predictions, employ the right methodology for the predictive method and find the parameters for optimal performance. The end result should be a system continuously producing a 24 hour ahead forecast of the traffic situation.

While constructing his workflow the scientist will benefit from advice provided by the VL-e assistant. The assistant uses meta data which the scientist provides for his data combined with existing meta data for components in the VL-e. First he is advised which of the existing generic software components within the VL-e are most suitable for constructing a predictive system. Generic software components are pieces of software that have been generalized so they can be used for different applications within the VL-e. Within the scenario a generic software component for data assimilation is used, see [18] for details on predicting traffic using data assimilation. Secondly the assistant provides advice on the correct methodology. The Data Assimilation component can be seen as a toolkit and contains several PFTs (see section 3.1) which define the methodology associated with individual tools. The assistance can be informa-

tion on similar previous experiments, but also an explanation of the methodology being used. Furthermore through the collaboration manager the scientist can easily communicate and perform work on the traffic prediction system with colleagues at different physical locations. The move from data flow to a Petri net based approach to modelling will allow the scientist to tune his predictive system while it is running. Finally the resource manager allows the scientist to easily exploit grid resources, for instance for performing a parameter sweep on specific executable steps within a Data Assimilation PFT, performing multiple instances of the same step in parallel with different parameters.

## 5 Discussion and summary

In this paper, we reported our work on scientific workflow in the VL-e project. We discussed the necessity to move reusable functionality of a SWMS in a generic framework for different domain applications. We first described our visions on scientific workflow and then introduced the research context of the VL-e project. After discussing the lessons we learned from the previous implementation, we proposed a plan for improving the quality of the current VL-e framework.

### 5.1 Related work

We compare the related SWMS projects with our current prototype and future design from the perspectives of modelling, composition, user support, and runtime control of scientific workflows.

A commonly used approach to model the dependencies among computing tasks in scientific workflows is to use a Directed Acyclic Graph (DAG). Prominent grid projects using DAGs are the Pegasus system from GriPhyn project [15], Taverna [27], GrADS [12], GridFlow [8] and GridBus [7]. Scripting languages are another approach, such as Matrix from SDSC [37], GridAnt [5] and JPython in XCat [24], and BPEL4WS (Business process Execution Language for Web Services) [10]. Both approaches have shortcomings. Scripting languages require scientists to have programming skills, and the DAG approach does not directly support loop and bidirectional coupling in the model. Our current prototype uses a DAG approach while for our future design we will extend this with the possibility of iterations and bidirectional connections. It should be noted that our approach (both in current and future versions) is unique in that not only the executable steps are modeled, but also the non executable steps performed by humans (see section 3.2).

In current systems, the user interface and support for composing workflow are mostly domain specific. To compose a relatively small workflow, a graphical user interface,

e.g., in [29] or [27] is preferable. Although our current prototype is not domain specific the user interface is still relatively simple. However, a graphical composing interface is not suitable to manage the layout of a very large and complex workflow. Improved solutions include using hierarchical representations, e.g., in Triana [32] or automating the composition procedure, e.g., in Pegasus [15]. For our future design we will adopt a hierarchical approach.

Workflow enactment in grid environment involves finding appropriate resources and scheduling the execution of workflow tasks. A workflow can be scheduled using a centralized or decentralized method. In [8] a global workflow manager schedules sub-workflows to resources of different organizations via their local resource schedulers, while in a decentralized approach the computing tasks are scheduled by distributed workflow managers as in Triana [32]. Several scientific management such as GridAnt [5] and UNICORE [29] systems use a full-plan-ahead mapping strategy, meaning that all tasks are statically mapped to the resources before the execution. [14] improved the dynamic support by using a just-in-time approach in which the mapping is performed whenever resources are available. The latter dynamic approach can adapt better to the changes in grid environment (although computation to perform resource reallocation might create additional overhead) while the former full-plan-ahead approach can optimize the plan based on the entire structure of the workflow. A better understanding of application behaviour and resource model is necessary [14]. Our current prototype uses centralized static scheduling, however for future design we plan to use peer to peer technology to make scheduling dynamic and distributed.

## 5.2 Summary

In the VL-e project, the research on domain specific applications and the development of a generic e-Science framework are dependent, but they have to be carried out in parallel. The domain scientists need a flexible e-Science framework to pursue their investigation on the domain specific problems, but on the other hand, the developers of the e-Science framework expect the domain scientists to provide clear requirements on the framework. In a project as large as VL-e, it is always challenging to synchronise the visions among different sub programs. To reduce the risks, we highlighted the importance for including domain scientists in the development loop of the generic e-Science framework.

In the paper, we did not argue that the scientific workflow support for different domains can be fully realised at a single generic level. We follow an incremental methodology in the development. On the one hand we continue with the current prototype and add new features which are requested

by the domain scientists for supporting new domains, and make the development in an *evolutionary* manner. On the other hand we also track the requirements from all the application domains and propose new design of the framework when a *revolutionary* construction is necessary.

As one of the fast growing areas in Grid computing, scientific workflow systems have been utilised in more and more application domains. Reusing the existing products and providing generic solutions to realise the interoperability among them is also a consideration in the VL-e framework. In this way, the state of the art scientific workflow results can be aggregated into the structure of PFT.

## 6 Future work

The VL-e project has just passed its first year. The application scientists have made remarkable progresses in their specific domain, which helps generic framework team to understand the problems in the specific domains and to improve the implementation of the VL-e framework. A lesson learned from VLAM-G project is that scientists will not choose a novel architecture simply because it looks beautiful unless it can work with the existing ones and provide exciting new features [20]. We are also particularly looking at the user support at different levels of a SWMS and proposing a mapping mechanism between the user elements and the concepts of the specific domain experiments.

## References

- [1] The astrogrid project homepage. In <http://www.astrogrid.org/>, 2005.
- [2] Workflow management coalition. In <http://www.wfmc.org/>, 2005.
- [3] H. Afsarmanesh, R. Belleman, A. Belloum, A. Benabdelkader, 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.
- [4] G. Allen, W. Benger, T. Goodale, H.-C. Hege, G. Lanfermann, A. Merzky, T. Radke, E. Seidel, and J. Shalf. The Cactus code: A problem solving environment for the Grid. In *Proceedings of the Ninth IEEE International Symposium on High Performance Distributed Computing (HPDC'00)*, page 253. IEEE Computer Society, 2000.
- [5] K. Amin, G. von Laszewski, M. Hategan, N. J. Zaluzec, S. Hampton, and A. Rossi. Gridant: A client-controllable grid workflow system. In *HICSS*, 2004.
- [6] 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.
- [7] R. Buyya. Grid economy comes of age: Emerging grid-bus tools for service-oriented cluster and grid computing. In *Peer-to-Peer Computing*, page 13, 2002.

- [8] J. Cao, S. A. Jarvis, S. Saini, and G. R. Nudd. Gridflow: Workflow management for grid computing. In *IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 198–205, May 2003.
- [9] H. Casanova. Distributed computing research issues in grid computing. *ACM SIGACT News*, 33(3):50–70, 2002.
- [10] K.-M. Chao, M. Younas, N. Griffiths, I. Awan, R. Anane, and C.-F. Tsai. Analysis of grid service composition with *bpel4ws*. In *AINA '04: Proceedings of the 18th International Conference on Advanced Information Networking and Applications Volume 2*, page 284, Washington, DC, USA, 2004. IEEE Computer Society.
- [11] G. Chin, Jr., L. R. Leung, K. Schuchardt, and D. Gracio. New paradigms in problem solving environments for scientific computing. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 39–46. ACM Press, 2002.
- [12] K. Cooper, A. Dasgupta, K. Kennedy, C. Koelbel, A. Mandal, and et al. New grid scheduling and rescheduling methods in the grads project. In *IPDPS Next Generation Software Program - NSFNGS - PI Workshop*, 2004.
- [13] L. Czaja. Place/transition Petri net evolutions: recording ways, analysis and synthesis. *Fundam. Inf.*, 51(1):43–58, 2002.
- [14] E. Deelman, J. Blythe, Y. Gil, and C. Kesselman. Workflow management in griphyn. *Journal of Grid Computing*, 2003.
- [15] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, K. Blackburn, A. Lazzarini, A. Arbre, R. Cavanaugh, and S. Koranda. Mapping abstract complex workflows onto grid environments. *Journal of Grid Computing*, 1(1):25–39, 2003.
- [16] J. R. R. Efstratios Gallopoulos, Elias Houstis. Computer as thinker doer: Problem-solving environments for computational science. *IEEE Computational Science and Engineering*, 2:13–23, 1994.
- [17] M. Ellisman, M. Brady, D. Hart, F.-P. Lin, M. Muller, and L. Smarr. The emerging role of biogrids. *Commun. ACM*, 47(11):52–57, 2004.
- [18] G. R. M. Frank P. Terpstra, Arnoud Visser. Intelligent adaptive traffic forecasting system using data assimilation for use in traveler information systems. In *The Symposium on Professional Practice in AI a stream within the First IFIP Conference on Artificial Intelligence Applications and Innovations AIAI-2004*, Toulouse France, August 2004.
- [19] L. Hassell and J. 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.
- [20] L. O. Hertzberger. Introduction of VLAM-G and VL-E. In *Internal seminar*, 2004.
- [21] T. Hey and A. E. Trefethen. The UK e-science core programme and the grid. *Future Generation Computer System*, 18(8):1017–1031, 2002.
- [22] C. Johnson, S. Parker, and D. Weinstein. Large-scale computational science applications using the SCIRun problem solving environment. In *Proceedings of Supercomputer*, 2000.
- [23] E. C. Karletas. *Scientific information management in collaborative experimentation environments*. PhD thesis, Universiteit van Amsterdam, Amsterdam, NL, (Promoter, Prof. Dr. L. O. Hertzberger), 2004.
- [24] S. Krishnan, R. Bramley, D. Gannon, M. Govindaraju, R. Indurkar, A. Slominski, B. Temko, J. Alameda, R. Alkire, T. Drews, and E. Webb. The xcat science portal. In *SC*, page 49, 2001.
- [25] R. McClatchey and G. Vossen. Workshop on workflow management in scientific and engineering applications report. *SIGMOD Rec.*, 26(4):49–53, 1997.
- [26] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics Journal.*, online, June 16, 2004.
- [27] T. M. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, R. M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, 2004.
- [28] M. Peleg, I. Yeh, and R. Altman. Modelling Biological Processes using Workflow and Petri Net Models. *Bioinformatics*, 18(6):825–837, 2002.
- [29] M. Romberg. The unicore grid infrastructure. *Scientific Programming*, 10(2):149–157, 2002.
- [30] A. B. S. de Ridder and L. Hertzberger. Grid service based collaboration for vl-e: Requirements, analysis and design. In *European Grid Conference*, 2005.
- [31] R. D. Stevens, A. J. Robinson, and C. A. Goble. mygrid: personalised bioinformatics on the information grid. In *ISMB (Supplement of Bioinformatics)*, pages 302–304, 2003.
- [32] I. J. Taylor, M. S. Shields, I. Wang, and O. F. Rana. Triana applications within grid computing and peer to peer environments. *J. Grid Comput.*, 1(2):199–217, 2003.
- [33] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distributed and Parallel Databases*, 14(3):5–51, 2003.
- [34] VL-e. Virtual laboratory for e-science. In <http://www.vl-e.nl/>, 2005.
- [35] K. Vladimir, A. Belloum, and B. Hertzberger. VL-e: Approach to design a grid-based virtual laboratory. In *DAPSYS, 5th Austrian-Hungarian workshop on Distributed and Parallel systems*, 2004.
- [36] VLAM-G team. VLAM-G and workflow support wish list. In <http://www.vl-e.nl/>, 2005.
- [37] J. Weinberg, A. Jagatheesan, A. Ding, M. Faerman, and Y. Hu. Gridflow description, query, and execution at sec using the sdsc matrix. In *HPDC*, pages 262–263, 2004.
- [38] 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.
- [39] Z. Zhao, D. van Albada, and P. Sloot. Agent based flow control for hla components. *Simulation transaction, Special Issue: Agent directed simulation. (to appear)*, 2005.