Collaborative e-Science Experiments and Scientific Workflows

Recent advances in Internet and grid technologies have greatly enhanced scientific experiments' life cycle. In addition to compute- and data-intensive tasks, large-scale collaborations involving geographically distributed scientists and e-infrastructure are now possible. Scientific workflows, which encode the logic of experiments, are becoming valuable resources. Sharing these resources and letting scientists worldwide work together on one experiment is essential for promoting knowledge transfer and speeding up the development of scientific experiments. Here, the authors discuss the challenges involved in supporting collaborative e-Science experiments and propose support for different phases of the scientific experimentation life cycle.

Adam Belloum, Marcia A. Inda, Dmitry Vasunin, Vladimir Korkhov, Zhiming Zhao, Han Rauwerda, and Timo M. Breit
University of Amsterdam

Marian Bubak
AGH University of Science and Technology, Poland

Luis O. Hertzberger
University of Amsterdam

Complex scientific experiments involve distributed data and computing resources and require collaboration among scientists with various backgrounds. Recently, workflows have become a popular approach to modeling and organizing such experiments.[1]

A *scientific workflow management system* explicitly models the dependencies between processes within an experiment and orchestrates resources' runtime behavior. Over the past few years, various research groups have developed workflow management systems.[2–5] Successful scientific workflows are evolving into commodity tools that we can use to build and run more complex and challenging applications. A new generation of social networking and sharing sites has emerged, one mindful of scientists' specific needs – a good example being the myExperiment site (www.myexperiment.org), which makes it easy to find, use, and share scientific workflows and build communities.[6] This sharing model is flexible enough to support various aspects of a scientific workflow's life cycle.

In his article "Science 2.0," Ben Shneiderman argues that we must expand traditional scientific methods to deal with the complex issues that arise as social systems meet technological innovation.[7] This vision extends the e-Science concept, introduced a decade ago, which focuses mainly on computationally intensive science

1089-7801/11/$26.00 © 2011 IEEE Published by the IEEE Computer Society

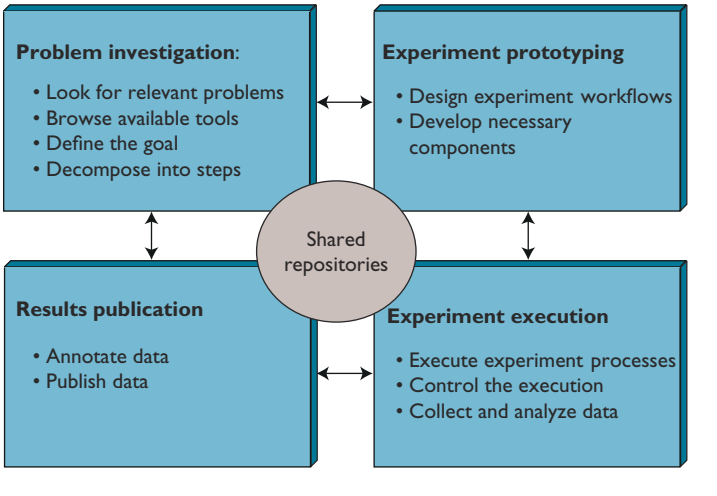

Figure 1. A scientific experiment life cycle. Such a life cycle has four phases: problem investigation, experiment prototyping, experiment execution, and publication of results.

and how to tackle it using highly distributed environments. Complex scientific experiments require not only access to geographically distributed data and computer resources but also methods for sharing and transferring the acquired knowledge among scientists. Grids and clouds, being virtual infrastructures that support data- and compute-intensive experiments, provide solutions to the inherent scaling problem.

However, these technologies offer little to boost the sharing of experiments or derive new findings by integrating various data sources, such as publications and scientific results across different scientific domains.

Here, we describe the functional life cycle of an e-Science experiment and present a suite of tools developed to support it. We also discuss various requirements for collaborative scientific experiments. We use the Dutch Virtual Laboratory for e-Science (VL-e) as an example to discuss how we can develop and integrate this functionality.

Collaborative Experimentation

Setting up scientific experiments involves various activities performed at different times.[8,9] These activities belong to the experiment's life cycle, which can be divided into four phases: problem investigation, experiment prototyping, experiment execution, and results publication (see Figure 1). Each phase requires support for collaboration and interactions. Integrating the information produced throughout this life cycle can be challenging, and it becomes even more so if we consider that scientists are continuously defining hypotheses, collecting data, running experiments, revising hypotheses, and publishing results. Multidisciplinary and geographically distributed teams of scientists should be able to locate, construct, execute, and maintain such scientific experiments. Thus, we look at support for collaborative experimentation from three angles: information sharing, communication, and coordination.

Problem Investigation

Before designing a new experiment, scientists study related research and identify existing methods and tools that they might reuse in the experiment. At this stage, scientists have two main requirements: access to existing knowledge within a scientific field — that is, scientific publications, books, software components, and so on — and the possibility of interaction with peers to brainstorm and share live documents. When performing these activities, scientists create new valuable knowledge, which should be recorded and shared. Support for collaboratively investigating a problem must address two main challenges: information integration (in heterogeneous formats) and access to collaborative tools.

Experiment Prototyping

Prototyping a new experiment is an iterative process in which scientists also require access to the knowledge and expertise developed within the scientific community. This knowledge is an important asset and brings great value to new application design, letting scientists focus on developing new components. In e-Science, using workflows helps scientists abstract the complex infrastructure and promotes knowledge transfer.

Experiment Execution

Experiment execution focuses on scaling the prototypes designed in the previous phase. Scientific workflows are data-centric and might have special requirements for both computing and network resources. Co-allocation of such resources is needed to enact workflows across multiple computing resources. In practice, executing such workflows includes staging

Virtual Laboratory for e-Science Tools Uptake

Various projects and institutions outside the Netherlands have adopted the Virtual Laboratory for e-Science (VL-e) generic tools described in the main text. In practice, two tools have been used as a framework for integrating and exposing virtual laboratory services. The Workflow System Virtual Laboratory Abstract Machine (WS-VLAM) is a tool for application developers that integrates services such as semantic annotation, data provenance, and matchmaking for e-Science resources. The biomedical engineering group at the Technical University of Eindhoven used WS-VLAM to help conduct a global sensitivity analysis of a blood pressure wave propagation model in arteries.[1] Munich Technical University used it as a platform for developing a new framework to facilitate the execution, monitoring, and management of computational science engineering (CSE) simulations in computational grids.[2]

VBrowser, on the other hand, is a framework for users; it offers an intuitive graphical interface and a high level of abstraction of the underlying e-Science infrastructure. Researchers from communities such as HealthGrid, the Enabling Grids for e-Science (EGEE) biomed community, GateLab, and D-Grid use VBrowser routinely in daily activities.[3]

References

1. C.A.D. Leguy et al., "Estimation of Distributed Arterial Mechanical Properties Using a Wave Propagation Model in a Reverse Way," *Medical Engineering Physics*, vol. 32, no. 9, 2010, pp. 957–967.
2. E. Elts et al., "Grid Workflows for Molecular Simulations in Chemical Industry," *High Performance Computing in Science and Eng.*, part 7, SpringerLink, 2010, pp. 651–662.
3. S. Olabarriaga et al., "A Virtual Laboratory for Medical Image Analysis," *IEEE Trans. Information Technology in Biomedicine*, vol. 14, no. 4, 2010, pp. 979–985.

the components that make up the experiment, efficiently executing them on available computing resources, and monitoring the experiment's progress. Interactive steering of execution is sometimes required to let scientists steer the execution path and tune component parameters.

Results Publication

We can view the results obtained and methods developed in an experiment as contributions to society. Any publication of these results should contain a clear description of the experiment steps, execution conditions, input data, interactions required to control the execution, and results analysis. Publishing all this information in public repositories requires careful annotation. Environments that let scientists annotate data and compose documents jointly are key components in supporting large-scale, multicenter scientific experiments.

Results published in papers don't easily lend themselves to verification and reuse. To shift focus from secondary sources (that is, publications) to actual data, algorithms, and workflows used in scientific research, we need a unified, collaborative framework. The open science movement has pushed toward developing such a framework,[10] while initiatives such as Object Reuse and Exchange (www.openarchives.org/ore) and open repositories such as those in the Directory of Open Access Repositories (OpenDOAR; www.opendoar.org) are addressing the challenges related to building it.

Such frameworks can enable users of various high-performance computing infrastructures, data repositories, and virtual laboratories to publish and directly reference their data and workflows.

Virtual Laboratory for e-Science

The VL-e project aims to provide generic tools that support a wide class of specific e-Science application environments and establish an experimental infrastructure for evaluating this vision (see the "Virtual Laboratory for e-Science Tools Uptake" sidebar). Different types of users — including domain scientists, domain application developers, generic component developers, framework developers, testing and certification engineers, and infrastructure administrators — will likely interact within the targeted framework.

The VL-e framework helps scientists develop CPU and data-intensive applications, lets them use a distributed and complex computing infrastructure, and improves sharing across multidisciplinary scientific domains. The VL-e architecture follows a service-oriented approach. Its main components are either simple Web services or Web Services Resource Framework (WSRF)-compliant services. This approach has a major benefit — that is, the virtualization of a complex and distributed computing and storage infrastructure. It allows for building a loosely coupled system, leading to a highly dynamic architecture, which is desirable in a dynamic environment such as the grid.

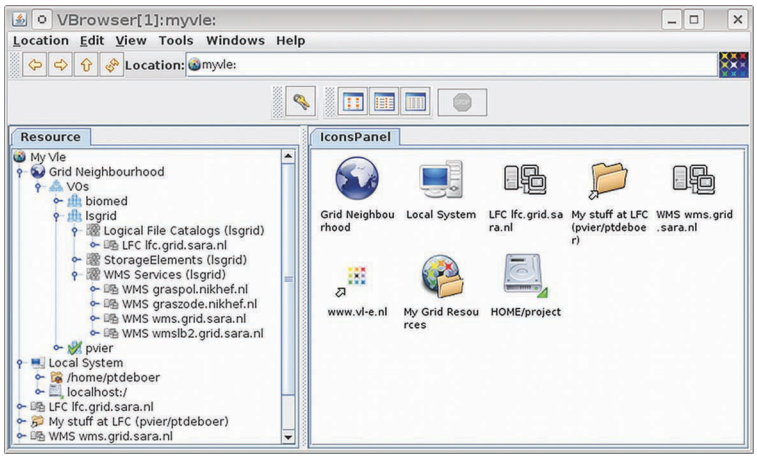

Figure 2. Virtual Browser screenshot. Here, we can see various storage systems, including Storage Resource Broker (SRB), Storage Resource Management (SRM), Grid-FTP, and the local file system.

Scientific Process Flow Template

Nowadays, collaboration among multidisciplinary teams located at geographically dispersed sites is common. We use the concept of the *process flow template* (PFT) to represent and formalize scientific workflows and facilitate information exchange in large collaborations. In our model, scientists can design experiments by instantiating such PFTs. Defined by application domain experts, PFTs capture scientists' expertise and serve as guidelines for junior researchers.[11] PFTs contain both laboratory activities and computing-based tasks, such as simulation and data processing.

Once a scientist creates and saves a PFT, that scientist or any member of the project team or virtual organization can view and extend it. Users can link together multiple versions of a given PFT via the provenance system, which gives precise information about a given PFT's evolution and who contributed to each version.

PFT customization typically involves tuning parameters, specifying new input data sets, and modifying or attaching a new application workflow. PFT use encourages best practices by providing access to domain experts' views of a given experiment's flow; by using PFTs, end users will perform the right tasks at the right time.

Mapping these abstract descriptions onto concrete experiment descriptions requires a detailed pool of knowledge. The following toolset, however, can support scientists in managing experiment templates and workflow components.

Olingo is a tool for mapping PFTs onto an underlying data representation.[12] It's available as a Web portal (the Olingo Web Tool [OWT]) or as a plug-in for the ontology editor Protégé 2 (http://protege.stanford.edu). Olingo employs an ontology-based approach that provides extensible mechanisms typically used to describe the domain of disclosure with human-readable text, enabling common understanding among scientists and software applications.

Semantic Annotation of Workflows (SAW) was designed as part of the Workflow System Virtual Laboratory Abstract Machine (WS-VLAM) Workflow Composer. It lets scientists semantically annotate to workflows and workflow components. It also generates the corresponding OWL-S profile, which describes the workflow components and workflows as a function of three basic types of information: the workflow component provider, the function the workflow component performs, and a set of features that specify some workflow characteristics.

The *H*ybrid-B*a*sed *M*atch*m*aker for *e*-Science *R*esources (Hammer) is a Web application for searching shared resources such as PFTs, workflows, and workflow components (http://elab.science.uva.nl:8081/ws-hammer). Hammer performs profile-based input and output matching. The matching algorithm can combine logic-based semantic matching with syntactic token-based similarity metrics.

Virtual Resource Browser

The Virtual Browser (VBrowser; www.vl-e.nl/vbrowser) offers scientists a graphical environment in which they can interactively access various types of resources to manipulate data (upload, download, search, annotate, and view), start applications (prepare and execute experiments), and monitor resources (status, control, and notification), as Figure 2 shows. VBrowser supports core grid file systems such as Grid-Enabled File Transfer (Grid-FTP), the Secure File Transfer Protocol (SSH-FTP), Storage Resource Management (SRM), Logical File Catalog (LFC), and Storage Resource Broker (SRB) out of the box, and presents them to the user in an intuitive treelike structure.[13] The VBrowser controls access to files based on users' virtual organization access rights.

Scientific Workflow Management

WS-VLAM is a workflow management system that coordinates the execution of distributed

grid-enabled software components. It was developed in accordance with the Open Grid Services Architecture (OGSA) WSRF standards. Because WS-VLAM's engine is a stateful Web service that supports runtime interaction with ongoing experiments, multiple users can follow the execution of a collective experience simultaneously and view any graphical output the ongoing experiment generates. A simple locking mechanism lets users taking part in the experiment customize it at runtime; the modification is reflected to all experiment participants.[14]

WS-VLAM provides a composition editor that complements the PFT by supporting an experiment's computer-executable phase. We can use this editor for two main tasks: composition and monitoring. It supports developing workflow components in several programming and scripting languages (Java, C++, R, and Python), allows access to Web services, establishes job farming and parameter sweep requirements, and is platform-independent. Once a scientist submits the workflow for execution, any other scientists participating in the experiment can monitor the ongoing execution. Because WS-VLAM is a stateful service, it provides a roaming capability that lets scientists connect from any Internet-enabled location. Monitoring workflow execution allows for following each particular atomic (or composite) process's effects and outcomes.

WS-VLAM lets scientists use the grid in a seamless way; they must deal only with the workflow graphical interface. The workflow engine is responsible for finding the appropriate resources, taking into consideration users' access rights (based on their virtual organization memberships) and software dependencies. Domain scientists construct or assemble workflows during the experiment-prototyping phase because they understand the semantics associated with the workflow components. In practice, these are the lab engineers and application developers, and they prepare workflows for end users who are either scientists using the workflow as a blueprint to develop further complex workflows or users who just want to rerun the workflow with different datasets and parameters.

Workflow Aggregation and Distributed Coordination

The diversity of application domains in the VL-e project makes it difficult to find a single workflow system with all the required features. Furthermore, each application domain has its own specific legacy tools and workflows that must be available to the entire community. We propose a workflow bus architecture (Virtual Laboratory Workflow Bus, or VL-WFBus) to aggregate workflows developed in different workflow management systems.

VL-WFBus provides an interface for wrapping and integrating legacy workflows. It also provides tools that recognize different workflow descriptions stored in the system and describe the meta information related to these workflows according to a predefined schema.

In a typical scenario, users can browse the legacy workflows stored in the system, select one, and execute it via the workflow bus, or customize it by viewing its contents and opening the respective legacy workflow system to adapt it.

Workflow Provenance

Data and workflow provenance are crucial for supporting e-Science experiments. Recently, the use of electronic lab notebooks (ELNs) by scientists has increased, because ELNs help researchers document experiments and procedures performed in laboratories. As a consequence, several ELN products have been developed (www.scientific-computing.com/scwjunjul06elns.html).

Workflow provenance goes one step further and helps trace back a workflow's execution, retrieving valuable information, which can help reproduce a successful workflow execution or discover problems that have led to a faulty one. Provenance systems must collect provenance data at each phase of a workflow life cycle. We distinguish between two types of data provenance: *system-level provenance*, related to the context in which the workflow has been executed and *application-level provenance*, related to the application logic.

WS-VLAM has a provenance service that collects provenance information at runtime and saves it in persistent storage. Each change in the status of the components and services composing the application workflow triggers an event. WS-VLAM then collects and structures the provenance events according to the Open Provenance Model (twiki.ipaw.info/bin/view/Challenge/WebHome).

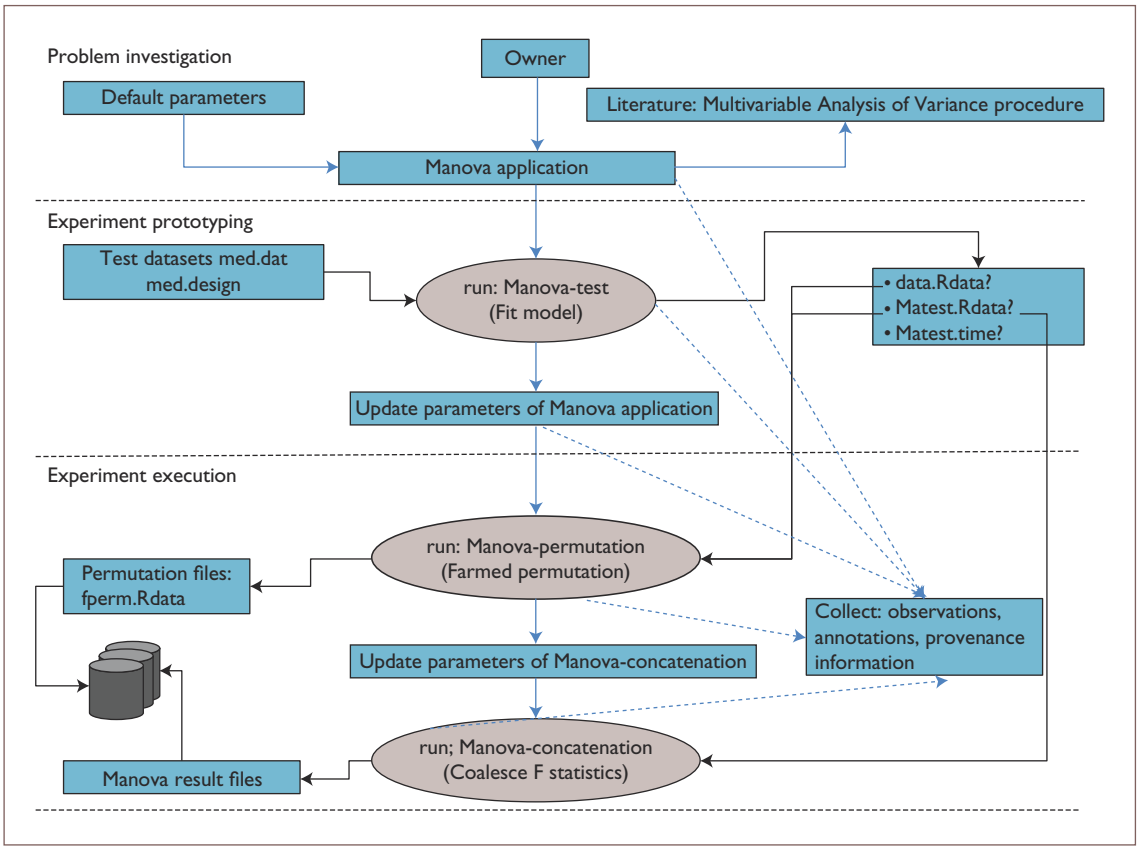

Figure 3. Process-flow template. We can see the different steps involved in designing the Affymetrix Permutation-Based Probe Level Estimation *(APPLE) application. Rectangles represent data and manual operation while ovals represent workflows.*

Application Use Case

To illustrate the e-Science life cycle, we describe the development phases of the *A*ffymetrix *P*ermutation-Based *P*robe *L*evel *E*stimation (APPLE) application, which has been modeled as a workflow that might require access to geographically distributed datasets and will generate data that scientists can publish and share within the bioinformatics community.

Problem Investigation Phase

Microarray experiments use statistical tests to look for genes that react differently to different circumstances — for example, finding which genes are activated in a cancerous cell. Owing to the arrays' costs, scientists frequently design microarray experiments on the Affymetrix platform (www.affymetrix.com) using relatively few biological or technical replicate arrays. This can result in low statistical power when inferring the number of differentially expressed genes. Researchers at the microarray department at the University of Amsterdam are investigating how to increase Affymetrix experiments' statistical power without increasing the number of replicate arrays.

In Affymetrix microarrays, specific genes and transcripts are reported using a set of short, DNA-like sequences or probes. The proposed approach skips the first step of compiling one expression value for each probe set, normally used in current statistical tests.[15] Instead, it uses individual probe values as input in a gene-specific mixed-effect analysis of variance (Anova) model to perform the statistical test. Figure 3 illustrates the PFT for the APPLE application.

The resulting statistical test contains a computationally intensive permutation test. This test's computation time increases with the number of arrays the experiment uses and the number of probes pertaining to a probe set. For large datasets containing large probe sets, this permutation test becomes too time-consuming to run on a single CPU or even in a cluster. Consequently, we designed a grid-enabled procedure with two levels of parallelization to carry out the necessary permutation test.

Experiment Prototyping Phase

We modeled a prototype of the three-step job-farm permutation level as three separate workflows that scientists can operate interactively and execute across grid-enabled resources. Experiment prototyping was carried out by domain scientists with specialized knowledge of the Affymetrix platform and statistical design. APPLE workflow execution is orchestrated either manually by one or more users or automatically via the VL-WFBus. Although manual orchestration is needed at early stages to tune, validate, and customize the APPLE workflow, the automated approach is more suitable for the production phase, when the workflow must execute numerous jobs concurrently.

Experiment Execution Phase

To perform computationally intensive permutation tests (1,000 permutations per gene), we moved the prototype to a grid-production environment, the Dutch e-Science Grid (www.biggrid.nl). At this stage, scientists use the VBrowser for browsing distributed storage resources to select the appropriate datasets, and submit the workflow to the WS-VLAM engine via the WS-VLAM viewer plug-in (see Figure 2). The integration of VBrowser and WS-VLAM hides the underlying infrastructure's complexity by automating several tasks, such as locating datasets and workflow components stored in shared repositories and deploying them at runtime on the available computing resource. In an ongoing experiment, we will perform the steps comprising the APPLE workflow using Taverna and Kepler, combining these steps with specific tools available in the two systems to create a more complex workflow (see Figure 4).

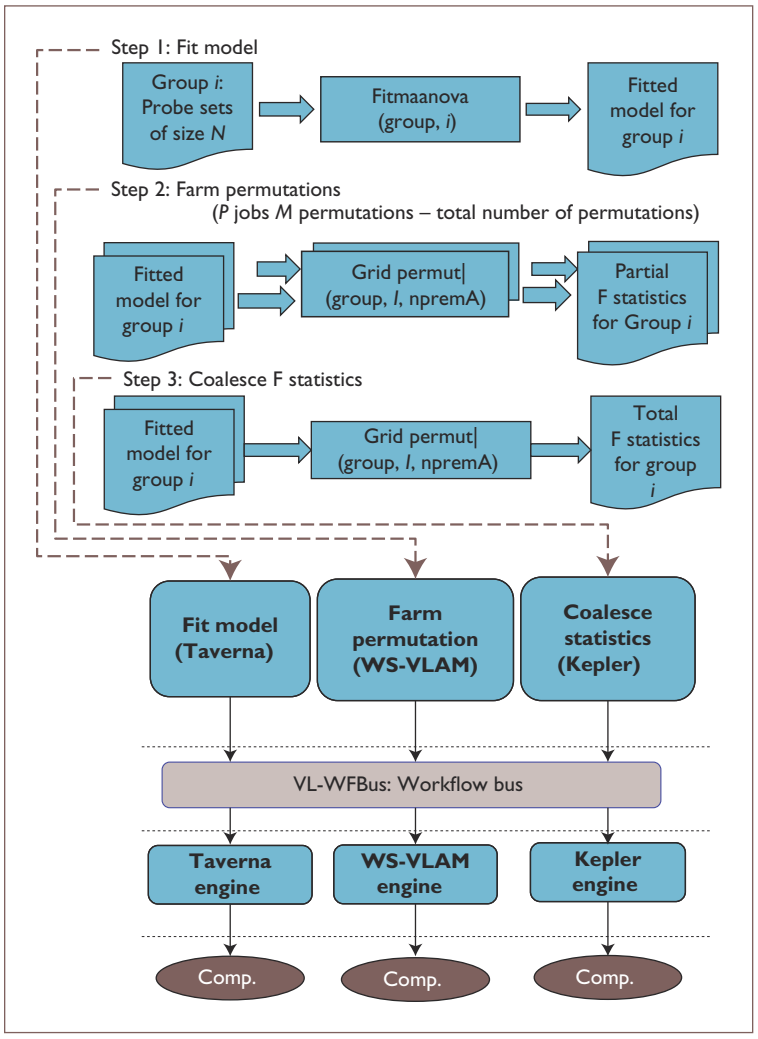

Figure 4. Prototyping the APPLE procedure. We have implemented each step at the job-farm permutation level as a stand-alone workflow that users can execute in a single run or in three separate ones. When the user executes the three steps at once, the Virtual Laboratory Workflow Bus (VL-WFBus) acts as a meta-level to orchestrate the execution of the three workflows independently.

Publication Results

Both the APPLE workflow and its newly developed components contain new knowledge that must be made available to other scientists through shared repositories such as Hammer or the myExperiment website (www.myexperiment.org/workflows/1211).

Our work shows the feasibility of the Science 2.0 vision Shneiderman proposes.[7] As we make clear, the technology for implementing this vision is available — however, it might be a while before the scientific community gets acquainted with such an environment and adopts it as an integral part of its research activities. To speed up the adoption of the e-Science environment described here, we're now adopting Web 2.0 approaches in which users have access to all software and storage facilities through a simple Internet browser.

Acknowledgments

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 (Besluit Subsidies Investeringen Kennisinfrastructuur) grant from the Dutch Ministry of Education, Culture, and Science and is part of the Information and Communications Technology (ICT)

Virtual Laboratory for e-Science and Related Work

The Virtual Laboratory for e-Science (VL-e) is one of many projects that aims to bridge the gap between a complex, distributed e-Science infrastructure and the scientific community, which comprises potential users for such an infrastructure but doesn't have the required technical background to use it. Scientists from various scientific domains — namely, food informatics, medical diagnosis and imaging, biodiversity, bioinformatics, high-energy physics, and telescience — helped specify the requirements and validation use cases involved in the design of the VL-e generic tools we describe in the main text.

Such a multidisciplinary working environment has given the VL-e approach a distinctive character compared to other, domain-specific virtual laboratory approaches. Some examples are MyGrid[1] and the Galaxy project (http://usegalaxy.org), which initially focused on managing and sharing experiments in biology. The Wide In Silico Docking for Malaria (Wisdom; http://wisdom.eu-egee.fr/), caBig (https://cabig.nci.nih.gov), and e-Health projects had the initial goal of developing a new approach for drug discovery, cancer research, and other health-related issues using recent achievements in information technology. The GridSpace virtual laboratory is a similar approach — developed within the ViroLab project — that specifies in silico experiments using scripts.[2]

Contrary to these projects, VL-e provides a solution to scalability issues with regard to both processing and data access; all VL-e tools offer native support for a number of grid and cloud system features, such as grid security, virtual organizations, and seamless access to grid-enabled storage and computing resources. Other distinguishing features include the workflow bus approach, which lets us create a workflow from other workflows developed in third-party systems such as Taverna, Triana, and Kepler; the ability to monitor and interact with ongoing experiments, putting the scientist in control; dynamic network control in distributed environments for data-intensive applications;[3] and on-demand resource acquisition for urgent computing using the elastic computing approach.[4]

References

1. P. Li et al., "Systematic Integration of Experimental Data and Models in Systems Biology," *BMC Bioinformatics*, vol. 11, article no. 582, 2010; doi:10.1186/1471-2105-11-582.
2. M. Bubak et al., "Virtual Laboratory for Collaborative Applications," *Handbook of Research on Computational Grid Technologies for Life Sciences, Biomedicine, and Healthcare*, IGI Global, 2009, pp. 531–551.
3. R. Strijkers et al., "Network Resource Control for Grid Workflow Management Systems," *Proc. 6th World Congress on Services*, IEEE CS Press, 2010, pp. 318–325.
4. R. Strijkers et al., "AMOS: Using the Cloud for On-Demand Execution of e-Science Applications," *IEEE e-Science Conf.*, IEEE CS Press, 2010, pp. 331–338.

innovation program of the Ministry of Economic Affairs (EZ). We thank Piter de Boer and Victor Guevara-Massis for their major contribution to the development of the VBrowser and Hammer tools.

References

1. Z. Zhao, A. Belloum, and M. Bubak, "Special Section on Workflow Systems and Applications in e-Science," *Future Generation Computer Systems*, vol. 25, no. 5, 2009, pp. 525–527.
2. B. Ludscher et al., "Scientific Workflow Management and the Kepler System," *Concurrency and Computation: Practice & Experience*, vol. 18, no. 10, 2006, pp. 1039–1065.
3. T. Oinn et al., "Taverna: A Tool for the Composition and Enactment of Bioinformatics Workflows," *Bioinformatics*, vol. 20, no. 17, 2004, pp. 3045–3054.
4. K. Lee et al., "Adaptive Workflow Processing and Execution in Pegasus," *Proc. 3rd Int'l Conf. Grid and Pervasive Computing Symposia/Workshops*, IEEE CS Press, 2008, pp. 99–106.
5. V. Korkhov et al., "VLAM-G: Interactive Dataflow Driven Engine for Grid-Enabled Resources," *Scientific Programming*, vol. 15, no. 3, 2007, pp. 173–188.
6. D. De Roure et al., "Towards Open Science: The myExperiment Approach," *Special Issue, Concurrency and Computation: Practice and Experience from the Microsoft eScience Workshop*, vol. 22, no. 17, 2010, pp. 2335–2353.
7. B. Shneiderman, "Science 2.0," *Science*, 7 Mar. 2008.
8. C. Wroe et al., "Recycling Workflows and Services through Discovery and Reuse," *Concurrency and Computation: Practice and Experience*, vol. 19, no. 2, 2007, pp. 181–194.
9. Y. Gil et al., "Examining the Challenges of Scientific Workflows," *Computer*, vol. 40, no. 12, 2007, pp. 24–32.
10. C. Wald, "Scientists Embrace Openness," *Science*, 9 Apr. 2010; http://sciencecareers.sciencemag.org/career_magazine/previous_issues/articles/2010_04_09/caredit.a1000036.
11. H. Afsarmanesh et al., "VLAM-G: A Grid-Based Virtual Laboratory," *Scientific Programming*, vol. 10, no. 2, 2002, pp. 173–181.
12. V. Guevara-Mass et al., "Semantic Workflow Discovery in VL-e," *Proc. Knowledge Grid Workshop Information Society* (IST) *2006 – Strategies for Leadership*, Elsevier, 2006; www.knowledgegrid.eu-project.info/know-ist.pdf.
13. T. Glatard et al., "fMRI Analysis on EGEE with the VBrowser and MOTEUR," *Proc. 3rd Enabling Grids for e-Science* (EGEE) *User Forum*, 2008; http://indico.cern.ch/contributionDisplay.py?contribId=65&sessionId=6&confId=22351.

14. A. Wibisono et al., "WS-VLAM: Towards a Scalable Workflow System on the Grid," *Proc. 16th IEEE Int'l Symp. High Performance Distributed Computing*, IEEE CS Press, 2007.
15. E. Hubbell, W.M. Liu, and R. Mei, "Robust Estimators for Expression Analysis," *Bioinformatics*, vol. 18, no. 12, 2002, pp. 1585–1592.

Adam Belloum is an assistant professor in the computer science department at the University of Amsterdam. He has a PhD in computer science from the Compiegne University of Technology, France. Contact him at a.s.z.belloum@uva.nl.

Marcia A. Inda is a senior scientist at Philips Research. At the time of this work, she was a post doctoral researcher in the Integrative Bioinformatics Unit of the Microarray Department at the University of Amsterdam. Inda has a PhD in mathematics from Utrecht University, the Netherlands. Contact her at marcia.inda@philips.com.

Dmitry Vasunin is a scientific programmer in the Virtual Laboratory project at the University of Amsterdam. He has an MSc in marine engineering of automatic control systems from Saint-Petersburg State Marine Technical University. Contact him at dvasunin@gmail.com.

Vladimir Korkhov is a postdoctoral researcher in the Academic Medical Center, University of Amsterdam. He has a PhD in computer science from the University of Amsterdam. Contact him at vkorkhov@amc.uva.com.

Zhiming Zhao is a researcher at the University of Amsterdam. His research interests include scientific workflow, distributed interactive simulation, and multi-agent systems. Zhao has a PhD in computer science from the University of Amsterdam. He's a member of IEEE. Contact him at z.zhao@uva.nl.

Han Rauwerda is a bioinformatician in the Microarray Department/Integrative Bioinformatics Unit at the University of Amsterdam. His research subject is transcriptome dynamics. Rauwerda has an MSc in environmental sciences from the Open University, Heerlen. Contact him at h.rauwerda@uva.nl.

Timo M. Breit is a group leader in the Microarray Department of the Swammerdam Institute for Life Sciences at the University of Amsterdam. Breit did his PhD studies on the molecular genetic mechanisms of gene rearrangement in lymphoid cells at the Erasmus University Rotterdam. Contact him at t.m.briet@uva.nl.

Marian Bubak is an adjunct in the Institute of Computer Science and ACC CYFRONET AGH University of Science and Technology, Poland, and a professor of distributed system engineering at the University of Amsterdam. He has a PhD in computer science from the AGH University of Science and Technology. Contact him at bubak@agh.edu.pl.

Luis O. Hertzberger is a retired professor of computer science at the University of Amsterdam. His research interests are in parallel computing, e-Science, and their application in industrial automation. Hertzberger has a PhD in experimental physics from the University of Amsterdam. Contact him at l.o.hertzberger@uva.nl.