Toward executable scientific publications

Strijkers, R.; Cushing, R.; Vasyunin, D.; de Laat, C.; Belloum, A.S.Z.; Meijer, R.

DOI
10.1016/j.procs.2011.04.074

Publication date
2011

Document Version
Final published version

Published in
Procedia Computer Science

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 4 (2011) 707–715

Procedia Computer Science

International Conference on Computational Science, ICCS 2011

Toward Executable Scientific Publications

Rudolf Strijkers[a,b], Reginald Cushing[a], Dmitry Vasyunin[a], Cees de Laat[a], Adam S.Z. Belloum[a], Robert Meijer[a,b]

[a]Informatics Institute, University of Amsterdam, The Netherlands
[b]TNO, Groningen, The Netherlands

Abstract

Reproducibility of experiments is considered as one of the main principles of the scientific method. Recent developments in data and computation intensive science, i.e. e-Science, and state of the art in Cloud computing provide the necessary components to preserve data sets and re-run code and software that create research data. The Executable Paper (EP) concept uses state of the art technology to include data sets, code, and software in the electronic publication such that readers can validate the presented results. In this paper we present how to advance current state of the art to preserve, data sets, code, and software that create research data, the basic components of an execution platform to preserve long term compatibility of EP, and we identify a number of issues and challenges in the realization of EP.

Keywords: Executable Papers, Workflows, Data Provenance, IaaS

1. Introduction

Research articles need to contain enough information to verify the methods and to reproduce the research data presented in a paper. Experiments should be described in such detail that researchers can reproduce the research results. Keeping detailed records and traces of the progress of an experiment increases the evidence that a procedure is correct. Though information technology is now indispensable in many disciplines of science, it is hardly used to improve the reproducibility of research data. In this paper, we investigate how information technology can be applied to reproduce experiments and to automatically collect traces while the experiment is executing, i.e. how to create the technologies for an Executable Paper (EP).

Nowadays, many papers are readable online using the Hyper Text Markup Language (HTML). HTML introduces formatted strings that indicate a reference to a resource known as hyper links, which can refer to sections of a document or to other documents. Hyper links give dynamism to a

doi:10.1016/j.procs.2011.04.074

static content and thus text can be read associatively by jumping from one hyper link to another. The core behind any HTML document is the interpreter, i.e. browser, which renders the HTML code and displays the content. More advanced interaction between user and document becomes possible when code or scripts are embedded in a HTML document, such as an online authoring environment [12].

A straightforward RP implementation embeds in the HTML document a link to where the data sets, code, and software can be found and a description of the experiment that created the research results. By clicking on a table or a graph, for example, the reader can use the embedded information to track how the research data was created. If the steps to create the research data are described accurately, unambiguously, and with enough detail, a computer program rather that the reader can re-run and validate research results. In this paper we show how state of the art e-Science tools can be applied to describe code, software, and parameters of scientific experiments. Furthermore, we present an architecture to realize RP.

However, describing how research data is created and linking to dependencies is not enough. In this paper we also address the platform to preserve RP dependencies and to re-run the experiment. We decompose platform issues into three problems. First, researchers use different methods to describe and run experiments, which might not be compatible or understandable by others. At least for RP it is necessary that an accepted method exists to describe and run experiments. We propose using workflows, which is a well established method in e-Science. Second, links to dependencies, such as large data sets, can break when a researcher moves affiliation and forgets to update old links. This problem is known in literature and solutions exist to store data for longer periods of time (Section 2). Third, The software and its dependencies once developed by a researcher may not work in modern or future systems, e.g. older libraries might be unavailable or a compiler might implement different optimizations. In our approach, the recent advance of Infrastructure as a Service Cloud computing [4] serves as a platform in which all the code and software dependencies can be encapsulated (Section 3).

We also present a use case in which we show that just a few components are missing to realize a proof of concept RP implementation (Section 4) and discuss some issues not addressed in our architecture (Section 5).

2. Background

Mathematica [22], a scientific computing environment integrates authoring of publications, code, and software to create research data into a single platform. Mathematica also includes a collection of *Computable Data*, selected data sets maintained by Wolfram, which can be accessed programmatically and used for experiments and model checking. Because such data sets can be large, it is better to reference the data than to include it in the notebook. Mathematica's approach is closely related to the RP concept, but users are locked into one platform to fully utilize its advantages. In the context of e-Science, we define an RP as a collection of static text, experiment descriptions, provenance, virtual resources, and datasets, which assist in reproducing research results.

It is often convenient to model data and computation intensive scientific experiments as a workflow [13, 16, 14] (Figure 1). Workflows are graphs where vertices describe a scientific process

while edges describe some dependency such as data or control. The popularity of MyExperiment [17] and scientific workflow management systems such as Kepler [3], Taverna [19], and Triana [27] are proof that workflow systems have become a commodity within the e-Science community. Workflow management systems help researchers to organize the interdependencies of processes/operations, and their dependencies, of which an experiment is composed into a well-formed description. The resulting workflow can be stored, shared, and executed by other researchers. Demonstrated by many real examples, workflows are suitable to describe a wide range of scientific experiments [11]. Therefore, in the EP model, to accurately and unambiguously describe a reproducible and executable experiment will be best implemented by workflows, which has three additional advantages: state-of-the-art in workflow management systems provide (1) execution environments supporting computational and data intensive experiments, (2) traceability of experiments, and (3) technology to implement short and long term compatibility.

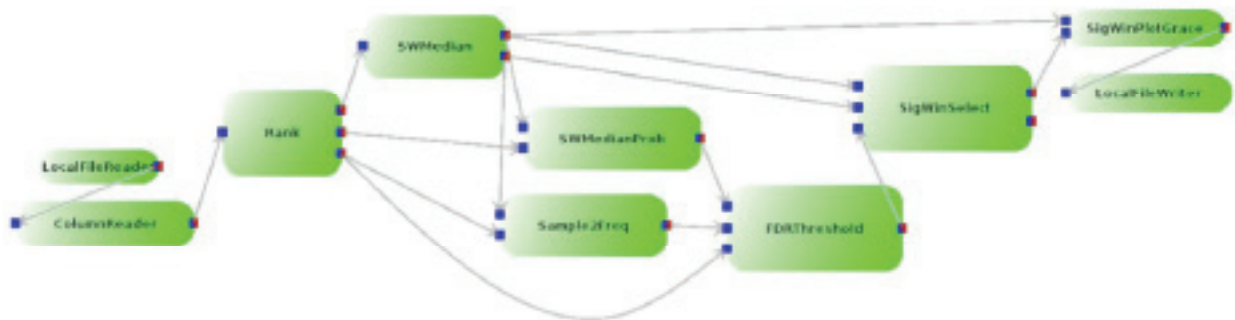

Figure 1: Example of a scientific experiment modeled as a workflow. This workflow is used to detect ridges in sequence of data sets such as Gene Expression sequence or Human transcriptome map.

Clearly, a workflow by itself is not enough to reproduce research data. An execution platform, such as Grids, is an additional requirement. When taking future compatibility into account, however, the level of detail to describe the execution platform of an experiment becomes an issue. For example, in order to run a distributed program, a number of Intel PC with the Linux Operating System, a number of system libraries of a specific version, input data in a specific format, etc., might be needed. Even if the level of detail to describe the experiment is enough, re-creating the execution platform for the experiment will be cumbersome. Additionally, the software to re-create has to be stored in case software distributions change. Two approaches to simplify the execution platform are possible: (1) provide abstractions for computation and storage, such as Grid architectures [10], or (2) encapsulate software details into Virtual Machines (VMs).

In 2008, we published and demonstrated a system to monitor and control programmable computer networks from Mathematica notebooks [24] and a multi touch table [25] . The presented system and the associated notebook were demonstrated at the Super Computing 2008 research exhibition in Austin, TX. Because the system had many software dependencies, we chose to save our software configuration in Virtual Machines (VM), i.e. the VM becomes a self-contained component with all the necessary software dependencies pre-loaded. Although the original infrastructure has been lost, the preserved VMs allows us re-execute the experiment as presented on Super Com-

puting 2008 on any infrastructure that supports VMs (with i386 architecture).

In Grid computing, we have extended workflow management system to support fine-grained provisioning of network services. With better control over Grid networks, it is easier to schedule workflows on geographically distributed computing resources [23]. Even if a single (large) super computer is unavailable, possibly a combination of various Grid resources will allow workflow execution. We implemented a Cloud computing approach to re-create Transient Grids (TGrids) on demand [26]. The UrbanFlood [28] project goes one step further: it will implement a framework for copying, adjusting, instantiating complete early warning systems, e.g. for dike failure, using workflows to describe the system. At the Super Computing 2010, research exhibition in New Orleans, LO, we demonstrated a prototype of a multi-cloud operating system that implements this vision.

Provenance plays an important role in the reproducibility and validation of an experiment. Provenance systems record how an experiment produced data. The de-facto standard for provenance models in e-Science has become the Open Provenance Model (OPM) [20] enables interoperable of provenance data between workflow management systems.

Some scientific experiments depend on potentially large input datasets. Since it is impractical or unfeasible to maintain local copies, large data sets can be provided by third parties, such as the annotated human genome data provided by Ensembl [1]. Recently also Cloud providers such as Amazon [15] host public datasets on their infrastructure. For smaller datasets, content delivery networks [21] provide intelligent data management, which hide the geographic location of data. On a request, parts or all of the dataset may be replicated at key locations on the network. With CDN, datasets are not statically referenced, but the CDN is responsible of providing the closest most recent datasets.

EP architecture not only provide experiment reproducibility, but the same architecture can also provide the basis for experiment long-term preservation. Projects such as CASPAR [6, 8, 7] and CAMILEON [5] specifically address digital preservation. CASPAR is based on the Open Archival Information Systems (OAIS) reference model [18] which is an initiative by NASA and created through necessity to organize decades of space data. OAIS is an abstract model for archiving digitally encoded data for long-term preservation. An example use of CASPAR is with European Space Agency (ESA) [2] satellite sensor data (Global Monitoring Experiment) where raw data gathered over the past years along with the processes that transform the data are preserved.

3. EP Architecture

The validation of the research results in an EP starts at the description of the experiments. For each research result in the EP, the experiment is described in a workflow (Figure 2(a)). Workflows can be stored separately from the paper or embedded in the document for interactive exploration of the methods. The *code* and *software* dependencies are encapsulated by one or more *virtual machines* (VM), which are created by the researcher. The *virtual machines* should be in such a state that the application can be started without installing additional software. Once created, the researcher will need to submit the VM to the *EP Data Store*.

To support coordinated application execution and automated data management, virtual machines need to be instrumented with a workflow agent *WFA*. *WFA* coordinates the execution and

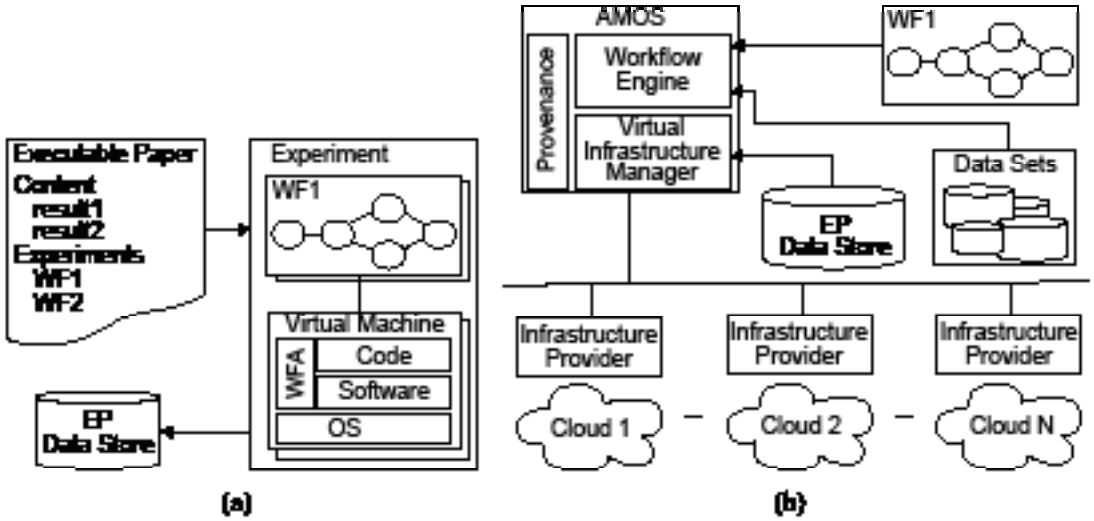

Figure 2: The architecture to describe (a) and to reproduce experiments (b).

management of remote processes, for example, to start and stop an application or to transfer input and output data. Moreover, in previous work we have shown that also infrastructure and related software can be automatically created and configured [26]. Our system, *AMOS*, uses a VM template with a minimal set of Grid tools installed, and implements a mechanism to start and configure VMs on-demand. We successfully demonstrated the automated creation of infrastructure and application execution of a real application using Grid technologies. *AMOS* demonstrates that complete Grid clusters can be saved in VM templates and can be recreated or cloned at a later stage to re-execute a workflow, i.e. a saved experiment.

Here, we make three additions to *AMOS* (Figure 2(b)) to make it suitable as a run-time environment for saved experiments. One, we generate *provenance* data in the execution of a workflow. Provenance data enables researchers to validate research results, but it can also be used to supply more detail to re-execute a workflow. For example, the workflow scheduler can use information about the order of execution of a previous experiment to re-execute tasks in the same order. Two, we add a *virtual infrastructure manager* which uses multiple *Cloud* resources to create the execution environment for the workflow components with the application templates in the *EP Data Store*. Currently, Amazon EC2 [15] is suitable for running experiments with less than 20 nodes, but we expect that in the future many Cloud *infrastructure providers* will be available to support larger experiments. Three, we use URLs to refer to data sets, which are needed for workflow execution. Though we use Grid technologies in AMOS to handle large data sets, a platform independent data referencing method is needed to support data management and annotation, such as in OpenDAP [9]. At run-time, the *workflow engine* resolves the references to the data sets.

We assume that applications use general purpose computers. Special purpose architectures, such as GPGPUs or FPGAs are currently not a commodity in Clouds. Though virtualization of special purpose computer architectures is possible, the resulting virtual machine might be slow and inefficient without hardware support. Therefore, some experiments depending on special purpose computer architectures might be unfeasible to execute in a Cloud. Our architecture, however,

is independent of the hardware platform supplied by the researcher. In the end, special purpose computer architectures require specific solutions. The *virtual infrastructure manager* could utilize Cloud providers that provide specific hardware architectures, for example.

4. Use Case

We describe, using a real scientific experiment as example, what is needed to build a proof of concept of the proposed EP architecture. In this section we focus on the principle elements of EPs identified in Section 2, namely: experiment description, provenance, and the virtualization of the infrastructure. The Affymetrix Permutation-based Probe Level Estimation (APPLE) application was developed by Micro-Array Department at the University of Amsterdam as an e-Science application following a workflow approach. Figure 3 shows the APPLE process flow template, which is used as a blueprint to describe the experiment in a way that allows the experiment to be re-execute with various parameters and data sets.

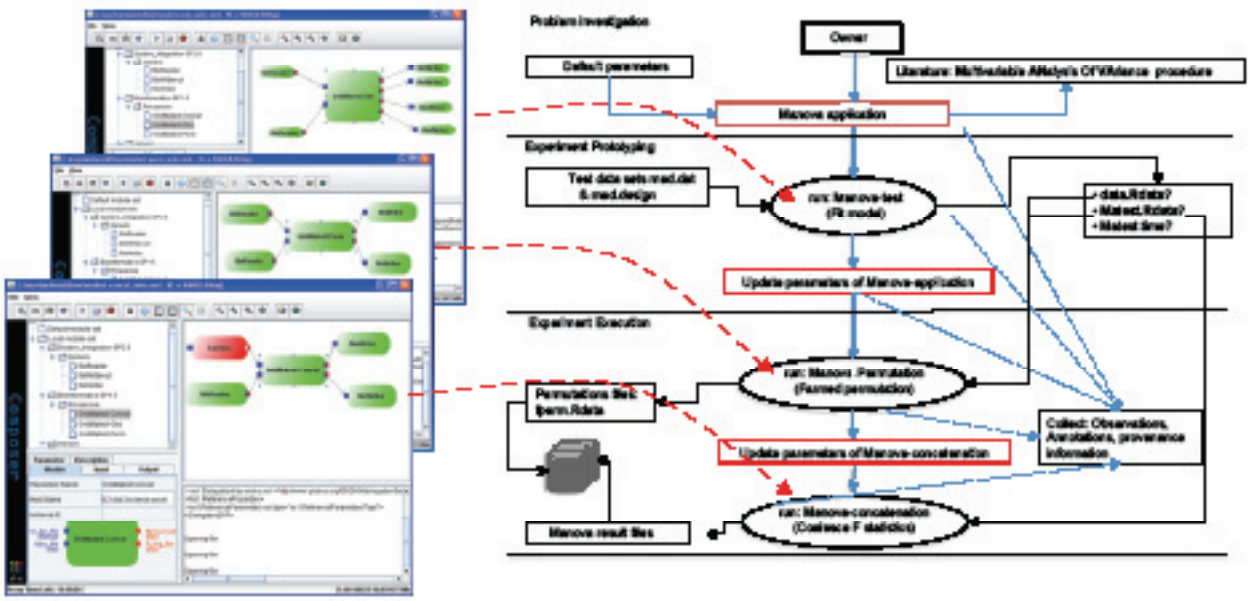

Figure 3: (right) Process flow template of the APPLE experiment. (Left) The executable workflow defined in a workflow management system

The first step is to convert the abstract presentation of the experiments (Figure 3) into executable workflows. A prototype of the three abstract steps namely the Fit model, permutations, and coalesce statistics were modeled as three separate workflows that can be operated interactively. We use WS-VLAM workflow management system to manage the execution of the three workflows across geographically distributed resources [14].

The workflow description contains the parameters, data sets, computing resources, and third party libraries necessary to execute a workflow. To track the workflow execution at run-time, a provenance tracking system part of WS-VLAM collects events while the workflows components are executed. These events are structured according to the Open Provenance model (OPM) [20] and saved.

To complete the cycle of an EP with research results from the APPLE experiment, we still miss one step: to embed the workflow aspects into a graphical user interface as part of the online EP. In the user interface, the reader triggers execution of the workflow by interacting with the EP. Then AMOS locates the appropriate VM template for the workflow, instantiating it, locates the data sets, and executes the workflow either from scratch or using some provenance data to re-create conditions.

5. Discussion

Today, authors of scientific papers follow to the instructions of the publishers regarding the format and the changes to be made to their paper before publication. This role places publishers in the leading position towards EP development, deployment, and standardization. Although the most of the building blocks of the proposed architecture are available, realizing EP is far from simply technical. Who will be responsible for maintaining data sets and VMs, and how feasible would it be to save all data associated with EP? Clearly, an EP infrastructure will implement tradeoffs between the responsibilities of authors, affiliations, and publishers.

Virtual machines provide the means to save code and software of EPs and provides at least short-term compatibility. The drawback is that virtualization by itself might not prove to be a long-term solution. Since future compatibility depends on the availability of virtualization software, maintaining the virtualization platform will be necessary. Would it then be feasible to augment the EP with enough meta-data to even re-create execution platform of the experiment in the absence of a virtualization infrastructure? To our knowledge no elegant solution exists for this problem, because a platform is always assumed.

Some experiments rely on commercial software. Therefore, EP platforms need to support some form of license management and tracking to allow experiment reproducibility. Here again it arguable who should provide the licenses to re-run the experiments. Moreover, the EP infrastructure provider effectively becomes a service provider, so it needs to build a trust relationship with the author. Even with trust, the EP infrastructure provider will need to implement mechanisms to prevent the author to submit programs that can cause (unintentional) harm.

Clearly, Cloud resource providers do not meet the demands of experiments that use *large systems*, i.e. top50 super computers. Often such experiments need specialized dedicated infrastructures. Using these resources, however, is costly. The mere costs of running the experiment might prevent researchers to validate the research results. In such scenarios we may have to look at different models to validate the research data.

6. Conclusion

In this paper, we investigated how information technology can be applied to reproduce experiments and to automatically collect traces while the experiment is executing, i.e. how to create the technologies for an Executable Paper (EP). We proposed an architecture that combines state of art in e-Science workflows, provenance, and infrastructure virtualization to implement EP. We also presented the use case of an experiment in bioinformatics, which we modeled as e-Science

workflow. The result is that the experiment can be automatically executed and with varying parameters. Moreover, by applying virtualization techniques, we were able to save the execution environment of workflows and restore it on-demand. Although some issues remain in the research and development of EP infrastructure, we expect that the biggest challenge in realizing EP will be non-technical.

References

[1] Ensembl. http://www.ensembl.org.

[2] Sergio Albani. The ESA approach to long-term data preservation using CASPAR. *ERCIM News*, 2010(80), 2010.

[3] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock. Kepler: an extensible system for design and execution of scientific workflows. *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*, pages 423–424, 2004.

[4] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.

[5] CAMILEON. Creative archiving at michigan and leeds: Emulating the old on the new. http://www2.si.umich.edu/CAMILEON.

[6] CASPAR. Cultural, artistic and scientific knowledge for preservation, access and retrieval. http://www.casparpreserves.eu.

[7] Esther Conway et al. Creating scientific research data for the long term: A preservation analysis method in context. *IPRES 2009: the Sixth International Conference on Preservation of Digital Objects*.

[8] Esther Conway et al. Preservation network models: Creating stable networks of information to ensure the long term use of scientific data. In *Proceedings of Ensuring Long-Term Preservation and Adding Value to Scientific and Technical Data*, 2009.

[9] P. Cornillon, J. Gallagher, and T. Sgouros. Opendap: Accessing data in a distributed, heterogeneous environment. *Data Science Journal*, 2:164–174, 2003.

[10] Ian T. Foster. The anatomy of the grid: Enabling scalable virtual organizations. In *Proc. First IEEE International Symposium on Cluster Computing and the Grid (1st CCGRID'01)*, pages 6–7, Brisbane, Australia, May 2001. IEEE Computer Society (Los Alamitos, CA).

[11] Carole A. Goble, Jiten Bhagat, Sergejs Aleksejevs, Don Cruickshank, Danius Michaelides, David Newman, Mark Borkum, Sean Bechhofer, Marco Roos, Peter Li, and David De Roure. myexperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic acids research*, 38(Web Server issue):W677–W682, July 2010.

[12] Google. Google docs, 2011. http//docs.google.com/.

[13] D Hull and M Pocock. Taverna a tool for building and running workflows of services, January 01 2006.

[14] Vladimir Korkhov, Dmitry Vasyunin, Adianto Wibisono, Victor Guevara-Masis, Adam Belloum, Cees de Laat, Pieter Adriaans, and L.O. Hertzberger. Wi-vlam: towards a scalable workflow system on the grid. In *WORKS '07: Proceedings of the 2nd workshop on Workflows in support of large-scale science*, pages 63–68, New York, NY, USA, 2007. ACM.

[15] Amazon Web Services LLC. Amazon elastic compute cloud, 2011. http://aws.amazon.com/ec2/.

[16] Bertram Ludäscher, Ilkay Altintas, Shawn Bowers, Julian Cummings, Terence Critchlow, Ewa Deelman, David De Roure, Juliana Freire, Carole Goble, Matthew Jones, Scott Klasky, Timothy McPhillips, Norbert Podhorszki, Claudio Silva, Ian Taylor, and Mladen Vouk. Scientific process automation and workflow management. In Arie Shoshani and Doron Rotem, editors, *Scientific Data Management*, Computational Science Series, chapter 13. Chapman & Hall, 2009.

[17] myExperiment. myexperiment. http://www.myexperiment.org.

[18] OAIS. Reference model for an open archival information system (oais). http://public.ccsds.org/publications/archive/650x0b1.PDF.

[19] T. Oinn, M. Greenwood, M. I. Addis, M. Nedim Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, D. J. Marvin, P. Li, P. Lord, M. R. Pocock, M. Senger, R. Stevens, A. Wipat, and C. Wroe. Taverna: Lessons in creating a workflow environment for the life sciences. *JOURNAL OF CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE*, 2002.

[20] OPM. Open provenance model (opm). "http://openprovenance.org".

[21] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content-addressable network. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '01, pages 161–172, New York, NY, USA, 2001. ACM.

[22] Wolfram Research. Mathematica. http://www.wolfram.com/mathematica.

[23] Rudolf Strijkers, Mihai Cristea, Vladimir Korkhov, Damian Marchal, Adam Belloum, Cees de Laat, and Robert Meijer. Network resource control for grid workflow management systems. *Services, IEEE Congress on*, 0:318–325, 2010.

[24] Rudolf Strijkers and Robert Meijer. Integrating networks with mathematica". 2008.

[25] Rudolf Strijkers, Laurence Muller, Mihai Cristea, Robert Belleman, Cees de Laat, Peter Sloot, and Robert Meijer. Interactive control over a programmable computer network using a multi-touch surface. In Gabrielle Allen, Jaroslaw Nabrzyski, Edward Seidel, Geert van Albada, Jack Dongarra, and Peter Sloot, editors, *Computational Science – ICCS 2009*, volume 5545 of *Lecture Notes in Computer Science*, pages 719–728. Springer Berlin / Heidelberg, 2009.

[26] Rudolf Strijkers, Willem Toorop, et al. Annie: Using the cloud for on-demand execution of e-science applications. In *Proceedings of the eScience2010*, pages 773–799, 2010.

[27] Ian Taylor. Triana generations. In *E-SCIENCE '06: Proceedings of the Second IEEE International Conference on e-Science and Grid Computing*, page 143, Washington, DC, USA, 2006. IEEE Computer Society.

[28] UrbanFlood. Urbanflood eu fp7 project, grant agreement no. 248767. http://urbanflood.eu.