Service Level Management for Executable Papers

Reginald Cushing[1], Spiros Koulouzis[1], Rudolf Strijkers[1,3], Adam S.Z. Belloum[1], and Marian Bubak[1,2]

[1] University of Amsterdam, Institute for Informatics, The Netherlands
[2] AGH University of Science and Technology, Department of Computer Science, Poland
[3] TNO Information and Communication Technology, The Netherlands

Abstract. Reproducibility of Science is considered as one of the main principles of the scientific method, and refers to the ability of an experiment to be accurately reproduced, by third person, in complex experiment every detail matters to ensure the correct reproducibility. In the context of the ICCS 2011, Elsevier organized the executable paper grand challenge a contest to improve the way scientific information is communicated and used. While during this contest the focus was on developing methods and technique to realize the idea of executable papers, in this paper we focus on the operational issues related to the creation a viable service with a predefined QoS.

1 Introduction

The idea of interactive paper is not new; the very first steps in this field were introduced by with HyperText Markup Language [1]. A reader of a web page was able to navigate from page to page by simply clicking on the link-associated with a certain concept. The technical details of the systems supporting HyperText Markup Language is rather complex, however, the way HyperText Markup Language are exposed to both the readers and writer of web pages is intuitive, for the reader its just a colour encoded text, while for the writer it is just a simple line of code with a very simple syntax. When applets and ECMAScript (http://en.wikipedia.org/wiki/ECMAScript) were introduced the concept of hypertext has been pushed further readers of web document were execute small applets and client-side scripts to run simple application. The Executable Paper (EP) Grand Challenge organized by the Elsevier in the context of International Conference on Computational Science (http://www.iccs-meeting.org/) is to push this concept one step further to include scientific publications. However, this is not a trivial transition as many scientific publications are about complex experiments, which are often computing and data intensive, or require special software and hardware. Propriety software used by experiments is also subject to strict licensing rules. The papers published in the grand challenge workshop propose various solutions to realize the executable paper concepts [6,7,8,9]. The papers focuses on the technical details and technology choices but give little attention

M. Alexander et al. (Eds.): Euro-Par 2011 Workshops, Part II, LNCS 7156, pp. 116–123, 2012.

to the operational aspects associated with the deployment of such a service and what would be the impact on the stakeholders to provide a reliable and scalable service allowing re-execution of published scientific.

The rest of the paper is organized as follows Section 2 describes the executable paper life cycle, Section 3, discussion the exploitation of executable papers, Section 4 describes the implementation of executable paper using Cloud approach, and Section 5 discusses SLM needed achieve a certain QoS.

2 Executable Papers Lifecycle

The concept of EPs is feasible only if the lifecycle governing this concept is clear and the role of the different actors is well defined through the entire lifecycle of the production of the executable paper. This lifecycle starts from the time authors decide to write the paper, going through the review process, and ending by the publication of the paper. The role of the authors, in the current publication cycle, finishes when the paper is accepted for publication. The publisher is the second actor, as he makes the paper available and accessible to potential readers. The third actor is not directly active in the creation of the paper but still very important as it provides the needed infrastructure to the author to perform the experiment to be included in the paper. The third actor is usually the institution to which the author belongs at the time he is writing his paper. After the publication of the paper, maintaining the infrastructure needed to reproduce scientific experiments is not the primary interest of research institutions. A very important question is then posed, which actor will take the role of providing the needed logistic to keep the EP alive. We believe that the publisher is the only actor that is capable to take over this task. However providing a service that allows a reader to re-run experiments is completely different from providing a service that just give access to a digital version of the paper. In this case the publisher will have to maintain a rather complex computing and storage infrastructure that might be beyond the scope of the publisher actual interests and expertise. Outsourcing this task to a specialized computing service provider might be a possible solution where Service Level Agreements (SLAs) play a vital role in maintaining an EP and re-running experiments in a timely fashion so as to maintain an acceptable reader experience. We will develop further this solution in the rest of the paper.

3 Exploitation of Executable Papers

Reproducibility of Science is considered as one of the main principles of the scientific method, and refers to the ability of an experiment to be accurately reproduced, by third person, in complex experiment every detail matters to ensure the correct reproducibility. Dissemination of the knowledge contained in scientific paper often requires details that be can hardly described in words and if added to the paper will make the paper more difficult to read.

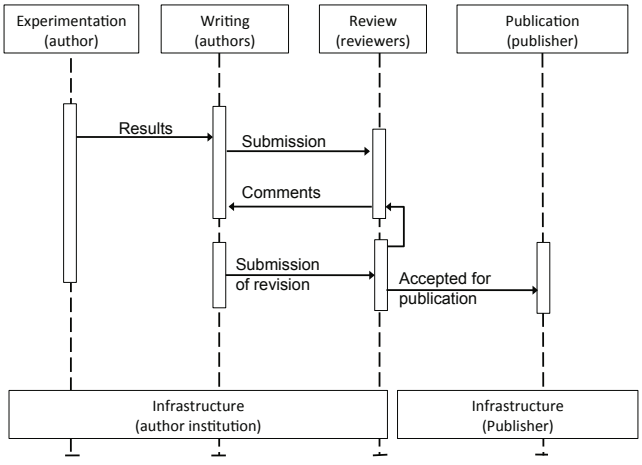

Fig. 1. Lifecycle of EP, experiment results trigger the writing of scientific papers, it is thus important that readers of these paper are able to explore and re-execute if needed these experiments

There are a couple of daily scenarios in science where the concept of executable paper is indeed needed. The first one is the review process of scientific publications, often reviewers selected by conference organizers and publishers to assess the quality of newly submitted papers have to verify the results published. For that, they need to trace back the path to initial data or to verify parameters used in a specific step of the proposed method and in certain cases even re-run part of the experiment.

The second most common scenario in an EP is while scientists are reading the already published paper. Often they are interested in reusing part of the published results whether these results are algorithms, methods or tools. Currently this is done by contacting the authors and try to get the needed information but often the authors are not reachable or their current research topics are different from the one published in the paper.

From these exploration scenarios, we can identify the actors active during the various phases of the lifecycle of the executable paper (Table 1).

With the emergence of reliable virtualization technologies, which are capable of hiding the intricacies of complex infrastructure, publishers can offer more than just a static access to scientific publications [5,4]. The reader of a published scientific publication should be able to re-execute part of the experiment. Figure 2 illustrates the interactions between various entities in the EP scenario. SLAs between readers and publisher exist which define a certain QoS expected by the reader such as maximum time for re-running experiments. Readers are often affiliated to institutions for which an SLA between the institution and the publisher could exist. The publisher manages a set of SLAs with service providers for outsourcing the re-execution of the experiment. Since experiments vary in complexity, the SLAs would define which provider is capable of executing the experiment within QoS parameters.

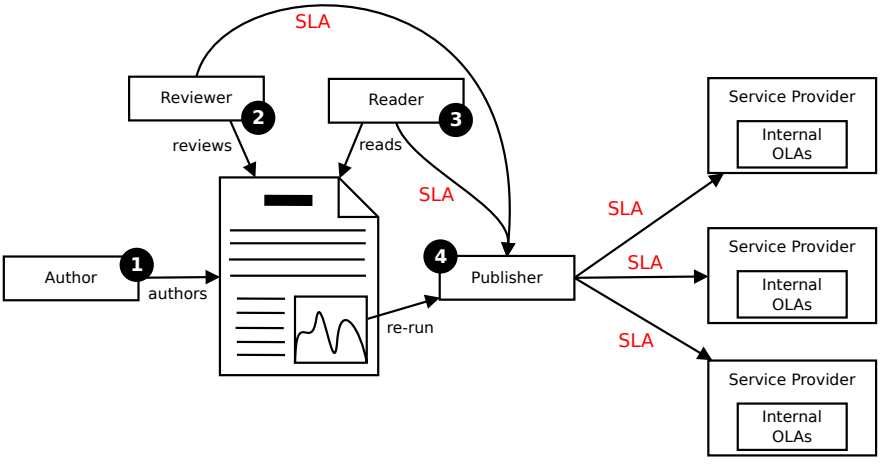

Fig. 2. Interaction of various entities during the lifecycle of an EP. In (1) the author creates an EP, (2) the reviewer reviews the paper and possibly re-run the experiment. (3) A reader that reads the EP after publication and can also re-run the experiment. (4) The publisher that upon request from the reviewer or reader can outsource the execution of the experiment. Depending on the SLA between reviewer, reader and the publisher, the publisher can choose amongst a set of SLA to pick the best service provider which can deliver the QoS requested by the reviewer or reader.

Table 1. Main Actors involved in the realization and executable paper lifecycle

Actor	Role	Active in phase
Actor 1: Scientist	author of a scientific publication	experimentation and writing
Actor 2: Scientist affiliation	provide the computing infrastructure	experimentation
Actor 3: Publisher	publishes scientific publications	review and publication
Actor 4: Reviewer	assess the quality of scientific publications	review
Actor 5: Provider	provide the computing infrastructure	publication
Actor 6: Scientist	any scientist who want to re-run experiments	publication (customer)

4 Challenges Facing the Implementation of Executable Papers

The implementation of the executable paper concept faces a number of challenging points of different nature: administrative, intellectual property and technical challenges.

- Administrative issues: are related to the role of the actor, which will provide the computing infrastructure to re-run part of the entire experiments. As we have pointed out is section 3, when the paper is published there is no guarantee, that infrastructure used to produce the results is available for re-runs.
- Intellectual property issues: most of scientific experiments are using third party software which is licensed to the institution of the author of the EP at a certain time and under certain conditions, which might change in time. In certain case even the data used in scientific experiment is subject to licensing, and privacy issues.
- Technical issues: are related to the environment in which the experiment has been performed, CPU architecture, operating system, and third party libraries.

The technical issues even if they might be in some case complex are still easier to solve as the virtual machine technology is nowadays able to create self-contained and reliable system platform which supports the execution of a complete operating system. Virtual images can be started on-demand to re-run a certain application, this approach is widely used in Cloud computing [10].

If the virtual machine approach can solve the problems of working environment and library dependencies, it still has some issues with IP issues, Jeff Jones explains in his blog how tracking software assets on virtual images is gaining momentum [2]. Even if it is possible to re-run experiment published in EP, there are still IP issues that need to be solved. Whoever will provide a service able to re-run published scientific experiments, has to acquire licenses for common software in a certain scientific domain that might partially solve the IP problem.

Publishers are the potential actors which are able to provide a service which implements the executable paper concept, using the virtualization technique, they can provide, without having to know the details of a given experiment, which is able re-run published experiment. To implement such a solution, publishers either has to develop in-house the expertise and the infrastructure needed to re-execute EP or to outsource the provisioning of the needed infrastructure to Cloud and Grid providers. Technically Cloud providers such Amazon, Microsoft, Google etc. are able to provide the need infrastructure against a fixed cost [3].

5 Discussions

Any solution for the EP has to be intuitive and should not add much further burden on the actors involved in the EP lifecycle. A number of tools and services

have to be developed to support all these actors in accomplishing their respective task. From the author point of view, the services needed are: a service for collecting provenance information when he/she is doing the experimentation, a service for creating annotation when writing the paper, and framework to create a virtual image of environment in which the experiment has been performed. From the reviewer point of view services are need to interact with the paper query details when needed, and re-run experiments.

The publisher, as a service provider, will play a key role in the realization of the EP. Currently publishers provide access to scientific papers, such a service has be extended to upload the created virtual images needed to re-run the published experiments.

Because in the proposed approach publishers will outsource the provisioning of the needed infrastructure to an independent service provider, a server level management is needed. In case of EP the usage of the resources may vary a lot from a couple computing nodes to a much more larger infrastructure. Publishers may offer a whole spectrum of EP categories covering a wide spectrum of features from fast and immediate to slow or scheduled at later time. Publisher acts as a composite service provider whereby they integrate externally provided services at run-time into end-to-end composite services.

From the provider point of view (publisher) each service can be provided by different infrastructure provider, in different implementations, and with different functional characteristics. The provider has to determine at runtime, which supplier to use in the composite service and has to manage the service provision in an automated fashion. Assuming the supplier fails under a SLA between the provider and the supplier, a fail-over supplier is provisioned and the execution is re-established automatically.

The SLA between the customer and the provider is fulfilled without the customer being aware of failures and the interaction with other SLAs that exist in

Table 2. Executable paper use case steps

DESCRIPTION	Step	Action
	1	Publisher asks the authors to describe the list of requirements needed to reproduce the experiment in term of CPU, memory, input data, software, special device, and list all Intellectual property issues
	2	Publisher decides based on the input received from the author either to deliver paper in an executable form or not.
	3	Publisher inform the author that the paper is going to be published in executable version, and ask them to prepare the virtual Machine
	4	The authors generate a Virtual Machine to re-run experiment and all the data needed for the re-execution
EXTENSIONS	Step	Branching Action: Publisher does not accept to publish the paper in an executable form
	5	Publisher publish the paper is a static form

the architecture; i.e. that the data, QoS, outage requirements between the customer and the service provider are fulfilled or the SLA consequence occurs. In Table 2, we identify the steps needed to publish the paper or not in an executable form. These steps describe the interaction between two actors: the publisher and the authors. The publisher initiates this use case after the paper has been accepted for publication. Not all papers can be published as executable papers because they are either very expensive to reproduce, need special hardware or software (intellectual property issues), or request access to private data that are not likely to be provided (privacy issues). In Table 3, we identify the steps needed to execute an executable paper. These steps describe the interaction between two actors: the publisher and the provider of the computing infrastructure. This use case is initiated by a scientists who want to re-execute a published experiment.

Table 3. Executable paper use case steps

DESCRIPTION	Step	Action
	1	Scientist request to re-executed an experiment published in an executable paper,
	2	Publisher offers different ways of re-execution of the experiment fast, immediate, slow, scheduled (each has a given cost)
	3	Scientist select one way to re-execute
	4	Publisher contact the infrastructure provider and ask to run the experiment based on the SLA established between the publisher and the infrastructure provider
EXTENSIONS	Step	Branching Action: scientist does not accept to any of the proposed way for re-execution
	5	Publisher drop a request for execution and close the case

6 Conclusion

We have identified a number of challenges facing the implementation of EP concept, we have classified them into three categories: technical, administrative, and intellectual property. In this positioning paper we have described one approach to address the technical challenges and identified the role that each actor involved in the lifecycle of EP. Among other issues we have stressed in this paper, the issue of provisioning the needed infrastructure when the EP is published, and pointed out a technique that can help to solve this problem which is the use of new virtualization techniques to provide a working environment for the published experiments. We discussed the feasibility of this technique and described two scenarios related to the operational aspects associated with the deployment of an executable paper service and the role of each actor throughout the executable paper lifecycle.

We believe that publisher can play a key role in implementing the EP concept, in our proposal the publisher does not have to develop in-house the expertise to

maintain the infrastructure needed to re-run published experiments. The publisher can outsource this task a to providers of computing infrastructure like Grid and Cloud providers. In order to achieve a certain QoS of the EP service, the publisher has to have a well established service level management with the infrastructure providers.

However there are open IP questions, which has to be solved. Typically a license is acquired for a single copy of software running on a specific hardware. For a server software like Microsoft Windows Server a license is needed even for each client who uses Microsoft server technology. With IaaS approach it is not easy to track which software is used for what and how many times. This is a main reason why IaaS providers encourage their users to use free open source solutions on cloud resources. In order to give the authors the legal possibility to use proprietary software and tools for their EP a new license strategy which combines the two approaches (IaaS and SaaS) has to be developed.

References

1. Markup Languages, http://en.wikipedia.org/wiki/Markup_language
2. Jones, J.: Tracking Software Assets on Virtual Images Gains Momentum for Software Asset Management Professionals, http://blogs.flexerasoftware.com/elo/2010/07/tracking-software-assets-on-virtual-images-gains-momentum-for-software-asset-management-professional.html
3. Basant, N.S.: Top 10 Cloud Computing Service Providers of 2009 (2009), http://www.techno-pulse.com/2009/12/top-cloud-computing-service-providers.html
4. Hey, B.: Cloud Computing. Communications of the ACM (51) (2008)
5. Armbrust, et al: A View of Cloud Computing. Communications of the ACM 4(53) (2010)
6. Strijkers, R.J., Cushing, R., Vasyunin, D., Belloum, A.S.Z., de Laat, C., Meijer, R.J.: Toward Executable Scientific Publications. In: ICCS 2011, Singapore's Nanyang Technological University, June 1–3 (2011)
7. Limare, N., Morel, J.M.: The IPOL Initiative: Publishing and Testing Algorithms on Line for Reproducible Research in Image Processing. In: ICCS 2011, Singapore's Nanyang Technological University, June 1–3 (2011)
8. Kauppinen, T.J., Mira de Espindola, G.: Linked Open Science - Communicating, Sharing and Evaluating Data, Methods and Results for Executable Papers. In: ICCS 2011, Singapore's Nanyang Technological University, June 1–3 (2011)
9. McHenry, K., Ondrejcek, M., Marini, L., Kooper, R., Bajcsy, P.: Towards a Universal Viewer for Digital Content. In: ICCS 2011, Singapore's Nanyang Technological University, June 1–3 (2011)
10. Strijkers, R., et al.: AMOS: Using the Cloud for On-Demand Execution of e-Science Applications. In: IEEE e-Science 2010 Conference, December 7–10 (2010)
11. Kertesz, A., et al.: An SLA-based resource virtualization approach for on-demand service provision. In: Proceedings of the 3rd ACM International Workshop on Virtualization Technologies in Distributed Computing (2009)
12. Belloum, A., Inda, M.A., Vasunin, D., Korkhov, V., Zhao, Z., Rauwerda, H., Breit, T.M., Bubak, M., Hertzberger, L.O.: Collaborative e-Science Experiments and Scientific Workflows. In: IEEE Internet Computing (August 2010)