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

Selection and/or peer-review under responsibility of Prof. Mitsuhisa Sato and Prof. Satoshi Matsuoka
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